Hindawi
Security and Communication Networks
Volume 2023, Article ID 8168075, 13 pages
https://doi.org/10.1155/2023/8168075

Research Article

A Blockchain-Oriented Framework for Cloud-Assisted System to Countermeasure Phishing for Establishing Secure Smart City

Narendra Kumar,[1] Vikas Goel,[2] Raju Ranjan,[3] Majid Altuwairiqi,[4] Hashem Alyami,[4] and Simon Atuah Asakipaam[5]

[1]*VPLearning Closet Pvt. Ltd., Noida, Uttar Pradesh, India*
[2]*Department of IT, Kiet Group of Institutions, Ghaziabad, India*
[3]*School of Computing Science and Engineering, Galgotias University, Greater Noida, India*
[4]*Department of Computer Science, College of Computers and Information Technology, Taif University, Taif, Saudi Arabia*
[5]*Department of Electrical and Electronics Engineering, Tamale Technical University, Tamale, Ghana*

Correspondence should be addressed to Simon Atuah Asakipaam; simonasakipaam@gmail.com

Received 3 July 2022; Revised 23 July 2022; Accepted 9 August 2022; Published 21 April 2023

Academic Editor: Rabie Ramadan

The information that is saved in the cloud about users is protected by a number of different safeguards in order to facilitate the development of smart cities. Phishing and other forms of social engineering are two examples of misleading tactics that may be used by hostile actors to get sensitive information about users. Phishing is still the first step of a multistage assault, despite the significant technological advancements that have been made to it in recent years. Phishing kits have evolved to become attack tools that are much simpler, more user-friendly, and more readily available as time has gone. Indicators of a successful phishing assault include using foreign characters in the URL, typosquatting of prominent domain names, reserved characters in redirections, and multichain phishing. When papers with these types of phishing URLs are uploaded to cloud storage, hackers get a helping push in the right direction. The use of cloud servers in the commission of these assaults is becoming more common. The currently available software to disallow list phishing URLs does not provide sufficient protection against multilevel phishing and instead places the onus of safety on the user to protect themselves. In addition, the immutability of blockchain data and the avalanche effect both demonstrate their effectiveness as crucial safety measures. In view of the recent advances in technology, we suggest an implementation of filtering that is based on blockchain technology to safeguard the cloud-based data of users. The Phish Block that has been presented is able to recognize homographic phishing URLs with an accuracy of 91 percent, thus ensuring the security of cloud storage.

1. Introduction

The bulk of technology users, including financial services, have shifted their focus to cloud resources as the demand for these services has increased. It is possible that this may encourage the attackers and turn cloud servers into a target for security breaches. Because other cloud users sometimes upload stuff that is both dishonest and harmful, the papers that are stored in the cloud could not be completely secure. There are numerous different methods in which phishing assaults might occur. Emails are the primary vector for phishing attacks. Phishing may also take place via the use of a technique known as angler phishing. Social media is a relatively new attack vector, and it provides a lot of different options for perpetrators of attacks to deceive victims. To trick users into divulging private information or downloading malware, imposter URLs, cloned websites, postings, tweets, and instant messaging is one of the tactics that are used. Attackers are able to build highly targeted assaults by making use of the data that users voluntarily provide on social media platforms. The use of phishing URLs, which may mislead or misdirect users of cloud computing environments, is the primary kind of attack that is feasible in a cloud computing environment. Because the primary purpose of phishing is to

steal the data of the user, the perpetrators of the assault make an effort to get access to the user's information without the user's awareness. The vast majority of phishing assaults begin with a URL that has been carefully constructed. Phishing URLs, when clicked on, take users to fake websites, download malicious software, or request login credentials from users. The user is tricked into accessing these websites because the false URL seems remarkably similar to the actual URL. Cloud computing, in its most basic form, refers to the practice of accessing and storing data across a linked network rather than on the hard disc of a computer.

The data and information that are saved in the cloud are protected from the majority of potential threats. The user who creates the data that is stored in the cloud is the one who is responsible for its creation, but the cloud service provider (CSP) has complete authority over the contents of the cloud. People like the cloud not only because it can store data, but also because utilizing cloud services enables them to exchange files and documents with other people. This is one of the main reasons why people favor the cloud. The cloud service may also be used for the purpose of generating backups of the data in order to safeguard vital documents and other data. The data may be retrieved from the cloud storage facility in the event that anything catastrophic occurs to the local computer. Sharing data via cloud computing makes it possible for several participants to freely share the data of a group. This not only increases the productivity of work done in collaborative settings but also offers a broad range of potential applications [1]. Despite the development of a number of different encryption methods, maintaining the cloud storage's level of security continues to be challenging. A blockchain is a list that is constantly being added to. It is a record kept in digital format of transactions. It gets its name from its structure, which consists of individual entries being connected together in a single list known as a chain. These individual records are called blocks. Transactions that are done using cryptocurrencies are recorded using a technology called blockchain. The term "cryptocurrency" refers to any digital asset that may be traded like traditional currency. It keeps track of and maintains a ledger or record in a digitally computerized database that contains the ownership records of individual coins. For the purposes of protecting, creating, and verifying transaction records as well as their respective ownership, the database is encrypted using robust encryption. As a result, they are referred to as distributed ledgers since no centralized authority is responsible for their issuance or approval.

Figure 1 shows the structure of block. The avalanche effect is a characteristic of cryptography that describes what happens when a little change is made to the input but results in a significant change to the output. If the information included in a single block of the blockchain was altered in any way, the hash value of that block would be subject to significant revision. Because the hash value that is created in each subsequent block contains the hash value of the block that came before it, any change in the content of a single block results in a change in the hash value of all of the blocks. As a result, the blockchain is resistant to modifications and updates, which helps to ensure that the integrity of the

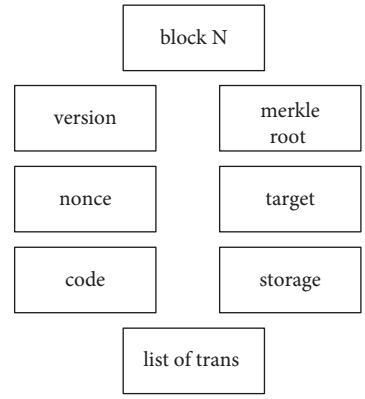

Figure 1: Structure of a block.

recorded data is maintained. Horst Feistel was the first person to adopt the phrase "avalanche effect," despite the fact that Shannon's diffusion was already using the notion. It is one of the most important goals of the design process for a hash function that makes advantage of the "butterfly effect." Through repeated uses of the hash algorithm, it enables even very little changes to have a significant impact in a short amount of time. Due to the avalanche effect, which makes the blockchain totally resistant to alterations, anything that has been entered as a block can only be read after it has been added, and it cannot be modified once it has been added. The user may begin the upload after they have successfully navigated the security system that is given by the cloud provider. The credentials of the user, which are maintained inside the blockchain, once again adhere to the avalanche effect, which guarantees that non-repudiation will occur. Modeling the data access and acquisition operations of the service provider as a series of access records that offer information about the data created and consumed for the service is one way to represent these activities. However, access records [2] are of no value if they cannot be relied upon, and it is not a good idea to place your faith on access records if you are not being provided with the appropriate protection. It is of the utmost importance to ensure that access records are both accurate and impossible to falsify.

