telecom

MDPI

Article

Phishing Detection in Blockchain Transaction Networks Using Ensemble Learning

Roseline Oluwaseun Ogundokun [1,2], Micheal Olaolu Arowolo [3], Robertas Damaševičius [4,*] and Sanjay Misra [5]

[1] Department of Computer Science, Landmark University, Omu Aran 251103, Nigeria
[2] Department of Multimedia Engineering, Kaunas University of Technology, 44249 Kaunas, Lithuania
[3] Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO 65211, USA
[4] Department of Applied Informatics, Vytautas Magnus University, 44404 Kaunas, Lithuania
[5] Department of Applied Data Science, Institute for Energy Technology, 1777 Halden, Norway
* Correspondence: robertas.damasevicius@vdu.it

Abstract: The recent progress in blockchain and wireless communication infrastructures has paved the way for creating blockchain-based systems that protect data integrity and enable secure information sharing. Despite these advancements, concerns regarding security and privacy continue to impede the widespread adoption of blockchain technology, especially when sharing sensitive data. Specific security attacks against blockchains, such as data poisoning attacks, privacy leaks, and a single point of failure, must be addressed to develop efficient blockchain-supported IT infrastructures. This study proposes the use of deep learning methods, including Long Short-Term Memory (LSTM), Bi-directional LSTM (Bi-LSTM), and convolutional neural network LSTM (CNN-LSTM), to detect phishing attacks in a blockchain transaction network. These methods were evaluated on a dataset comprising malicious and benign addresses from the Ethereum blockchain dark list and whitelist dataset, and the results showed an accuracy of 99.72%.

Keywords: blockchain; network security; phishing; attack recognition; deep learning

Citation: Ogundokun, R.O.; Arowolo, M.O.; Damaševičius, R.; Misra, S. Phishing Detection in Blockchain Transaction Networks Using Ensemble Learning. *Telecom* **2023**, *4*, 279–297. https://doi.org/10.3390/telecom4020017

Academic Editor: Lucia Seno

Received: 25 February 2023
Revised: 24 May 2023
Accepted: 26 May 2023
Published: 31 May 2023

1. Introduction

1.1. Background and Motivation

Blockchains are a transparent permanent public ledger of transaction records, independently verified and recorded, and available for inspection at any moment [1]. A great benefit of blockchains is that they are a form of decentralized information technology that could apply to many situations beyond its current most common use of cryptocurrency and financial asset management [2,3]. The growth and popularity of blockchain have sparked interest, especially in the financial sector. Most traditional blockchain applications focus on asset transfers and knowledge sharing through networks using smart contracts, suitable for commercial processes and business sectors, such as secure data sharing [4]. Initially, blockchain technology was positioned as an enabling mechanism for cryptocurrencies [5]. Later, its decentralized, distributed, immutable ledger capability attracted many e-Governance initiatives [6], resulting in its utilization in various public notary services, voting systems, citizen identity services, passport registration, and migration services [7]. Recently, also new applications have emerged, such as for internet-of-things (IoT) [8], smart home [9], industrial manufacturing [10], intelligent transportation systems [11], and vehicular networks [12]. Blockchain networks are decentralized, immutable, and provide high security, making them an attractive option for businesses and organizations. However, the security and privacy concerns related to sharing sensitive data on blockchain networks and the potential threat of attacks against blockchains have hindered their widespread adoption.

1.2. Blockchain Technology

Blockchain, a trust-free collaboration platform, was identified as an opportunity to build the next generation of cyber security systems. There are many ongoing research initiatives in this direction [13,14]. The smart contract [15] capability of blockchains over a distributed, decentralized data storage platform is perceived as a massive opportunity for implementing the decentralized IoT paradigm [16]. Many corporate entities and startups have already started investing in this compelling opportunity, and a few innovative products and services have surfaced. The financial products/services industry has also perceived blockchains with smart contracts as a huge opportunity to enable secure, trust-free micro/macro-financial transactions (see an illustration in Figure 1) with crypto/digital currencies instead of fiat currencies [17]. Many mission-critical tasks heavily dependent on data validity and verifiability, such as data management in clinical trials are also being perceived as a potential opportunity for applying blockchains with smart contracts. Blockchains with smart contracts are emerging as a high-potential platform with secure, trust-free, distributed, and decentralized data storage, enabling many novel technology business solutions [18,19].

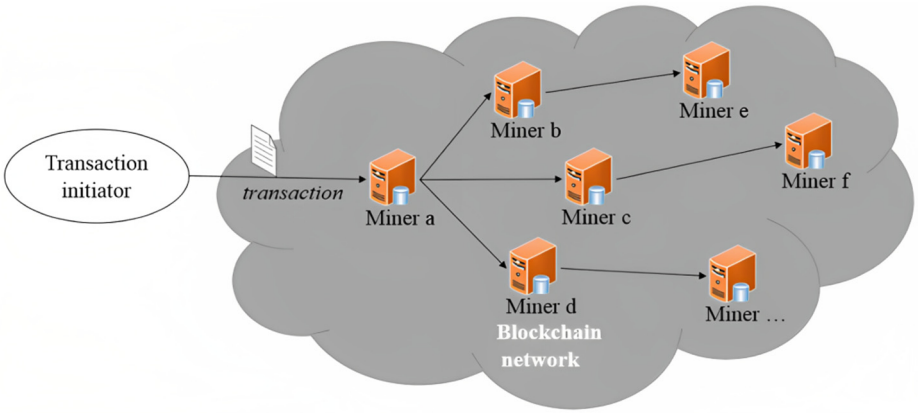

Figure 1. Transaction propagation in the blockchain.

Blockchain offers a feasible option to eliminate intermediaries, cutting operating costs, and boosting the efficiency of the sharing economy and e-governance, ultimately leading to the realization of the smart city concept [20]. The sharing economy [21] is characterized as an economic/social paradigm that broad segments of society may use to collectively use underutilized assets, in which supply and demand interact for the supply side to give products/services directly. The goal is to use underutilized assets better while lowering transaction costs [22]. Blockchain has the potential to significantly speed up and reduce the cost of transferring value-based digital assets across networks [23]. While enhancing the shared use of assets and resources has various good outcomes, such as energy savings and reduced urban congestion, artificial intelligence (AI) is still required to discover the most cost-effective solutions [24]. Real-time information and knowledge mining are critical for addressing wasteful asset use. Blockchain technology also contributes to sustainable development by facilitating a creative industries-based economy by adding value to artists, authors, bloggers, game developers, and anyone who wants to sell their unique items online [25]. The internet of value's (IoV) intelligent services are meta-products, a dedicated network of services, people, goods, and the environment nourished by the information flows enabled by the web and other ubiquitous technologies [26]. Apart from their entertaining value, such meta-products can be used to produce real value by sharing the anonymized data their sensors produced securely using blockchain.

The blockchain's main objective is to build a creditworthy environment amid autonomous members in an untrustworthy dispersed setting [27,28]. The bound blocks, peer-peer nodes, consensus-centered record processes, unidentified financial records, self-structured data proprietorship, and programmable smart contracts make a blockchain

system safe and autonomous [29]. Ongoing debates exist on which Bitcoin features are essential for blockchains and are only optional [30]. Furthermore, while blockchains offer an approved certified platform for data-level depository and processes, a dedicated acclaim structure is needed to make blockchain systems creditworthy. The need, effectiveness, and efficacy are all factors to consider [31]. When evaluating a blockchain venture's needs, viability, performance, and expected benefits are all important considerations. Several standard and capable blockchain procedures, such as blockchain trust, productivity, protection, confidentiality, management, and online-to-offline incorporation, are still in the initial phases of advancement.

