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

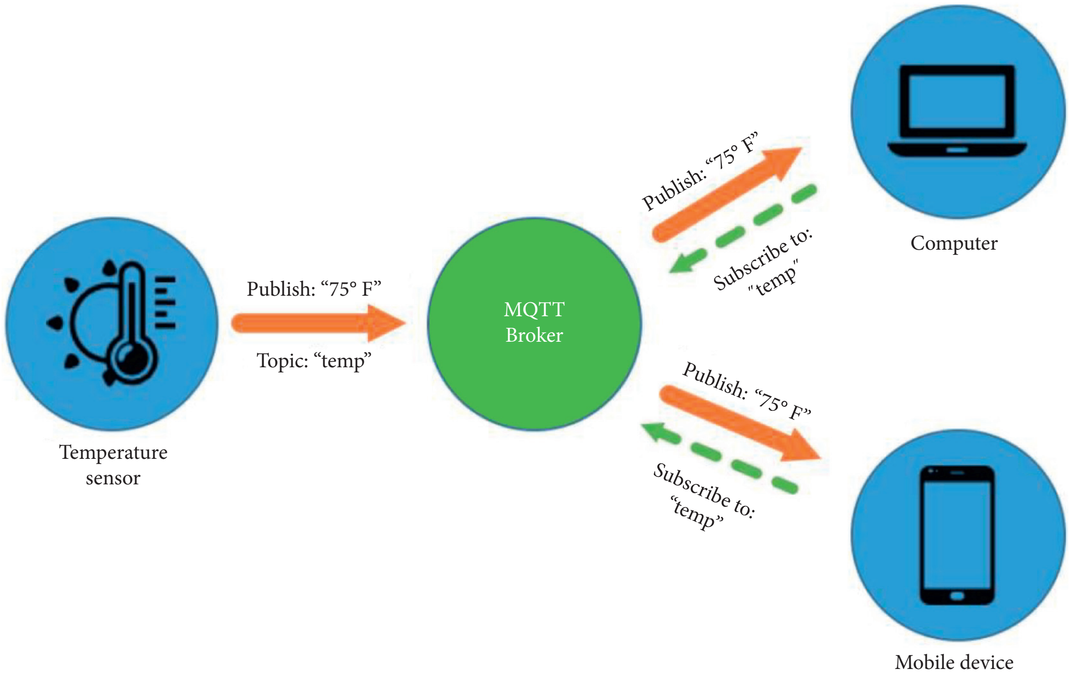


Figure 1: M2M communication based on the MQTT protocol (https://bbs.huaweicloud.com/).

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],

$$\underset{a}{\max} \; \underset{w, b}{\min} \; L(w, b, a). \tag{6}$$

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]:

$$\underset{w, b}{min} \left\{ J(w, \xi) = \frac{1}{2} w^T w + c \sum_{k=1}^{N} \xi_k \right\}, \tag{8}$$

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]:

$$L(w, b, \xi, a) = \frac{1}{2} w^T w + c \sum_{k=1}^{N} \xi_k - \sum_{k=1}^{N} a_k \left[t_k\left(w^T x_k + b\right) - 1 + \xi_k\right] - \sum_{k=1}^{N} v_k \xi_k, \tag{9}$$

where $v_k \geq 0, k = 1, \ldots, N$ is a new set of Lagrange multipliers, in addition to α_k. Thus the optimization problem is described as follows [30, 31]:

$$\underset{a, v}{\min} \; \underset{w, b, \xi}{\min}^{L(w, b, \xi, a, v)} . \tag{10}$$

So [30, 31]

$$\begin{gathered} \underset{a}{\min} \, Q(a) = \sum_{k=1}^{N} a_k - \frac{1}{2} \sum_{i=1}^{N} \sum_{m=1}^{N} a_l a_m t_l t_m x_l^T x_m, \\ \sum_{k=1}^{N} a_k t_k = 0, \quad 0 \leq a_k \leq c, k = 1, \ldots, N. \end{gathered} \tag{11}$$