Because of these many security considerations, blockchain technology has been selected as the platform of choice for safeguarding cloud servers. A Blockchain-as-a-Service (BaaS) platform can provide developers with convenient, high-performance blockchain ecosystems [3] and related services by embedding the blockchain framework into the cloud computing platform. This allows the platform to leverage the deployment and management advantages of cloud service infrastructure. There has not been any implementation of a phish-detecting blockchain yet. The currently available methods for preventing phishing URLs are carried out on the browser level utilizing domain certifications. In addition, there are several tools available, which enable client programs to verify URLs against continually updated lists of risky online sites. This helps to protect cloud users from falling victim to phishing scams. The usage of blockchain technology to preserve phishing information might be helpful in identifying the perpetrator

of the crime. Together, the non-repudiation feature of the cloud and the integrity of the documents stored in the blockchain are helpful in determining whether users are engaging in malevolent activity. Users will be able to protect themselves against phishing attacks if the content of fraudulent documents is exposed and made transparent. Criminals can then no longer hide their tracks. It will be less important for users to make sure they are using secure browsers if the proposed Phish Block is implemented. It functions as a utility between the users of the cloud and the storage while insertions and updates are being made, hence eliminating the need for a twofold check when accessing data from the cloud.

The objective of the Phish Block solution that has been developed is to distinguish safe homographic URLs from harmful homographic URLs inside the data that is being saved in the cloud environment. Adding a blockchain service to a cloud environment provides an additional degree of security, which will be to the users' advantage. The deployment of blockchain-based document filtering that has been suggested would guarantee that only valid documents are saved to cloud storage. By identifying phishing methods that are URL-based, the Phish Block technology prevents uploaded harmful documents from accessing the cloud and so protects users.

2. Literature Review

The blockchain technology is becoming an essential component of many different systems. In fog computing, a scalable blockchain may be created using a structure called groupchain that consists of group blocks and vice blocks. The transactions that take place in the environment that was formed are checked and authorized by a leader group using a round robin process. This helps to lower the confirmation latency while simultaneously increasing the transaction throughput. Through this solution, both selfish mining and unnecessary expenditure may be avoided [4]. Concerns relating to privacy, reputation systems, and transaction negotiation may be amenable to resolution using a platform like blockchain-based CloudEx [5]. Policy driven permissioned blockchain network has been designed for transport systems with a set of policies which contain the signing key of each user, and these signing keys are associated with a policy set [6]. In order to guarantee the integrity of the huge data throughout the process of controlling the Internet of Things, a permissioned blockchain that uses decentralized administration was used. A blockchain-based token incentive system has also been implemented in order to improve the overall quality of the data that has been provided. This particular blockchain architecture may potentially be practicable for very large amounts of data [7]. NutBaaS is a Blockchain-as-a-Service platform that has been designed as a layered architectural design. It has the capability of providing blockchain services to cloud computing environments. The development of blockchain ecosystems also includes the provision of various security services. Using a multilayered strategy, blockchain technology may offer security for the Internet of Things (IoT) and answer concerns about the confidentiality, integrity, and availability of IoT data. In order to guarantee the system's safety, the SHA-2 hashing algorithm is often implemented using Merkle trees and hash tables [8]. Blockchain taxonomies such as consensus protocols, smart contracts, and forks have been used to ensure the safety of cryptocurrencies, in addition to the Internet of Things (IoT) security. The immutable blockchain enables authorized access to the whole transaction history as well as multi-token trades [9]. For the purpose of bolstering the safety precautions taken by the wireless sensor nodes, a hybrid blockchain model has been developed to provide mutual authentication. Integrity, non-repudiation, and flexibility are provided by the hybrid model [10], which is comprised of base stations, cluster head nodes, and ordinary nodes. A blockchain-based smart contract that ensures fair remuneration may take the role of an independent auditor and help make auditing more safe. Storage protects private information and guarantees that parties do not have to communicate with one another during audits [11]. Blockchains are a well-known method used in cloud settings for the purpose of maintaining security. It is possible for the blockchain, which contains the data, to be stored on the cloud. When data is stolen in this cloud computing environment, a "block and respond" request is sent to the cloud owner. This helps to verify that data integrity is maintained. Hash trees and encryption algorithms were used to offer this security in order to ensure that cryptographic transactions are both quick and safe [12]. There have been several efforts that have focused on the smart contracts with the purpose of tailoring blockchain for certain purposes. Along with the explanation of a full overview of smart contracts by using Ethereum and Hyperledger blockchain frameworks, [13] also includes a suggestion of a six-layered framework that covers the essential parts of smart contracts. This framework covers the major features of smart contracts.

There has been development of a smart contract architecture that includes access control contracts (ACC), judge contracts (JC), and register contracts (RC). There are a variety of methods that may be used to identify and foil phishing scams. A SAFE-PC (Semiautomated Feature Generation for Phish Classification) model that carries out keyword extraction, feature engineering, and natural language processing in order to filter phishing attacks that come through the Internet and electronic mails is a method that can be utilized to protect against these types of attacks. SAFE-PC addresses real-world issues with a portable feature selection and uses the fastest boosting algorithm (RUSBoost) as a classifier. Its performance is superior to that of other filtering applications such as Sopho and SpamAssassin [14]. In order to identify instances of phishing on the Internet, an Adaptive Neuro-Fuzzy system was developed. This system takes a layered approach, integrating aspects of text, images, and frames. Phishing websites may be identified as having hybrid traits by the use of ANFIS feature classification, SVM, and KNN [15]. A fog network has to be established in order to identify and delete the phishing URLs. The phishing URLs are located by this fog network via the use of feature extraction, which is performed on online traffic characteristics. The WHOIS identification tool and the Google API open the

system up to massive amounts of real-time data and provide it with a higher quality of service [16]. The prediction of phishing is based on a generally used collection of 12 indicators that are collected from research conducted by third parties. These characteristics are a collection of URL patterns that are used by respectable websites with the goal of phishing the site [17].