1.3. Blockchain Challenges

Blockchain is a decentralized ledger technology that can fit multi-center trust-based cooperation well. Multiple blockchain-based IoT trust solutions have been proposed [32]. However, there are several issues with existing blockchain-based approaches. First, storing all data on the blockchain would be inefficient and difficult to handle; thus, not sharing data would lead to confidence issues; therefore, a compromise must be considered. Second, while research on blockchain-based trust has always concentrated on identity authentication and proof storage, there has yet to be any research on blockchain-based trust evaluation. Blockchain has been employed in various subsequent-generation implementations in recent years, and it has been a popular alternative for concerned parties in multiple businesses. This is due to the blockchain's support for trustless and shared implementations, in which data is distributed across the network in electronic-dispersed records [33]. Instead of a trustworthy broker, these online distributed ledgers allow decentralized application activity. Without the help of a trusted intermediary, untrustworthy entities may communicate in a peer-to-peer system and share data in a certifiable way using the blockchain [34]. To resolve the single point of failure issue in the IoT system, each fog node will negotiate with one another by employing blockchain instead of depending on a fundamental cloud expert.

With the explosion of blockchain applications in various fields, attacks on blockchains are also increasing, such as double-spending attacks, eclipse attacks, dust transaction attacks, and so on [35,36]. Smart homes, power grids, culture, housing, and health care are all examples of where blockchain is expected to perform a critical function in upcoming technological developments, and its use will likely skyrocket in the coming years [37]. The topic of blockchain security has arisen because of the growing quantity of blockchain users and the vast volume of data connected with them [38]. Privacy-preserving predictive modeling using data from blockchain networks can accelerate research and facilitate quality improvement initiatives [39,40]. The main requirements are to protect the privacy of individuals and sensitive personal information, work online (in real-time), and be based on the decentralized architecture of the blockchain-based data-sharing network.

1.4. Attacks on Blockchain

Phishing attacks are one such security concern observed in blockchain transactions. Phishing is a malicious technique attackers use to gain sensitive information, such as usernames, passwords, and private keys, from unsuspecting victims [41]. Phishing attacks can be carried out via email, messaging apps, or social media platforms, leading to financial loss, identity theft, and other serious consequences [42]. For example, fraudsters are taking advantage of flaws in major search engines' ad presentations to drive consumers to phishing sites. In contrast to a common strategy of sending emails to everyone, links to phishing sites in search engine advertising may be more convincing, especially if the domain they are targeting is listed as the destination. To guard against phishing, users' credentials are actively protected, which involves storing password hashes rather than passwords. Even though many firms spend large sums of money researching phishing and developing unique tools and algorithms for its detection and blocking, no products provide complete protection from such assaults. Because phishing attacks are typically carried out with the victim's involvement, anti-phishing defense uses technical and social engineering tools [43].

Phishing detection has been intensively researched in recent decades, with several approaches presented [44,45]. However, despite the properties of blockchain, there has been little study on phishing fraud detection. Andryukhin [46] classifies the significant types and schemes of phishing attacks on the blockchain project and proposes ways of phishing attack defense from the blockchain project's perspective. Unlike them, we are focusing overall blockchain ecosystem and alerting users to phishing schemes as soon as they appear. AI and deep learning methods can be applied to detect network intrusions [47,48] and recognize malware and botnet attacks [49–51]. To improve recognition accuracy, ensemble learning architectures have been successfully adopted [52–54].

1.5. Aims, Novelty, and Contribution

The motivation for this paper is to propose a deep-learning-based approach to detect phishing attacks in blockchain transaction networks. The use of deep learning methods, such as long short-term memory (LSTM), bi-directional LSTM (Bi-LSTM), and convolutional neural network LSTM (CNN-LSTM), can help to identify and prevent phishing attacks in real time. Deep learning is increasingly recognized as a powerful tool for advancing big data technologies, including those related to the internet of things (IoT). However, developing effective deep learning models for IoT applications requires addressing several critical issues, such as single points of vulnerability, privacy leakage, lack of usable knowledge, and data poisoning attacks. In addition, phishing scams are a severe threat to the financial security of users in blockchain ecosystems. To address this issue, this paper proposes a systematic approach to detecting phishing scams in the Ethereum ecosystem using an improved ensemble-based LSTM architecture combining deep learning and blockchain technology. The proposed system aims to provide possible solutions for mitigating phishing scams in IoT networks by integrating ensemble learning techniques, such as convolutional and recurrent neural networks, with blockchain and deep learning.

The novelty of this study lies in the proposal and evaluation of a deep learning-based approach to detect phishing attacks in blockchain transaction networks. Specifically, the use of long short-term memory (LSTM), bi-directional LSTM (Bi-LSTM), and convolutional neural network LSTM (CNN-LSTM) for detecting phishing attacks in real-time on blockchain networks is a new and innovative application of deep learning techniques.

The contributions of this study are summarized as follows:

- A novel deep-learning model for detecting phish scams in blockchain transactions is presented.
- Using Bi-LSTM, CNN-LSTM, and embedded-LSTM on the Ethereum transaction network dataset is demonstrated.

A dataset of malicious and benign addresses from the Ethereum blockchain dark list and whitelist dataset is used to evaluate the proposed approach. The evaluation results demonstrate the proposed approach's effectiveness in detecting phishing attacks, which can contribute to developing more secure and trustworthy blockchain-based systems.

The rest of the article is arranged as follows. Section 2 discusses a literature review on the security of blockchains. The methods used to implement this study are presented in Section 3. Section 4 discussed the findings discovered, and the outcomes deduced from the implementation of the proposed system. The paper is concluded in Section 5, and future work is discussed.

2. Literature Review

In comparison to centralized database systems, blockchain-centered depository systems use stock data in an extra stable and accessible means. The user's personal information, records, or system-wide information are examples of stored content. The following sections address several proposals for improving emerging blockchain-centered depository systems and employing blockchain technologies to strengthen prevailing centralized structures. This section also discussed the related works on phishing scams in blockchain transactions.

2.1. Blockchain Technology and Machine Learning

Blockchain technology [55] promises to change business operations in both the commercial and public sectors. It provides a method for securing processed transactions in distributed and decentralized systems while ensuring transparency and immutability [56,57]. Nonetheless, security, scalability, interoperability, and regulation are all issues that must be addressed when using technology. On the other hand, machine learning (ML) applications have risen in popularity recently due to the abundance of data and the potential of ML algorithms to enable systems to learn and improve automatically using historical data [58,59]. Blockchain technology can benefit from using ML algorithms because of their capacity to analyze large amounts of data. It can considerably improve the security of such systems [60–63]. Anomaly, fraud recognition, malignant activity detection, biometrics monitoring, illness detection, and other applications that use blockchain technology and ML methods are rapidly growing in many domains, such as healthcare, finance, and energy, and for various objectives. In recent years, several reviews have addressed these issues. Some of them deal with combining blockchain technology with AI and other technologies to achieve decentralized authentication [60], to enable different features within 5G networks [62], or to de-anonymize bitcoin addresses or recognize entities within cryptocurrency transaction networks [10].