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 $H \in R^{(l+u) x n_i}$ 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:

$$\beta = \left(I_{n_h} + H^T C H + \lambda H^T L H\right)^{-1} H^T C \tilde{Y}. \tag{12}$$

If $n_h > N$, then the output weights β are calculated using the following equation:

$$\beta = H^T \left(I_{l+u} + C H H^T + \lambda L H H^T\right)^{-1} C \tilde{Y}. \tag{13}$$

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]

$$x^* = \underset{x \in U}{\arg\max} - \sum_i p_\theta\left(y_{C_i} \mid x\right) \log\, p_\theta\left(y_{C_i} \mid x\right), \tag{14}$$

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]:

$$x^* = \underset{x \in U}{\arg\max} - \sum_i \frac{V\left(y_{C_i}\right)}{k} \log \frac{V\left(y_{C_i}\right)}{k}, \tag{15}$$

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]:

$$x^* = \underset{x \in U}{\arg\max} \frac{1}{k} \sum_{c=1}^{k} \sum_i p_{\theta(c)}\left(y_{C_i} \mid x\right) \log \frac{p_{\theta(c)}\left(y_{C_i} \mid x\right)}{p_c\left(y_{C_i} \mid x\right)}, \tag{16}$$

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

$$h(C, q, m, L, U). \tag{17}$$

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].

```
Input $L$–labeled data, $U$–unlabeled data, $C$–regularized S-hFFNN, $q$–active learning strategy, $m$–new labeled data, $maxIter$–max iterations
$h \longleftarrow C(L_0)$
for $i$ from 1 to maxIter
    choose $m$ from $x \in U$ based on $q$ strategy
    $\omega \longleftarrow f(x)$
    $L \longleftarrow L \bigcup (x, \omega)$
    $U \longleftarrow U - (x, \omega)$
    $h \longleftarrow h(L_0)$
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].

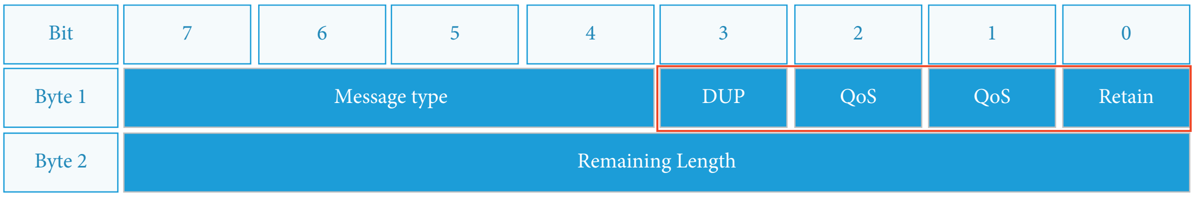
Respectively, the variable header, when exists, contains the data shown in Figure 3 [10].

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].

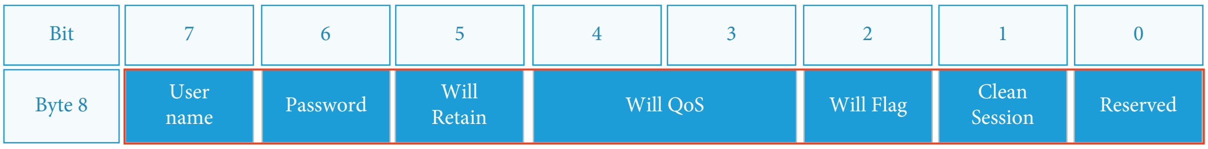
To carry out a more thorough analysis that can allow the proposed intelligent system to perform better, without compromising its predictive capacity, the initial dataset was reduced based on evaluative criteria of the information provided in each feature [10, 12, 42]. This stage is critical for this 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



Bit	7	6	5	4	3	2	1	0
Byte 1	Message type				DUP	QoS	QoS	Retain
Byte 2	Remaining Length							

Figure 2: MQTT fixed header (https://support.smart-maic.com/).



Bit	7	6	5	4	3	2	1	0
Byte 8	User name	Password	Will Retain	Will QoS		Will Flag	Clean Session	Reserved

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.