Carrying out a phishing assault provides additional opportunities for learning about the aftermath of the attack as well. Extreme phishing attacks that look and feel nearly exactly like the genuine websites that are being targeted have been developed and exhibited to assess how successful they are. It was determined that 92 percent of the subjects did not exhibit any suspicious behavior [18]. There have been discussions on the creation of a variety of updated URLs and updated contents with the intention of tricking people. It has been said that not only URLs but even logos and visuals are phished in spam emails, which makes it exceedingly difficult to identify the existence of phishing since it makes it more difficult to tell what is being impersonated. Phishing is being addressed using a solution that consists of three stages: prevention, detection, and training for stakeholders [19]. Along with a comprehensive examination of the distinguishing traits that set spear phishing apart from regular phishing, a comprehensive description of the absence of preventative actions that can be taken against spear phishing is also provided [20]. There are ideas floating about with various detection methods for the many kinds of phishing, such as studies on the various kinds of phishing and spear phishing assaults. These ideas have been proposed [21].

3. Proposed Phish Block

The usage of blockchain technology to preserve phishing information might be helpful in identifying the perpetrator of the crime. Finding the malicious user is made easier because of the non-repudiation aspect of the cloud, which, when paired with the integrity of the documents stored in the blockchain, makes it possible to identify the user. The consumers have a better chance of becoming more knowledgeable about the phishing tactics that the crooks may utilize as the content of the fraudulent papers becomes more transparent. It will be less important for users to make sure they are using secure browsers if the proposed Phish Block is implemented. It functions as a utility between the users of the cloud and the storage while insertions and updates are being made, hence eliminating the need for a twofold check when accessing data from the cloud. The papers that are uploaded to cloud storage are screened for legitimacy by the Phish Block method that has been suggested. A homographic phishing URL detector is used by the employed framework of smart contract algorithms to identify documents that include phishing material. Once identified, these documents are withheld on the blockchain, which has the feature of the avalanche effect. The improved Proof-of-Work algorithm is used in the process of selecting the block miner from among the cloud users. After the block has been mined, the contents of the block are made available to all of the users of the cloud. Users who are authorized to utilize the cloud will now be aware of phishing. When the process of screening has been finished, any papers that have been left over but have not been uploaded to the blockchain will be regarded as secure. The framework of the smart contract will begin to operate on the most recent block of Phish Block as soon as an input has been identified. If the compilation of the contracts is successful, the block that contains the content of the malicious input document is mined via the Enhanced Proof-of-Work into the Phish Block. The remaining papers are now being uploaded to the server in the cloud. The Phish Block module will initially use the smart contract architecture in order to identify potential phishing URLs. Blocks are made out of any papers that are discovered to include phishing URLs, and these may then be removed. The Enhanced Proof-of-Work algorithm makes it possible for users to mine blocks based on whether or not a smart contract is legitimate. Only once a phishing URL has been successfully identified and removed is the assembled contract regarded to be legitimate. The deployment of a contract that is not legitimate does not result in the mining of the block. During this phase of the mining process, one participant will be chosen to take on the role of miner. That participant will be responsible for adding the document that contains the harmful code to the blockchain.

Figure 2 shows the phish detection architecture. The safe documents are encrypted using SHA-3 and sent to the respective cloud storage centers. As shown in Figure 2, the input documents are obtained from cloud users.

Once the document is added to the blockchain, its block contents are made visible to all the users. The documents that were identified as safe are for encryption. A user-friendly interface is created to communicate with the blockchain and display the content of the documents that were added to the blockchain in the enhanced PoW process. The safe documents are encrypted and sent to the cloud servers. Figure 3 shows the flow of functionalities incorporated by the modules of detection inside the contract framework. Check homograph is responsible for checking whether the given URL is a homographic phishing URL or not. Three strategies of homographic URL detection are considered, namely, Internationalized Domain Name in Applications (IDNA), typosquatting, and Reserved Character Usage (RCU). If any of the URL detection techniques returns true, the URL is directly considered to be phishing; otherwise, it is sent for detecting chained phishing.

(i) Internationalized Domain Name in Applications (IDNA). It extracts the domain name from the given URL and checks for homographs using multilingual characters.

(ii) Typosquatting. It extracts the domain name from the given URL and checks for homographs using deceptive spellings.

(iii) Reserved Character Usage (RCU). It searches for the reserved characters on a URL that can be used as an escape for redirections.

Web crawling is done with the URLs detected from the documents for the web page content to find the possible