A framework with a keyword search service can solve the search capacity in storage blockchains [64]. Both data and listed keywords will be encoded afore time being sent to the blockchain nodes (depository providers) in this proposed scheme, much as in different blockchain systems, and data will be consigned to the blockchain depository providers to be processed. Even though keywords are stored on the blockchain, service providers will issue permissions to the network's other nodes. Subsequently, data owners and authorized nodes will use the blockchain to scan for data. Ramachandran et al. [33] also suggested a scheme to deter data theft. A meta-data set containing data origin, data holders, and data transformations should be registered in this structure. It is thought that considering the rest of the people on this system are trustworthy, spiteful data alteration will be avoided. Attribute-centered encoding is one of the suggested solutions to the issue of conventional cloud storage services' data protection (ABE). This approach, however, has a flaw: the private key generator's ability to encrypt the info.

Traditional cloud computing services face the danger of a solitary point of failure simply because of their clustered existence [65–68]. Wang et al. [34] propose a blockchain-centered architecture for decentralized database networks with fine-grained entrée control. The open database server global file structure, the Ethereum network, and ABE technologies are all included in this model. The data owner may define the admission policies for further nodes in this platform, and keyword search on encoded data is also available.

2.2. Blockchain-Based Approach for IoT

Several studies have been conducted on blockchain-based approaches for IoT networks, and some of them are discussed as follows. Dorri et al. [39] suggested a classification approach based on the ML techniques applied to data stored in a blockchain network, which intends to improve IoT protection by identifying unauthorized machines. Putra et al. [27] postulated a trust protection protocol for ensuring safe and trustworthy access control in a decentralized IoT system and detecting and eliminating malicious and corrupted nodes. Some developers also recommended using TrustChain, a three-layered trust management system built on consortium blockchain to monitor supply chain member experiences and automatically assign trust and credibility ratings based on these interactions [28]. Researchers also proposed a novel blockchain-built architecture for offering a private and stable communication model for smart vehicles, ensuring that the data they receive is provided by a trustworthy node [69]. Smart grids utilize machine learning techniques to analyze large data sets, extract valuable insights, identify potential cyber-security threats, and safeguard the data [70]. Anthi et al. [71] projected a 3-layer intrusion detection system (IDS) that used a supervised learning approach of ML to differentiate between spiteful and

benevolent system operations as well as to perceive system-built cyberattacks like DoS, MITM/Spoofing, replay, and reconnaissance, including to perceive a multi-phase bout on IoT systems. Wei et al. [72] recommended an ML-built spiteful app identification gadget that employs a naïve Bayesian J48 decision tree as a classification technique to identify spiteful apps in Android gadgets in real time. To cope with the issue of unpredictable deeds and conglomeration of IoT networks, the authors implemented an autonomous and adaptive monitoring mechanism by employing ML and software-defined networking (SDN) for their IoT protection platform [73]. Diverse ML techniques that can be exploited for data produced in IoT system-built surroundings like smart cities to extract higher-level knowledge were similarly presented [74].

2.3. Fraud Detection on the Blockchain Network

In [63], Pham and Lee look at the challenge of detecting dubious users and transactions using two graphs built from the Bitcoin network. Users are represented as nodes in the first graph, while trades are represented as nodes in the second. They employed Mahalanobis distance, k-means clustering, and support vector machine (SVM) approaches to detect aberrant activity.

Currently, no solutions aim to incorporate deep learning methods into the blockchain transaction phishing scam detection process to enable automatic phishing scam blockchain transaction detection while also actively detecting possibly fraudulent transactions subject to the user's phishing attacks. Even though the work on identifying abnormalities among transactions was done, none of the techniques were examined to adapt to the address of the transaction patterns, and none of the methods evaluated the strategy of utilizing labeled user data (anomalies are not known). The suggested approach is unique because it can be used with any network address with a good transaction history, independent of transaction patterns, and can identify abnormal transactions. Several models have been proposed by different researchers in this field; however, working with LSTM has shown outstanding results, and this study suggests its approach as a novelty further to enhance the detection of phishing spam in blockchain transactions.

2.4. Phishing Scam-Related Studies in Blockchain Transactions

On the Ethereum blockchain, Kumar et al. [75] developed a technique for detecting fraudulent accounts. Malicious and non-malicious addresses were included in the data collection. A string comparison was utilized to filter out duplicate addresses regardless of case sensitivity. Addresses that had null transactions were also removed from the data set. Their method compared KNN, decision tree, RF, and XGBoost supervised learning algorithms. Dalal and Abulaish [76] suggested a multilayer perceptron design for detecting cryptocurrency dishonesty. The data for their assessment was gathered from the CMC website and classified as either lawful or fraudulent. Linear regression, Softmax regression, SVM, and MLP were used in the analysis, while MLP was shown to be the most accurate. Yuan et al. [77] suggested an improved graph classification technique (Graph2Vec) for phishing detection on the Ethereum blockchain. They took phishing addresses from etherscan.io and added the same number of legitimate addresses to generate data collection. They gathered the transactions for each lesson, deleted the duplicate data, and excluded addresses with fewer than ten transactions and more than 300 transactions. They compared the results to node2vec, WL-kernel, and Graph2Vec, among other approaches. Chen et al. [78] introduced a phishing scam detection method for the Ethereum blockchain. They employed a convolutional graph network and an autoencoder to detect phishing accounts. They utilized the Ethereum transaction history for data collection. They compared the performance of their GCN approach, DeepWalk, Node2Vec, and LINE methods. Chen et al. [79] employed ML methods to identify Ponzi schemes on the Ethereum blockchain. They gathered smart contract source code from etherscan.io and manually evaluated whether they were Ponzi scheme contracts to produce data collection. The characteristics were extracted without the course code, along with all linked and failed

transactions. The contracts were then translated from bytecode to opcode, and the features were categorized before feature extraction. XGBoost was utilized by Chen et al. [80] to find Ponzi schemes on the Ethereum blockchain. They used etherscan.io to gather smart contracts to test their technology. The frequency of the bytecodes was determined after they were translated into opcodes. Ponzi and non-Ponzi contracts were labeled.

3. Methods

This section describes the material, such as the dataset used to test the proposed system. The method employed deep learning classifiers discussed in this section.

3.1. Dataset

As a dataset, we use a repository for maintaining lists of things in the Ethereum transaction network available on GitHub [81]. The dataset has 2407 addresses, of which 2373 are malicious (from the URLs-dark list), and 34 are harmless addresses (from the URLs-light list). As such, the dataset is highly imbalanced.

The "Ethereum-lists" dataset is a collection of curated lists related to Ethereum and the Ethereum blockchain. The purpose of this dataset is to provide various types of information that can be used by developers, researchers, and enthusiasts working with Ethereum. Here, is a breakdown of the dataset and its contents:

- Tokens List: This list contains information about different tokens built on the Ethereum blockchain. It includes token names, symbols, addresses, decimals, and additional attributes.
- Contracts List: This list focuses on smart contracts deployed on the Ethereum blockchain. It provides information about the contract address, ABI (application binary interface), and other relevant contract details.
- Addresses List: This list includes Ethereum addresses associated with specific entities or projects. It includes addresses of known wallets, exchanges, dApps (decentralized applications), and other relevant Ethereum participants.
- ENS (Ethereum name service) List: ENS is a decentralized domain name system built on the Ethereum blockchain. This list contains ENS domain names and their corresponding Ethereum addresses.
- Airdrops List: Airdrops refers to the distribution of free tokens to the Ethereum community. This list provides information about past and upcoming airdrops, including details about the airdrop project, token, and distribution methods.
- ENS Reverse Resolution List: This list is a reverse lookup for ENS domain names. It maps Ethereum addresses back to their corresponding ENS domains.

The dataset is provided in JSON format, making it easy to parse and integrate into various Ethereum-related applications and services. By providing this dataset, the creators aim to offer a comprehensive and reliable resource for Ethereum-related information, ensuring accuracy and facilitating the development of applications and research within the Ethereum ecosystem.

3.2. Proposed Framework

As input, we use Blockchain transaction addresses which are preprocessed and transformed into the numerical domain using word embedding. The classification uses three types of neural network models (LSTM, BI-LSTM, and CNN-LSTM). We have selected Bi-LSTM over other alternatives such as GRU (Gated Recurrent Unit) because Bi-LSTM is generally considered better than GRU in tasks where modeling long-term dependencies is important. This is because Bi-LSTM has a more complex architecture that allows it to remember information from earlier in the sequence for longer periods. Additionally, the bidirectional aspect of Bi-LSTM allows it to capture dependencies in both directions, which can be especially useful in tasks where the ordering of the data is important. Finally, the results are aggregated using ensemble voting to produce the final prediction (malicious or benign transaction).

3.3. Word Embedding

Word embedding refers to representing words in a high-dimensional vector space, where each word is assigned a unique vector that captures its semantic and syntactic properties. The main idea behind word embedding is that similar words should have identical vector representations and that the distance between two vectors can be used to measure the similarity between their corresponding words. Word embedding typically starts with a neural network used to learn the vector representations of each word in the training corpus. The neural network is trained to predict the context in which each word appears based on its neighboring words. The resulting vector representations capture the meaning of each word based on the context in which it is used and can be used as input to many natural languages processing tasks, such as sentiment analysis, machine translation, and information retrieval.

Here, we used word embedding with an external neural network to train to connect words with their context, yielding a set of numeric vectors with the necessary dimensions.

3.4. Deep Convolutional Neural Networks

We discuss long short-term memory (LSTM), bidirectional LSTM (Bi-LSTM), and convolutional neural network (CNN)-LSTM in the subsections below.

3.4.1. Long Short-Term Memory (LSTM)

Long short-term memory (LSTM) is a type of architecture for recurrent neural networks (RNNs) that, unlike ordinary RNNs, can learn long-term dependencies. Unlike traditional RNNs, each neuron in them is a memory cell whose contents may be changed or reset. Figure 2 depicts a schematic representation of the LSTM memory cells. Like feedforward networks, recurrent networks have input, hidden, and output layers. LSTM employs a more complicated calculation approach that uses three filters: input filter i, forget filter f, and output filter o. Let us imagine there are K, M, and L neurons in the input, hidden, and output layers, respectively.

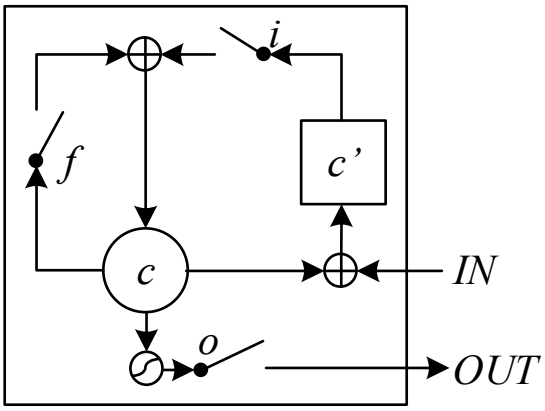

Figure 2. Schematic representation of LSTM architecture: *IN* are inputs, *OUT* are outputs, i is input filter, f is forget filter, o is output filter, c and c' are memory cells.

The activation vectors of these layers are given in Equations (1)–(3), respectively.

$$x[n] = \{x_1[n],\ x_2[n], \ldots, x_K[n]\}^t \tag{1}$$

$$h[n] = \{h_1[n],\ h_2[n], \ldots, h_M[n]\}^t \tag{2}$$

$$y[n] = \{y_1[n],\ y_2[n], \ldots, y_L[n]\}^t \tag{3}$$

where t is the transposition operation, the weights of the neuron layers are:

$$w^{in} = \left(w_{ij}^{in}\right)_{M\times K},\ w^{h} = \left(w_{ij}^{h}\right)_{M\times M},\ w^{out} = \left(w_{ij}^{out}\right)_{L\times M} \tag{4}$$

In the direct transfer mode, using Equations (5) and (6), the value of activation of the neuron j of the hidden level is obtained.

$$h'_j[n] = \sum_{k=1}^{K}\left(w_{jk}^{in} \times x_k[n]\right) + \sum_{m=1}^{M}\left(w_{jm}^{h} \times h'_m[n-1]\right), \tag{5}$$

$$h_j[n] = \theta_j\left(h'_j[n]\right) \tag{6}$$

where θ_j is the nonlinear activation function of neuron j in the hidden layer. The value of the output neuron y_l is calculated using Equation (7).

$$y_l[n] = \sum_{m=1}^{M}\left(w_{lm}^{out} \times h_m[n]\right) \tag{7}$$

3.4.2. Bi-Directional Long Short-Term Memory (Bi-LSTM)

Bidirectional LSTM (Bi-LSTM) is an extension of the LSTM network in which a sequence from beginning to finish is supplied to the network simultaneously. bidirectional LSTMs have the potential to increase model performance in sequence classification challenges. Bidirectional LSTMs train two LSTMs instead of one in the input sequence in instances where all time slots of the input sequence are accessible. The first refers to the original input sequence, while the second refers to an inverted replica of the original input sequence. This can offer the network extra context, allowing for faster and more thorough problem analysis. The memory unit, which can keep its state over time, and nonlinear gate units, which control the network's information input and output flow, are the components of the LSTM architecture. Considering the insights gained from secure networks, it is believed that because the LSTM neuron comprises internal cells and gate units, it is necessary to look not only at the neuron's output but also at its internal structure when designing new features for LSTM.

The Bi-LSTM architecture used in this paper has three LSTM layers, two feedforward layers, and a SoftMax layer for predictions. This fully connected architecture enables us to capitalize on the inherent correlations between connections. Co-occurrence exploration is used on the relationship before the second layer of the network to learn from the input features. Finally, backpropagation is used to allow learning in the LSTM layer.

3.4.3. CNN-LSTM

A convolutional neural network (CNN) is a deep learning model that uses convolution and pooling processes. A typical CNN model has one or more layers of subsampling and convolution, followed by fully connected layers that can be one or more and an output layer.

CNN may utilize max-pooling layers. Max pooling performs nonlinear down-sampling that minimizes the longitudinal size of the conventional layer's output using the production subdivision in rectangular boxes, which is the best value for each filter. By bringing down the quantity of connection and computational cost, max-pooling lessens overfitting by making features more spatially accessible. Max pooling layers can be numbered after every convolutional layer or after some layers.

Using dropout, during the training process, randomly selected neurons are ignored. The operation means that we "drop out" some neurons randomly. In easy words, the contribution of neurons is temporarily removed on the forward pass during the activation of downstream neurons, and no weight updates are done to neurons on the backward pass.