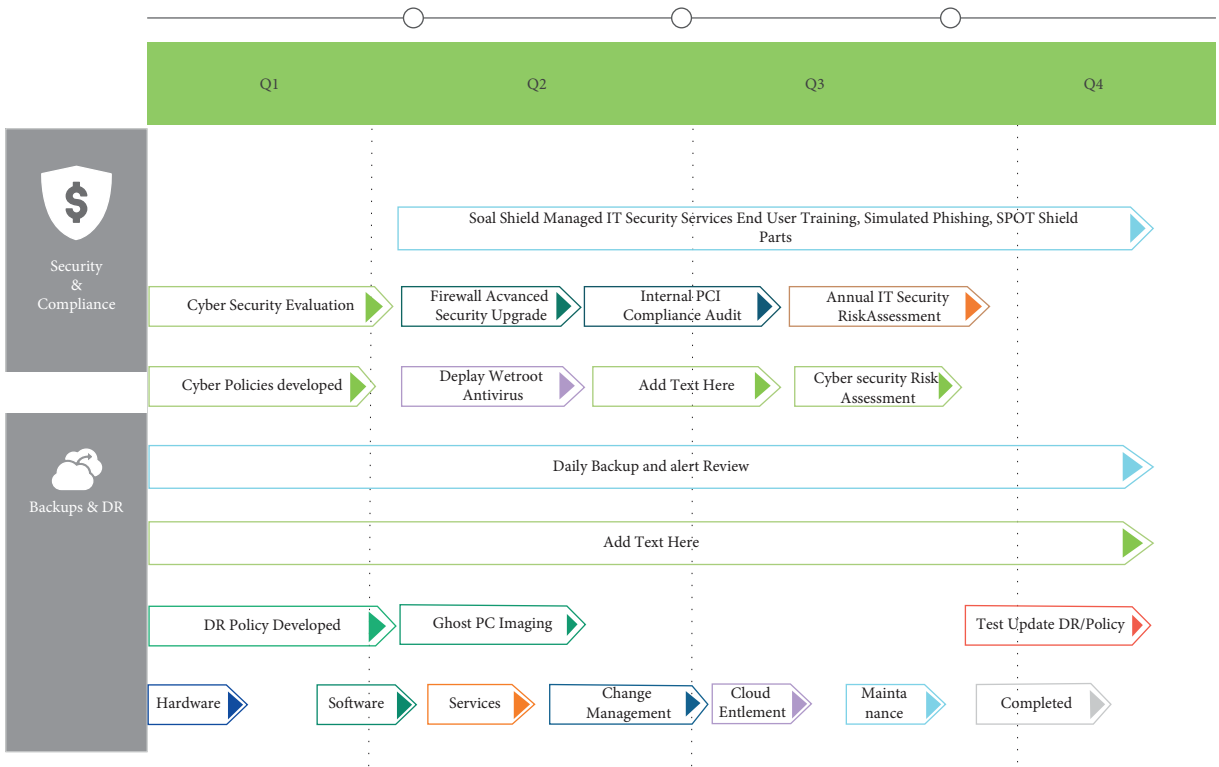

Figure 2: Proposed phish detection blockchain architecture.

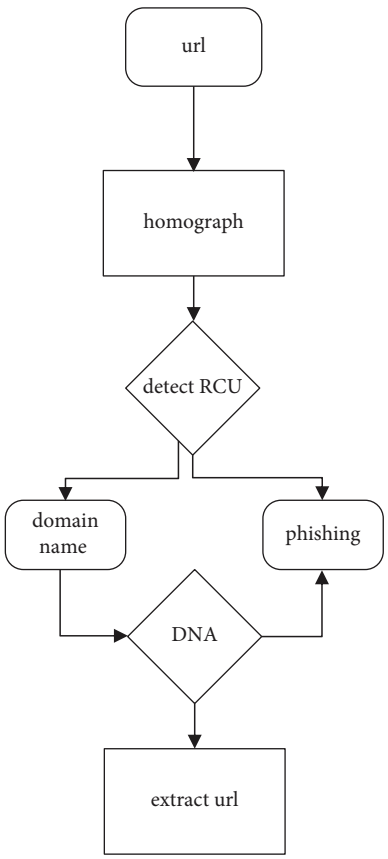

Figure 3: Flow of detection by the proposed Phish Block smart contract.

hyperlinks. The found hyperlink URLs are checked for homographs recursively until no more hyperlink is found.

As shown in Algorithm 1, the Phish Block algorithm takes a random list of documents as input from the dataset. The dataset contains 200 documents with and without phishing URLs. Phishing URLs are coined or referred from Wandera and Phishbank for creating .txt files. The input documents in the list are considered as those the cloud users try to upload to the cloud. The list is traversed to obtain each document. The document is scanned for the presence or absence of a phishing URL. The presence of phishing URL confirms the validity of the contract and deploys the same for creating a block with the document content as the entry. Each block contains a nonce value, hash value, difficulty, coin base, timestamp, file data, gas limit, and configuration details as fields. The content found to have the phishing URL is placed as the file data of the block. The count of blocks in the Phish Block increases as the number of malicious documents in the input list increases. Once the block is created, the Proof-of-Work employed by the Ethereum blockchain chooses the miner for Phish Block and the block gets mined.

The input list is traversed again to remove the documents corresponding to the created blocks from the list. The entire process is repeated until the list reaches the end. All the malicious documents are removed from the list after the completion of these iterations, causing the list size to either remain the same or decrease. The list size remains the same if there are no malicious documents among the list of selected documents. The list size gets reduced according to the respective number of malicious documents in the randomly

```
//n represents the number of documents, doc represents the list containing the n documents, a block is the genesis block
Input:
List of documents (LD)
Output:
Phish Block
Procedure:
input "n" documents in a list "doc"
initialize i = 0
initialize j = 1
do
Scan doc[i]
if (doc[i].CHECK_HOMOGRAPH() = = TRUE) then
   create_block()
   block[j] = doc[i]
   increment j
   PhishBlock_PoW
end
if (User_solves_PoW AND user_details IS VALID) then
   user ←minercreate_block()
   if (add_block == TRUE) then
   initialize k = i
   for k in n − 1:
   doc[k] = doc[k + 1]
   increment k
   end
end
increment i
while (doc.next! = NULL)
display Block Contents
initialize x = 0
do:
   encrypt_doc = Encrypt doc[x]
   add encrypt_doc to cloud
   increment x
   while (doc.next! = NULL)
end
```

Algorithm 1: Phishing detection blockchain.

selected input list. The documents remaining in the list are safe to upload to the cloud. An encrypted version of these files is ready for cloud storage. Algorithm 2 shows the traversal of each and every inputted document to check the presence of phishing URL. The code converter is used to convert the analog to digital values in the proposed work with the help of the boyer Moore algorithm. Punycode convertor has been incorporated by various browsers for phish detection and is the conversion tool that can be incorporated in Python. It is a simple and efficient transfer coding syntax designed to be used with IDNA [24]. Under auto correction using Python, models like error model and candidate model are available. Error model sticks to the proximity of the characters in the keypad for suggesting auto correction whereas candidate models use distance calculation of the words against a dictionary. Under text distance calculation, there are several categories like edit-based, token-based, sequence-based, and phonetic-based. Taking into consideration computational efficiency, Jaccard distance algorithm, a token-based technique, has been used [25]. Reserved characters like ";", ",", and "@" are used in URLs for redirection and are considered as escape characters. A URL with any other domain name followed by reserved characters can be a phishing URL. A naive pattern search algorithm can detect the same [25].

As shown in Algorithm 3, the multichain phishing is implemented using recursive calls. An empty list is given as the input for web crawling, and web contents are stored as .txt files into the list if hyperlinks are detected [26]. The items in the returned list are appended to scan for homographic phishing URLs again and again, until no such URLs are found. Web crawling uses Beautiful Soup, a web crawling framework in Python. Beautiful Soup enables the detection of multichain phishing. It also enables extracting URLs from web pages. It is used to visit web pages corresponding to the extracted URL and crawl through them for retrieving other available hyperlinks [27].