The next layer used is the batch normalization layer. It normalizes each layer's inputs to overcome covariate shift problems. We do normalization for an input layer by changing the activations to speed up the learning process. During our training process, the batch normalization, we do the following tasks:

First, we compute the batch mean (μ_B) and batch variance (σ_B^2) of input layers as follows:

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{8}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \tag{9}$$

The normalization of input layers takes place using previously calculated values.

$$\overline{x}_1 = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{10}$$

Lastly, we shift and scale the input to obtain the layer's output as follows.

$$y_i = \sqrt{x_i} + \beta \tag{11}$$

3.5. Parameter Settings

We chose the values of hyperparameters by first examining training data and then looking for the optimal hyperparameters. We decided on the optimizer, activation unit, and dropout that gave our model the best results. The settings are adjusted differently for each dataset. On this dataset, we experimented with several stages. We selected the locations that performed the best and produced the best outcomes. Table 1 depicts the parameterization of variables for our model. When developing our network, we employed the ReLU activation unit, commonly used as an activation unit in many deep-learning models. We used several optimizers to train our model and evaluated its performance. On our dataset, the nAdam optimizer provided the highest accuracy. The dropout value was set to 0.2 and applied to all layers. The learning rate parameter was set to five, indicating that the model waited for five epochs before decreasing the learning rate.

Table 1. Hyperparameter values for the CNN model.

Parameter	Value
Num of filters (CNN)	16
Filter length (CNN)	5
Max Sequence Length	32
Batch size	64
Epochs	100
Loss	Binary cross entropy
Optimizer	nAdam
Activation unit	ReLU -Rectified linear units
Dropout	0.2
LR patience	5

Each network's initial layer is an embedding layer with varying embedding size parameters. The embedding layer has an output size of 200, and the following layer after embedding is spatial dropout with a value of 0.2. The last layer is wholly integrated with the Softmax activation algorithm. The measure is accuracy, the loss function is categorical cross-entropy, the optimizer is Adam with default values, and the loss function is categorical cross-entropy. The hyper-parameters for the Bi-LSTMs are shown in Table 2. The optimization algorithm (optimizer) trains the neural network by minimizing the sum of errors between predicted and actual values, also known as the cost function. The learning

rate determines how frequently the weights are updated concerning the estimated error. Dropout is a regularization technique that reduces overfitting by dropping network nodes during training. The batch size specifies the number of instances to be propagated before the model's parameters are updated. The number of layers represents the number of LSTM cells, and the number of hidden states represents the number of hidden states. Abstraction of features increases proportionally to the hidden conditions over time.

Table 2. Hyperparameter values for the Bi-LSTM model.

Parameters	Tested Values	Bi-LSTM
Input features	16, 32, 64	32
Hidden size	32, 64, 128, 256	64
Number of layers	2, 3, 4	3
Batch size	16, 32, 64	32
Dropout	0.2, 0.3, 0.4, 0.5, 0.6, 0.7	0.3
Learning rate	0.00001, 0.0001, 0.001	0.0001
Optimizer	Adam, nAdam, SGD	Adam

The values of the model architecture hyper-parameters are summarized in Table 3.

Table 3. Selected values of architectural hyper-parameters for the models.

Model	Parameter Values
LSTM	Hidden layers = 128 Dropout = 0.3 Recurrent dropout = 0.3
Bidirectional LSTM	Hidden layers = 128 Dropout = 0.3 Recurrent dropout = 0.3
CNN + LSTM	Number of convolution filters = 512 Kernel size = 3 Activation function = RELU Hidden layers = 128 Dropout = 0.3 Recurrent dropout = 0.3

3.6. Ensemble Voting

Ensemble learning is a technique used in machine learning that involves training multiple imperfect models, also known as "weak learners," to perform the same task. Then, their results are combined to achieve a better outcome. These weak learners are then used to build more complex models by combining multiple them. Typically, these base models do not perform independently due to either bias or variability, making them less robust. Ensemble techniques address this issue by merging multiple models to create a strong learner with improved performance. More accurate and reliable models can be generated by adequately combining weak models. A stacking ensemble model is designed using base models and a meta-learner, the final stage classifier that utilizes the predictions made by the base models (see Algorithm 1). The base models are trained on the training data and used to generate predictions, which are then used as input for the meta-learner. The meta-learner is trained on the base model predictions using new, unseen data, to aggregate the base model predictions and make the correct output prediction. This is achieved by feeding the meta-learner with input and output pairs of data from the base learners. Ensemble classification models are potent machine learning methods that have the potential to perform exceptionally well and generalize well to novel, new datasets. An ensemble classifier's benefit is that combining the predictions of several classifiers may correct for mistakes produced by any of them, improving total accuracy.

Algorithm 1. Ensemble stacking algorithm. Ensemble learning algorithm

Input: Training dataset D, base learners $B = \{B_1, B_2, \ldots, B_m\}$, meta-learner L.
Output: Ensemble model predictions.
Stage 1: Construct an ensemble of base models:

1 Select base learners B such that $B_i \neq B_j$ for $i \neq j$.
2 Select a meta-learner L.

Stage 2: Train the ensemble:

3 For i = 1 to m do.
4 Train base model B_i on training dataset D.
5 Cross-validate base model B_i.
6 End for.
7 Combine the predictions from the base models to form a new training dataset $D' = \{(X_{tr}, B_1(X_{tr}), B_2(X_{tr}), \ldots, B_m(X_{tr}))\}$.
8 Train the meta-learner M on the new dataset D'.

Stage 3: Test the meta-learner on new data:

9 Record output decisions from the base models B.
10 Feed base model decisions into the meta-learner M to make the final decision.

3.7. Performance Evaluation

Performance metrics employed in this study include accuracy, precision, specificity, sensitivity, Kappa, and F1 score. There are four terms associated with the confusion matrix: *true positive (TP)*, *true negative (TN)*, *false positive (FP)*, and *false negative (FN)*. The formula for the metrics is stated thus:

$$Sensitivity = \frac{TP}{TP + FN} = True\ Positive\ Rate \tag{12}$$

$$Specificity = \frac{TN}{TN + FP} = True\ Negative\ Rate \tag{13}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

When the number of *TPs* is small compared to *TNs*, precision can be calculated.

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

4. Experimental Results

4.1. Implementation and Dataset

All computations were implemented using Python on an Intel® Core i5-8265U 64bit processor with 8GB RAM and Windows 10 Home operating system. The suggested technique was built-in Keras using a TensorFlow library. The model was trained over 100 epochs with a patience value of five. If the performance does not improve, the procedure is scheduled to cease. The dataset is divided into 70/30 (training/validation) sets. After each period, the model is assessed. In this manner, the model with the highest validation set score is utilized to create predictions for testing data.

4.2. Results

The confusion matrices of the classification results are presented in Figure 3 and summarized in Table 4. The classification performance measures are summarized in Table 5. A comparison with related works is shown in Table 4. The best result was achieved using an ensemble of embedding LST, CNN-LST, and Bi-LSTM classifiers, which earned an accuracy of 0.9988, and an F-score of 0.9985.