The proposed mathematical procedure aims at materializing the efficiency of the selected features for phish detection. The features used by Phish Block have been selected based on the ability to integrate with blockchain [28]. The rate of error detection can be found from the system's phish

```
Input: Document, D with x lines
Output: Boolean value
Procedure:
initialize i = 0
do:
if (D[i].is_URL == true) then
    RCU = pattern_search(URL, reserved characters)
  if (RCU == true)
    return true
  end
  DN = extract_domain_name(URL)
  IDNA = verify_punycode(DN)
  if (IDNA == true) then
    return true
  end
  TS = autocorrection_probability(DN)
  if (TS ≥ 0.4) then
    return true
  end
  MC = Multichain_Phishing(URL)
  if (MC is_not_null) then
    D.append(MC)
  end
end
  while (i is_lesser_than x)
  return false
```

Algorithm 2: Check_homograph.

```
//href_list represents the list of hyperlinks obtained from the source code of the web content corresponding to the extracted URL, EU
Input: Extracted URL, EU from Document, D
Output: href_list
Procedure:
  initialize href_list = empty
  crawl the HTML source code of EU
  extract hyperlinks
  href_list.add(hyperlinks)
  return href_list
```

Algorithm 3: Multichain_phishing.

block and logical parameters. This set of mathematical equations is used to prove the same.

Let w be the document that needs classification as safe or phishing [29]:

$$wX \longrightarrow \{\text{safe}, \text{phishing}\}. \tag{1}$$

Then, X is the anti-phishing Phish Block system that considers features $f_i \in w$, such that

$$w = \sum_{i}^{n} x f_i, \quad n > 0. \tag{2}$$

w is a non-empty set.

Table 1: Terms and description.

Terms	Description
PSR	Phish success rates
PFR	Phish failure rates
SSR	Safe success rates
SFR	Safe failure rates
P_P	Phishing sites classified as phishing
P_s	Phishing sites classified as safe
S_P	Safe sites classified as phishing
Ss	Safe sites classified as safe
P	Total number of phishing sites
S	Total number of safe sites

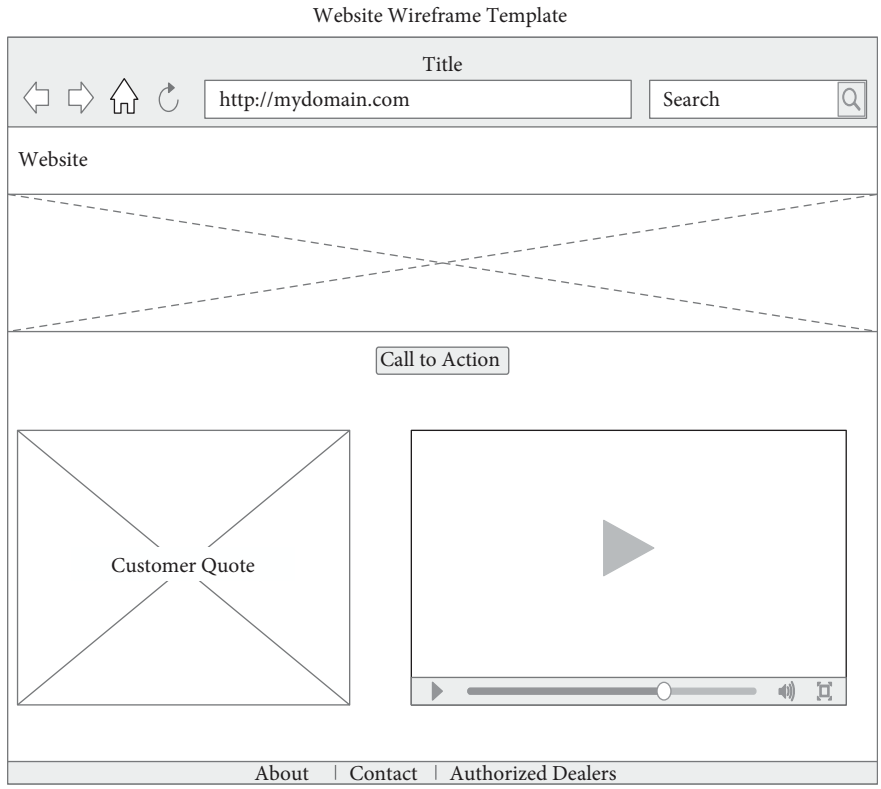

Figure 4: Ethers transferred to the accounts of the private blockchain.

The standards of classifications are applied to the system to analyze the accuracy. The used terms are shown in Table 1.

$$\begin{aligned} \text{PSR} &= \frac{P_P}{P} \times 100, \\ \text{PFR} &= \frac{P_S}{P} \times 100, \\ \text{SFR} &= \frac{S_P}{S} \times 100, \\ \text{SSR} &= \frac{S_S}{S} \times 100. \end{aligned} \tag{3}$$

Accuracy of detection is calculated by the following equation:

$$A = \frac{P_P + S_S}{S + P} \times 100. \tag{4}$$

The reliability of the Phish Block is calculated using Matthews Correlation Coefficient (MCC). When the chosen MCC approaches the value 1, the detection is considered chosen to perfection.

$$\text{MCC} = \frac{P_P \times S_S - P_S \times S_P}{\sqrt{(P_P + P_S)(P_P + S_P)(P_S + S_S)(S_P + S_S)}}. \tag{5}$$

The standards of classifications are applied to the system to analyze the accuracy. The used terms are shown in Table 1.

4. Implementation

4.1. Experimental Setup. The experimental setup for the implementation of the proposed Phish Block involves MetaMask, Rinkeby, Remix, Truffle, and Go Ethereum. MetaMask acts as a gateway to access Phish Block through Firefox browser. Rinkeby is a test network used to collect ethers for compiling the contracts in Phish Block [30], accessed via MetaMask. Remix is the Integrated Development Environment used to run and deploy the Phish Block smart contract. Truffle framework is used to integrate the driver code in Python with the Ethereum smart contract in solidity. Go Ethereum is the client where the accounts can be created and smart contracts for Phish Block can be implemented through Truffle suite. Web3.py library is used for interacting with the blocks [31]. SHA-3 algorithm is used for encrypting the safe documents. It is connected to MetaMask to interact with the private Ethereum blockchain. The interaction with the console is tested [32].

Rinkeby test network is used for collecting ethers. Figure 4 shows that the collected ethers are then transferred to the MetaMask account. Web3 is used to call Ethereum smart contracts [33] using Python. The web interface uses Python, JSON, JS, and Google scripts API.

4.1.1. Dataset. The dataset has been generated with and without URLs. Documents containing URLs consist of safe and phishing URLs. Phishing URLs are framed as homographs belonging to all the three strategies [34] for detection.

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=None, max_features='auto', max_leaf_nodes=N
one,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=1, n_jobs=
-1,
                       oob_score=False, random_state=None, verbose=0,
                       warm_start=False)
                                                                        Out[341]:
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=None, max_features='auto', max_leaf_nodes=N
one,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=2, n_jobs=
-1,
                       oob_score=False, random_state=None, verbose=0,
                       warm_start=False)
                                                                        Out[341]:
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=None, max_features='auto', max_leaf_nodes=N
one,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=4, n_jobs=
-1,
                       oob_score=False, random_state=None, verbose=0,
                       warm_start=False)
                                                                        Out[341]:
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=None, max_features='auto', max_leaf_nodes=N
one,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=8, n_jobs=
-1,
                       oob_score=False, random_state=None, verbose=0,
                       warm_start=False)
                                                                        Out[341]:
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=None, max_features='auto', max_leaf_nodes=N
one,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=16, n_jobs
=-1,
                       oob_score=False, random_state=None, verbose=0,
                       warm_start=False)
```

Figure 5: Compiling smart contracts using Truffle.

Phishing URLs are coined or referred from Wandera and Phishbank for creating .txt files.

Total documents generated: 200 (safe: 50, phishing: 150).

Documents with no URL: 25.

Documents with safe URL: 15.

Documents with multiple safe URLs: 10.

Documents with phishing URL (IDNA): 25.

Documents with phishing URL (typosquatting): 25.

Documents with phishing URL (RCU): 25.

Documents with multiple phishing URLs (IDNA): 25.

Documents with multiple phishing URLs (typosquatting): 25.

Documents with multiple phishing URLs (RCU): 25.

4.1.2. Implementation of Phish Block. As shown in Figure 5, the Geth client is accessed via Truffle framework. When the complete set of 200 documents are given as input, the files with safe content are encrypted and those with phishing content are added as blocks [35]. The block content and type of phishing as well as the block times are displayed as shown in Figure 6.