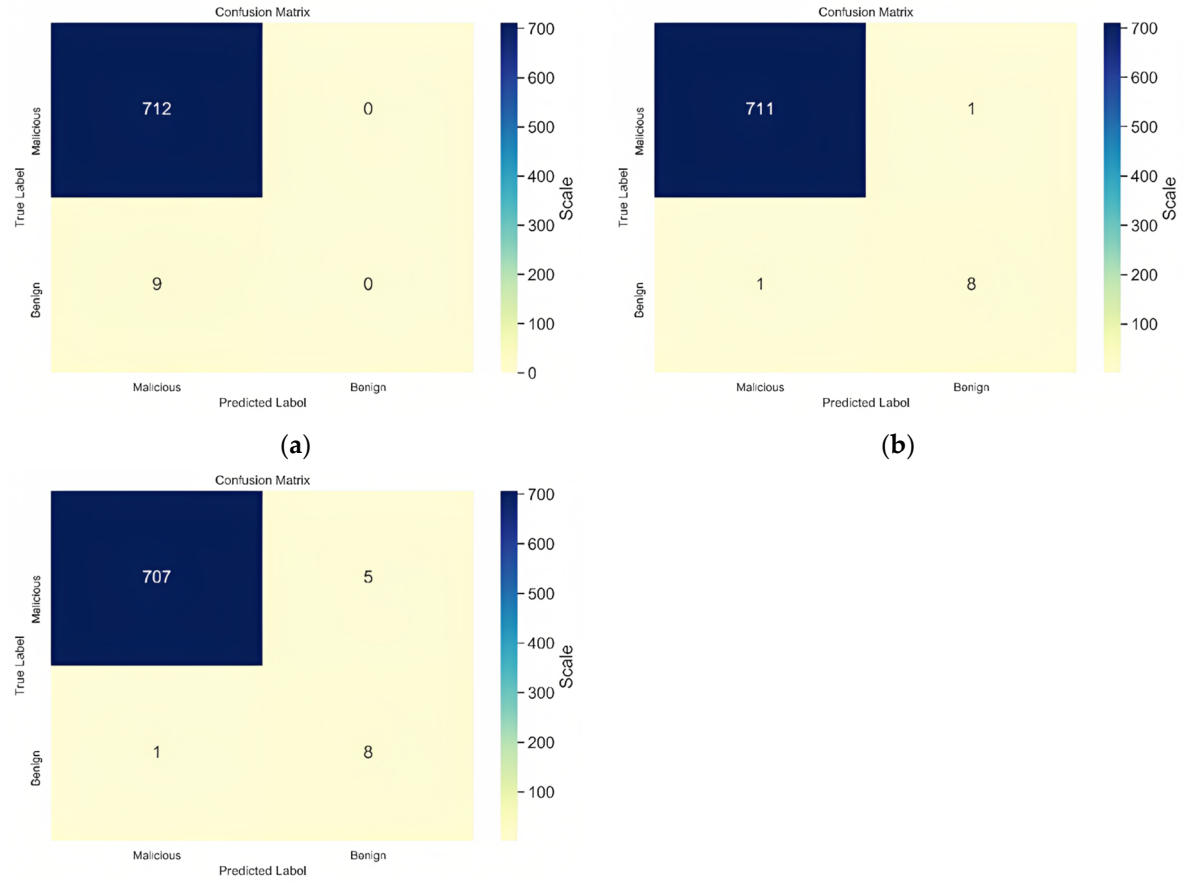

Figure 3. Confusion matrices for (**a**) Bi-LSTM, (**b**) CNN-LSTM, and (**c**) embedding LSTM.

Table 4. Confusion matrices for the classifiers.

Classifiers	***TP***	***TN***	***FP***	***FN***
Bi-LSTM	712	0	9	0
CNN-LSTM	711	8	1	1
Embedding LSTM	707	8	1	5

Table 5. Comparative analysis between the classifiers. The best results are shown in boldface.

Classifier	**Accuracy**	**Sensitivity (Recall)**	**Precision**	**F-Score**
Embedding LSTM	98.75%	100%	98.75%	99.37%
CNN-LSTM	98.75%	99.86%	98.89%	99.37%
Bi-LSTM	99.17%	99.86%	99.30%	99.58%
Ensemble	**99.72**%	**99.86**%	**99.86**%	**99.86**%

Table 5 shows the comparative analysis between the classifiers. The ensemble classifier outperformed the other three classifiers with high metrics of 99.72% accuracy, 99.86% recall, precision of 99.86%, and f-score of 99.86%. Using the deep learning ensemble technique for phishing scam detection allowed for better methods, thus increasing the classification accuracy.

4.3. Comparison with Existing Methodologies

Phishing detection in blockchain transaction networks is a crucial task to protect users from malicious activities and ensure the security of their transactions. There are several methodologies and approaches that can be used to detect phishing attempts in blockchain transaction networks. Let's discuss and compare some of these methodologies:

- The blacklist-based approach involves maintaining a blacklist of known phishing addresses or patterns. Phishing addresses or patterns are added to the blacklist based on historical data or user reports. When a new transaction occurs, it is checked against the blacklist, and if a match is found, the transaction is flagged as potentially malicious. However, this approach relies on the availability and accuracy of the blacklist, which can be challenging to maintain and may not cover all possible phishing attempts.
- The heuristics-based approach utilizes predefined heuristics or rules to identify potential phishing transactions. These heuristics can include unusual transaction patterns, high gas fees, suspicious addresses, or known phishing indicators. Phishing attempts are flagged based on the violation of these heuristics. While heuristics can be effective in detecting some phishing attempts, they may also generate false positives or miss new and evolving phishing techniques.
- The machine-learning (ML)-based approaches leverage algorithms and models trained on historical data to detect phishing attempts. Features such as transaction patterns, transaction metadata, address reputation, and network behavior are extracted, and ML models are trained to classify transactions as phishing or legitimate. ML models, such as decision trees, random forests, or neural networks, can make predictions based on these features. ML-based approaches have the advantage of adapting to new phishing techniques by continuously retraining the models. However, they require a significant amount of labeled training data and may have difficulty handling adversarial attacks aimed at bypassing the detection models.
- The consensus-based approach involves utilizing the consensus mechanism of the blockchain network to detect phishing attempts. By analyzing the behavior of nodes in the network and comparing their transaction validation results, discrepancies or suspicious behavior can be identified. Nodes that consistently provide incorrect validation results or exhibit malicious behavior can be flagged as potential phishing nodes. This approach relies on the assumption that a majority of the network nodes are honest and can be challenging to implement in networks with a low number of participating nodes.

The effectiveness of each methodology can vary depending on the specific characteristics of the blockchain network, the nature of phishing attempts, and the available resources. Combining these methodologies, such as machine learning for initial detection and heuristics-based rules for rule-based validation, can provide a more robust phishing detection system. Regular updates, continuous monitoring, and collaboration between researchers, developers, and users are essential to staying ahead of emerging phishing techniques in blockchain transaction networks.

Ensemble learning is a powerful methodology that combines multiple models to improve the overall performance and robustness of a system. Applying ensemble learning to phishing detection in blockchain transaction networks can enhance the accuracy and effectiveness of the detection process. We compare ensemble learning-based methodologies with the previously discussed methodologies for phishing detection in blockchain transaction networks as follows:

- Ensemble learning can be applied by combining multiple machine learning models, such as decision trees, random forests, or gradient-boosting algorithms. Each model in the ensemble is trained on different subsets of the data or with different feature representations. The outputs of individual models are combined, either through majority voting or weighted averaging, to make the final prediction. Ensemble learning can improve detection accuracy by leveraging the strengths of different models and

reducing the impact of individual model weaknesses. It can also help mitigate false positives and false negatives, leading to more reliable phishing detection in blockchain transaction networks.

- Ensemble learning can also be applied by combining different methodologies discussed earlier, such as combining blacklisting, heuristics, and machine learning-based approaches. Each methodology can contribute unique strengths to the ensemble, leading to a more comprehensive and robust phishing detection system. For example, the outputs of blacklisting, heuristics, and machine learning models can be combined using ensemble techniques to make the final decision. This approach helps leverage the complementary nature of different methodologies and enhance the overall accuracy and effectiveness of phishing detection.
- Ensemble learning can be made adaptive by continuously monitoring the performance of individual models or methodologies and dynamically adjusting their contributions to the ensemble. This adaptability allows the system to respond to changes in the phishing landscape and quickly incorporate new detection techniques or update existing models. By combining ensemble learning with adaptive mechanisms, the phishing detection system can stay updated with evolving phishing techniques and improve its resilience against emerging threats in blockchain transaction networks.

Ensemble learning-based methodologies offer several advantages for phishing detection in blockchain transaction networks. They can improve detection accuracy, handle a wide range of phishing techniques, and adapt to changing attack patterns. However, it's important to consider the challenges associated with ensemble learning, such as the need for diverse and high-quality models, potential overfitting, computational complexity, and the requirement for continuous model maintenance and updates.

In summary, ensemble learning-based methodologies for phishing detection in blockchain transaction networks provide an effective approach to enhance the accuracy and robustness of the detection system by combining multiple models or methodologies. The choice of ensemble technique and the composition of the ensemble depend on the specific requirements and characteristics of the blockchain network, the available data, and the resources at hand.

4.4. Comparison with Existing Studies

We compared our findings to previous studies investigating phishing scans using ML techniques to validate our proposed method, as shown in Table 6. The proposed model gives better accuracy, recall, precision, and f-score results.

Table 6. Comparative analysis with existing studies.

References	Method	Scam	Accuracy	Recall	Precision	F-Score
[75]	XGBoost	Malicious users' detection	96.54%	——	——	——
[76]	Multilayer perception	Cryptocurrency deception	98.00%	——	98.98%	——
[77]	Graph2Vec	Phishing	——	77.00%	69.00%	73.00%
[78]	GCN	Phishing	——-	14.53%	72.94%	23.57%
Proposed method	**Ensemble**	**Phishing**	**99.72%**	**99.86%**	**99.86%**	**99.86%**

It is shown in Table 6 that researchers [77,78] did not use accuracy for their system performance evaluation. Researchers [75,76] did not use recall, f-score, and AUC for their performance evaluation. Comparing using the five metrics used by our study, we can see that our proposed system outperformed that of the existing studies with an accuracy of 99.72%. This comparison with existing studies shows that our proposed system improved classification accuracy with low misclassification errors.

5. Conclusions

In conclusion, this research article proposed a deep learning-based approach for detecting phishing attacks in blockchain transaction networks. The study used long short-

term memory (LSTM), bi-directional LSTM (Bi-LSTM), and convolutional neural network LSTM (CNN-LSTM) to detect phishing attacks in real-time on blockchain networks. The proposed system integrated ensemble learning techniques, such as convolutional and recurrent neural networks, with blockchain and deep learning to address low precision, latency, stability, and privacy issues. The evaluation results demonstrate the proposed approach's effectiveness in detecting phishing attacks, which can contribute to developing more secure and trustworthy blockchain-based systems. The novelty of this study lies in the proposal and evaluation of a deep learning-based approach to detect phishing attacks in blockchain transaction networks, a new and innovative application of deep learning techniques.

Future research can focus on developing more effective and knowledgeable deep learning algorithms to improve the system's performance. Additionally, an improved architecture that employs deep learning principles, such as feature extraction, scaling, and classification, can be suggested in a decentralized medium to address these concerns.

Author Contributions: Conceptualization, R.D., R.O.O. and S.M.; Formal analysis, R.O.O. and M.O.A.; Investigation, R.O.O., M.O.A. and R.D.; Methodology, R.D. and S.M.; Software, R.O.O. and M.O.A.; Supervision, S.M.; Validation, R.O.O. and M.O.A.; Visualization, R.D.; Writing—original draft, R.O.O. and M.O.A.; Writing—review and editing, R.D. and S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset used in this study is available from: https://github.com/MyEtherWallet/ethereum-lists (accessed on 1 February 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Malla, T.B.; Bhattarai, A.; Parajuli, A.; Shrestha, A.; Chhetri, B.B.; Chapagain, K. Status, Challenges and Future Directions of Blockchain Technology in Power System: A State of Art Review. *Energies* **2022**, *15*, 8571. [CrossRef]
2. Ogundokun, R.O.; Misra, S.; Maskeliunas, R.; Damasevicius, R. A review on federated learning and machine learning approaches: Categorization, application areas, and blockchain technology. *Information* **2022**, *13*, 263. [CrossRef]
3. Ogundokun, R.O.; Arowolo, M.O.; Misra, S.; Damasevicius, R. An efficient blockchain-based IoT system using improved KNN machine learning classifier. In *Blockchain Based Internet of Things*; De, D., Bhattacharyya, S., Rodrigues, J.J.P.C., Eds.; Lecture Notes on Data Engineering and Communications Technologies; Springer: Singapore, 2022; Volume 112. [CrossRef]
4. Aslan, B.; Ataşen, K. COVID-19 information sharing with blockchain. *Inf. Technol. Control.* **2021**, *50*, 674–685. [CrossRef]
5. Omohundro, S. Cryptocurrencies, smart contracts, and artificial intelligence. *AI Matters* **2014**, *1*, 19–21. [CrossRef]
6. Li, Y. Emerging blockchain-based applications and techniques. *Serv. Oriented Comput. Appl.* **2019**, *13*, 279–285. [CrossRef]
7. Sun, J.; Yan, J. Blockchain-Based sharing services: What blockchain technology can contribute to smart cities. *Financ. Innov.* **2016**, *2*, 26. [CrossRef]
8. Dorri, A.; Kanhere, S.S.; Jurdak, R. Towards an Optimized BlockChain for IoT. In Proceedings of the Second International Conference on Internet-of-Things Design and Implementation (IoTDI'17), Pittsburgh, PA, USA, 18–21 April 2017; ACM: New York, NY, USA, 2017; pp. 173–178. [CrossRef]
9. Dorri, A.; Kanhere, S.S.; Jurdak, R.; Gauravaram, P. Blockchain for IoT Security and Privacy: The Case Study of a Smart Home. In Proceedings of the IEEE Percom Workshop on Security Privacy and Trust in the Internet of Things, Kona, HI, USA, 13–17 March 2017.
10. Wenting, L.; Alessandro Sforzin, A.; Fedorov, S.; Karame, G.O. Towards Scalable and Private Industrial Blockchains. In Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies, and Contracts (BCC'17), Abu Dhabi, UAE, 2 April 2017; ACM: New York, NY, USA, 2017; pp. 9–14. [CrossRef]
11. Yuan, Y.; Wang, F.Y. Towards blockchain-based intelligent transportation systems. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 2663–2668. [CrossRef]
12. Sharma, P.K.; Moon, S.Y.; Park, J.H. Block-VN: A distributed blockchain-based vehicular network architecture in smart City. *J. Inf. Process. Syst.* **2017**, *13*. [CrossRef]
13. Svetinovic, D. Blockchain Engineering for the Internet of Things: Systems Security Perspective. In Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security (IoTPTS'17), Abu Dhabi, UAE, 2 April 2017; ACM: New York, NY, USA, 2017; p. 1. [CrossRef]