```
BaggingClassifier(base_estimator=DecisionTreeClassifier(class_weight=None,
                                                        criterion='gini',
                                                        max_depth=None,
                                                        max_features=None,
                                                        max_leaf_nodes=None,
                                                        min_impurity_decreas
e=0.0,
                                                        min_impurity_split=N
one,
                                                        min_samples_leaf=1,
                                                        min_samples_split=2,
                                                        min_weight_fraction_
leaf=0.0,
                                                        presort=False,
                                                        random_state=None,
                                                        splitter='best'),
                  bootstrap=True, bootstrap_features=False, max_features=1.0
,
                  max_samples=1.0, n_estimators=500, n_jobs=None,
                  oob_score=False, random_state=8, verbose=0, warm_start=Fal
se)
```

Figure 6: Classification of 200 safe and phishing files by Phish Block.

Table 2: Accuracy of Phish Block.

Test cases (files)	Input documents		Detection by Phish Block		Accuracy (%)
	Phishing	Safe	Phishing	Safe	
Test case 1 (50 files)	37	13	34	16	91.89
Test case 2 (100 files)	75	25	68	32	90.67
Test case 3 (150 files)	113	37	103	47	91.15
Test case 4 (200 files)	150	50	132	68	88
Average accuracy:					90.42 = ~ 91

An Ethereum interface has been developed for the interaction with the front end. It allows the safe documents to get uploaded to Google Drive (considered as cloud).

The proposed Phish Block system is tested through 4 test cases. The number of the input documents is increased gradually for each case [31]. Test case 1 takes 50 documents as input, test case 2 takes 100 documents as input, test case 3 takes 150 documents as input, and test case 4 takes 200 documents as input. These documents are randomly chosen by the driver program from the generated dataset containing 200 documents. For evaluating the system, two different measures have been considered. The first measure to be calculated is the accuracy of phish detection by Phish Block, which is shown in Table 2, for each test case [32]. The Phish Block system gives approximately 91% accuracy upon the generated dataset as an average. Figure 7 also clearly shows the misclassification of 9% of the actual files.

The block time is the other measure that has to be computed. It is the amount of time that passes between successive blocks being mined and added to the blockchain [33]. The duration of a block in Phish Block is determined by subtracting the timestamps of its predecessors from those of its successors as they are added to the blockchain in sequence. A function call to the smart contract is made in order to retrieve the timestamp of the newly inserted block. The smart contract then sends the timestamp of the currently active block to the driver application [34]. The total amount of time required to complete the most recent addition of Phish Block is 327 seconds. The result that was obtained is equal to the difference in the timestamps of the first block and the final block that was inserted in test case number 4.

5. Result Analysis

Figure 8 presents the findings, which indicate the inputs that were incorrectly identified for each test scenario. Test case number four has been shown to have the highest amount of incorrect classifications, while test case number one has been shown to be the most accurate of the four. The severely misspelt URLs that were identified by the typosquatting detection system led to the misclassification that occurred. Because our typosquatting detection relies on an auto correction technique, it is not possible to identify large degrees of variance in the text. Despite this, the efficiency of the system has not been compromised in any way since URLs with significant misspellings are easy for human eyes to see. Phish Block has achieved a maximum accuracy of 91.89 percent in test case 1 and a minimum accuracy of 88 percent in test case 4. This is due to the number of misclassifications, as well as the quantity of documents that were supplied as input in each of the relevant test cases. When evaluated with 100 files, the system demonstrates an accuracy of 90.67 percent, whereas in test case 2, it shows an accuracy of 91.15 percent.

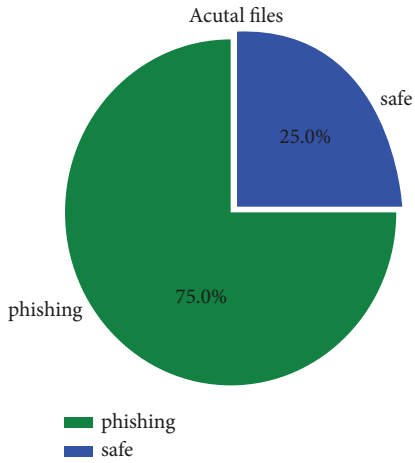

Figure 7: Displaying the ratio of the phishing files and safe files.

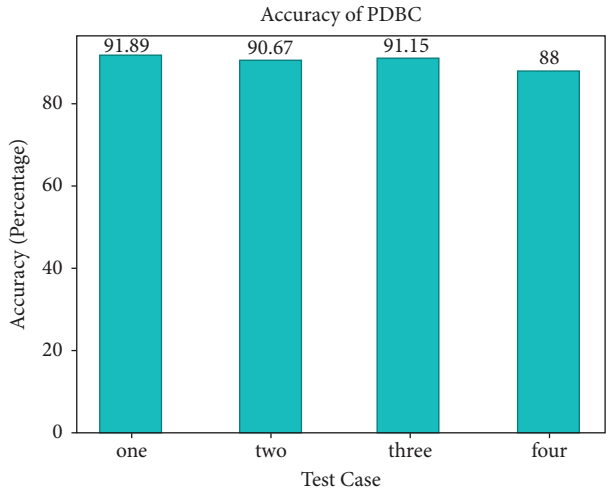

Figure 8: Accuracy shown by Phish Block for different test cases.

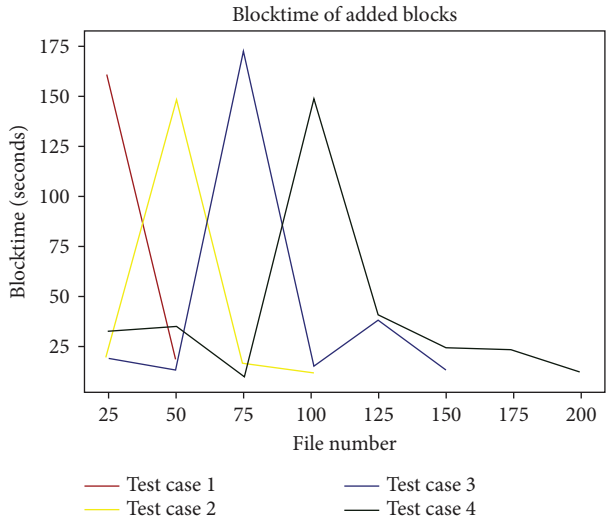

Figure 9: Graph representing the block time for different test cases.

In each of the test cases, the block timings of the subsequent blocks that were appended to the Phish Block are shown in Figure 9. Recordings of the block timings were taken at regular intervals of every 25 files that were provided as input. The files are mapped to the times at which their individual blocks were created in order to ensure that they are successfully mined into the Phish Block. When the succeeding blocks are added to the Phish Block at the same timestamp or when the accompanying files are not added to the chain, the block time is recorded as zero seconds. The blockchain does not include the files that do not include phishing URLs, and the block time values for such files are set to zero seconds. It is abundantly clear that the block timings of the blocks that were added in test case 1 suffered highest volatility among values that were not zero. This might be the explanation for the successful first run of the deployed contracts as well as the high level of accuracy achieved while identifying phishing files. The recording time of case 2 and case 4 is 147 seconds and 148 seconds which is taken as input in the system. In addition, the minimum block timings for test cases 2 and 4 are comparable, being 11 seconds and 10 seconds for 25 files, respectively. Case 3 reaches its maximum point with a block time of 175 seconds, which makes it the most successful of the tests. The first test case takes 20 seconds, which is the maximum time among the lowest block timings. Because succeeding blocks have identical timestamp values, the block duration has been cut down, which may be ascribed to this fact. At the conclusion of test case 4, it became apparent that the blocks were being mined to Phish Block at a quicker rate. According to the detected pattern, the blocks are mined slowly at first but then quickly pick up the pace as they get closer to the finish, which causes the block time values to fall. It has come to our attention that test case 3 has required the greatest amount of time for processing files, which ranges from 50 to 75 seconds. It is possible to draw the conclusion that the 25 files were either an examination of multichain phishing or alternative safe files. The system has shown signs of becoming more consistent as the test cases continue to be carried out. Both test case 2 and test case 4 have maintained a constant block time for blocks mined within the intervals of 25–50 and 75–100 seconds, respectively. It would seem that test case 1 has reached its maximum block time while processing the first batch of 0–25 files. It has been observed that the maximum value always happens up to the first 100 files in each of the four scenarios; beyond that point, the system begins to operate at a quicker pace. After conducting a thorough examination of the patterns, it has been deduced that the block duration of a newly added block may rapidly increase after an extended period of mining inactivity. In any case, the existence of harmful files in the inputs collected from cloud users is not playing the major role in real time, and as a result, the block timings of the blocks are anticipated to be quite high.

6. Conclusion and Future Work

The proposed Phish Block has been implemented on a private Ethereum blockchain, and it has been effective in keeping the homographic phishing URLs (about 91 percent). On the other hand, papers that were unnoticed included various sorts of phishing URLs (around 9 percent). The

phishing filter that has been suggested also includes a few restrictions. The default difficulty level designed for the Ethereum platform has been an overhead in block time. This is changeable when the Phish Block algorithm is applied on a self-configured private blockchain that the CSP could afford, and the modification leads to improved block time. Not only would the addition of the Phish Block system as a utility offer safety for cloud storage and cloud users, but it would also provide an additional value as a trust element in the service level agreement (SLA) that is supplied by the cloud service provider (CSP). Therefore, the suggested phishing block has the potential to bring about a significant influence on the customer's choice of cloud services among the several CSPs that are competing. In upcoming projects, we will be working to make the Phish Block more resistant against documents that include phishing content by including detection of various forms of phishing and providing sufficient compensation for the cost involved.

Data Availability

The data used to support the findings of this study can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Acknowledgments

This research was supported by Taif University researchers supporting project no. TURSP-2020/306, Taif University, Taif, Saudi Arabia.

References

[1] J. Shen, T. Zhou, D. He, Y. Zhang, X. Sun, and Y. Xiang, "Block design-based Key agreement for group data sharing in cloud computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 6, pp. 996–1010, 2019.

[2] J. Xue, C. Xu, and Y. Zhang, "Private blockchain-based secure access control for smart home systems," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 12, pp. 6057–6078, 2018.

[3] W. Zheng, Z. Zheng, X. Chen, K. Dai, P. Li, and R. Chen, "NutBaaS: a blockchain-as-a-service platform," *IEEE Access*, vol. 7, Article ID 134433, 2019.

[4] J. Mahatpure, M. Motwani, and P. K. Shukla, "An electronic prescription system powered by speech recognition, natural language processing and blockchain technology," *International Journal of Science & Technology Research (IJSTR)*, vol. 8, no. 8, pp. 1454–1462, 2019.

[5] K. Lei, M. Du, J. Huang, and T. Jin, "Groupchain: towards a scalable public blockchain in fog computing of IoT services computing," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 252–262, 2020.

[6] S. Xie, Z. Zheng, W. Chen, J. Wu, H. N. Dai, and M. Imran, "Blockchain for cloud exchange: a survey," *Computers & Electrical Engineering*, vol. 81, Article ID 106526, 2020.

[7] A. Patel, N. C. Debnath, and P. Kumar Shukla, "SecureOnt: a security ontology for establishing data provenance in semantic web," *Journal of Web Engineering*, vol. 21, no. 4, pp. 1347–1370, 2022.

[8] Y. Mu, F. Rezaeibagha, and K. Huang, "Policy-driven blockchain and its applications for Transport systems," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 230–240, 2020.

[9] S. B. Goyal, N. Pradeep, P. K. Shukla, M. M. Ghonge, and R. V. Ravi, *Utilizing Blockchain Technologies in Manufacturing and Logistics Management*, IGI Global, PA, USA, 2022.

[10] M. Zhaofeng, W. Lingyun, W. Xiaochang, W. Zhen, and Z. Weizhe, "Blockchain-enabled decentralized trust management and secure usage control of IoT big data," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4000–4015, 2020.

[11] D. Minoli and B. Occhiogrosso, "Blockchain mechanisms for IoT security," *Internet of Things*, vol. 1–2, pp. 1–13, 2018.

[12] P. K. Shukla, L. Sharma, K. Raj Bhatele, P. Sharma, and P. Shukla, K. I. Lakhtaria, Design, architecture, and security issues in wireless sensor networks," in *Next Generation Wireless Network Security and Privacy*, pp. 211–237, IGI Global, Hershey, PA, 2015.

[13] Z. Cui, F. Xue, S. Zhang et al., "A hybrid BlockChain-based identity authentication scheme for multi-WSN," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 241–251, 2020.

[14] P. Rani, P. N. Singh, S. Verma, N. Ali, P. K. Shukla, and M. Alhassan, "An implementation of modified blowfish technique with honey bee behavior optimization for load balancing in cloud system environment," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 3365392, 14 pages, 2022.

[15] H. Wang, H. Qin, M. Zhao, X. Wei, H. Shen, and W. Susilo, *Blockchain based fair payment smart contract for public cloud storage auditing*, Elseiver - Information Sciences, vol. 519, pp. 348–362, 2020.

[16] K. S. Khan, K. Saleem, M. M. Hazzazi, M. Alotaibi, P. K. Shukla, and M. Aqeel, "Human psychological disorder towards cryptography: true random number generator from EEG of schizophrenics and its application in block encryption's substitution box," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 2532497, 20 pages, 2022.

[17] P. C. Wei, D. Wang, Z. Yu, S. K. S. Tyagi, and N. Kumar, "Blockchain data based cloud data integrity protection mechanism," *Future Generation Computer Systems*, vol. 102, pp. 902–91, 2020.

[18] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F. Y. Wang, "Blockchain-enabled smart contracts: architecture, applications, and future trends," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 11, pp. 2266–2277, 2019.

[19] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract based access control for the internet of things," *IEEE Internet Of Things Journal*, vol. 6, no. 2, pp. 1594–1605, 2019.

[20] C. N. Gutierrez, T. Kim, R. D. Corte et al., "Learning from the Ones that got away: detecting new forms of phishing attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 6, pp. 988–1001, 2018.

[21] M. A. Adebowale, K. T. Lwin, E. Sanchez, and M. A. Hossain, "Intelligent Web Phishing Detection and protection scheme using integrated features of images, Frames and Text," *Elsevier- Expert Systems with Applications*, vol. 115, pp. 300–313, 2019.

[22] M. Kaur and D. Singh, "Multiobjective evolutionary optimization techniques based hyperchaotic map and their applications in image encryption," *Multidimensional Systems and Signal Processing*, vol. 32, no. 1, pp. 281–301, 2021.

[23] C. Pham, L. A. T. Nguyen, N. H. Tran, E. N. Huh, and C. S. Hong, "Phishing – aware: A Neuro – Fuzzy approach for anti – phishing on fog networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 1076–1089, 2018.

[24] C. M. R. D. Silva, E. L. Feitosa, and V. C. Garcia, "Heuristic based strategy for Phishing prediction: A survey of URL - based approach," *Elsevier- Computers and Security*, vol. 88, 2020.

[25] R. Zhao, S. John, S. Karas et al., "Design and Evaluation of the highly insidious extreme phishing attacks," *Elsevier- Computers and Security*, vol. 70, 2017.

[26] I. Vayansky and S. Kumar, "Phishing - challenges and solutions," *Elsevier - Computer Fraud and Security*, vol. 2018, no. 18, 2018.

[27] M. Kaur, D. Singh, V. Kumar, B. B. Gupta, and A. A. Abd El-Latif, "Secure and energy efficient-based E-health care framework for green internet of things," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 3, pp. 1223–1231, Sept. 2021.

[28] S. Gupta, K. K. Gupta, P. Kumar Shukla, and M. Kumar Shrivas, "Blockchain-based voting system powered by post-quantum cryptography (BBVSP-pqc)," in *Proceedings of the 2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T)*, pp. 1–8, IEEE, Raipur, India, March 2022.

[29] A. Aldaej, T. A. Ahanger, M. Atiquzzaman, I. Ullah, and M. Yousufudin, "Smart cybersecurity framework for IoT-empowered drones: machine learning perspective," *Sensors*, vol. 22, no. 7, p. 2630, 2022.

[30] L. Allodi, T. Chotza, E. Panina, and N. Zannone, "The need for new antiphishing measures against spear-phishing attacks," *IEEE Security & Privacy*, vol. 18, no. 2, pp. 23–34, 2020.

[31] J. B. Awotunde, S. Misra, O. B. Ayoade, R. O. Ogundokun, and M. K. Abiodun, "Blockchain-based framework for secure medical information in internet of things system," in *Blockchain Applications in the Smart Era. EAI/Springer Innovations in Communication and Computing*, S. Misra and A. Kumar Tyagi, Eds., Springer, Cham, Switzerland, 2022.

[32] T. Ahamed Ahanger, A. Aldaej, M. Atiquzzaman, I. Ullah, and M. Yousuf Uddin, "Securing consumer internet of things for botnet attacks: deep learning approach," *Computers, Materials & Continua*, vol. 73, no. 2, pp. 3199–3217, 2022.

[33] S. Khezr, M. Moniruzzaman, A. Yassine, and R. Benlamri, "Blockchain technology in healthcare: a comprehensive Review and directions for future research," *Applied Sciences*, vol. 9, no. 9, p. 1736, 2019.

[34] P. Ratta, A. Kaur, S. Sharma, M. Shabaz, and G. Dhiman, "Application of blockchain and internet of things in healthcare and medical sector: applications, challenges, and future perspectives," *Journal of Food Quality*, vol. 2021, Article ID 7608296, 20 pages, 2021.

[35] S. S. Kushwaha, S. Joshi, D. Singh, M. Kaur, and H.-N. Lee, "Systematic Review of security vulnerabilities in ethereum blockchain smart contract," *IEEE Access*, vol. 10, pp. 6605–6621, 2022.

[36] A. Ghosh, S. Gupta, A. Dua, and N. Kumar, "Security of Cryptocurrencies in Blockchain Technology: State-Of-Art, Challenges and Future Prospects," *Journal of Network and Computer Applications*, vol. 163, Article ID 102635, 2020.