14. Sharma, P.K.; Singh, S.; Jeong, Y.S.; Park, J.H. Distblocknet: A distributed blockchain-based secure sdn architecture for IoT networks. *IEEE Commun. Mag.* **2017**, *55*, 78–85. [CrossRef]
15. Luu, L.; Chu, D.H.; Olickel, H.; Saxena, P.; Hobor, A. Making Smart Contracts Smarter. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS'16), Vienna, Austria, 24–26 October 2016; ACM: New York, NY, USA, 2016; pp. 254–269. [CrossRef]
16. Wangsaputra, N.; Catur Candra, M.Z. Cadfort: A Decentralized Internet of Things Platform Based on Kademlia. In Proceedings of the 2018 5th International Conference on Data and Software Engineering (ICoDSE), Mataram, Indonesia, 7–8 November 2018. [CrossRef]
17. Nguyen, Q.K. Blockchain—A Financial Technology for Future Sustainable Development. In Proceedings of the 2016 3rd International Conference on Green Technology and Sustainable Development (GTSD), Kaohsiung, Taiwan, 24–25 November 2016; pp. 51–54. [CrossRef]
18. Asharaf, S.; Adarsh, S. *Decentralized Computing Using Blockchain Technologies and Smart Contracts: Emerging Research and Opportunities*; IGI Global: Hershey, PA, USA, 2017.
19. Weking, J.; Mandalenakis, M.; Hein, A.; Hermes, S.; Böhm, M.; Krcmar, H. The impact of blockchain technology on business models–a taxonomy and archetypal patterns. *Electron. Mark.* **2019**, *30*, 285–305. [CrossRef]
20. Treiblmaier, H.; Rejeb, A.; Strebinger, A. Blockchain as a Driver for Smart City Development: Application Fields and a Comprehensive Research Agenda. *Smart Cities* **2020**, *3*, 853–872. [CrossRef]
21. Xu, L.; Shah, N.; Chen, L.; Diallo, N.; Gao, Z.; Lu, Y.; Shi, W. Enabling the Sharing Economy: Privacy Respecting Contract based on Public Blockchain. In Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts (BCC'17), Abu Dhabi, UAE, 2 April 2017; ACM: New York, NY, USA, 2017; pp. 15–21. [CrossRef]
22. Gori, P.; Parcu, P.L.; Stasi, M.L. *Smart Cities and Sharing Economy, Vol 96, Robert Schuman Centre for Advanced Studies Research Paper No*; RSCAS. European University Institute: Fiesole, Italy, 2015.
23. Zutshi, A.; Grilo, A.; Nodehi, T. The value proposition of blockchain technologies and its impact on Digital Platforms. *Comput. Ind. Eng.* **2021**, *155*, 107187. [CrossRef]
24. Karunakaran, A.; Divakaran, P. Decentralized blockchain data storage using artificial intelligence. Our Heritage. In Proceedings of the GRCF Dubai International Conference on Sustainability and Innovation in Higher Education, Engineering Technology, Science, Management and Humanities, Dubai, UAE, 23–24 November; 2019; Volume 67, pp. 8–13.
25. Lavanga, M.; Drosner, M. Towards a New Paradigm of the Creative City or the Same Devil in Disguise? Culture-led Urban (Re)development and Sustainability. In *Cultural Industries and the Environmental Crisis*; Oakley, K., Banks, M., Eds.; Springer: Cham, Switzerland, 2020. [CrossRef]
26. Rubino, S.C.; Hazenberg, W.; Huisman, M. *Meta Products: A Meaningful Design for Our Connected World*; BIS Publishers: Amsterdam, The Netherlands, 2011.
27. Putra, G.D.; Dedeoglu, V.; Kanhere, S.S.; Jurdak, R. Trust management in decentralized IoT access control system. In Proceedings of the 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Toronto, ON, Canada, 3–6 May 2020; pp. 1–9.
28. Malik, S.; Dedeoglu, V.; Kanhere, S.S.; Jurdak, R. Trustchain: Trust management in blockchain and IoT supported supply chains. In Proceedings of the 2019 IEEE International Conference on Blockchain, Atlanta, GA, USA, 14–17 July 2019; pp. 184–193.
29. Zhang, R.; Xue, R.; Liu, L. Security, and Privacy on Blockchain. *ACM Comput. Surv.* **2019**, *52*, 1–34. [CrossRef]
30. Crosby, M.; Pattanayak, P.; Verma, S.; Kalyanaraman, V. Blockchain technology: Beyond bitcoin. *Appl. Innov.* **2016**, *2*, 6–10.
31. Shrestha, A.K.; Vassileva, J.; Deters, R. A Blockchain Platform for User Data Sharing Ensuring User Control and Incentives. *Front. Blockchain* **2020**, *3*, 497985. [CrossRef]
32. An, Y.; Liu, Y.; Zeng, J.; Du, H.; Zhang, J.; Zhao, J. Trusted collection, management, and sharing of data based on blockchain and IoT devices. In Proceedings of the 2019 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), Zhengzhou, China, 6–8 November 2019; pp. 27–32.
33. Ramachandran, A.; Kantarcioglu, M. Smartprovenance: A distributed, blockchain-based data provenance system. In Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy, Tempe, AZ, USA, 19–21 March 2018; pp. 35–42.
34. Wang, S.; Zhang, Y.; Zhang, Y. A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. *IEEE Access* **2018**, *6*, 38437–38450. [CrossRef]
35. Saad, M.; Spaulding, J.; Njilla, L.; Kamhoua, C.A.; Nyang, D.; Mohaisen, A. Overview of Attack Surfaces in Blockchain. In *Blockchain for Distributed Systems Security*; Wiley: Hoboken, NJ, USA, 2019; pp. 51–66. [CrossRef]
36. Saad, M.; Spaulding, J.; Njilla, L.; Kamhoua, C.; Shetty, S.; Nyang, D.; Mohaisen, D. Exploring the Attack Surface of Blockchain: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1977–2008. [CrossRef]
37. Glaser, F. Pervasive Decentralisation of Digital Infrastructures: A Framework for Blockchain-enabled System and Use Case Analysis. In Proceedings of the 50th Hawaii International Conference on System Sciences, HICSS 2017, Hilton Waikoloa Village, HI, USA, 4–7 January 2017.
38. Voskobojnikov, A.; Skwarek, V.; Mashatan, A.; Matsuo, S.; Rowell, C.; Weingärtner, T. Balancing Security: A Moving Target. In *Building Decentralized Trust*; Lemieux, V.L., Feng, C., Eds.; Springer: Cham, Switzerland, 2021. [CrossRef]
39. Dorri, A.; Roulin, C.; Jurdak, R.; Kanhere, S.S. On the activity privacy of blockchain for IoT. In Proceedings of the 2019 IEEE 44th Conference on Local Computer Networks (LCN), Osnabrueck, Germany, 14–17 October 2019; pp. 258–261.

40. Zhao, Y.; Zhao, J.; Jiang, L.; Tan, R.; Niyato, D.; Li, Z.; Liu, Y. Privacy-preserving blockchain-based federated learning for IoT devices. *IEEE Internet Things J.* **2020**, *8*, 1817–1829. [CrossRef]
41. Chiew, K.L.; Yong, K.S.C.; Tan, C.L. A survey of phishing attacks: Their types, vectors and technical approaches. *Expert Syst. Appl.* **2018**, *106*, 1–20. [CrossRef]
42. Basit, A.; Zafar, M.; Liu, X.; Javed, A.R.; Jalil, Z.; Kifayat, K. A comprehensive survey of AI-enabled phishing attacks detection techniques. *Telecommun. Syst.* **2021**, *76*, 139–154. [CrossRef]
43. Jain, A.K.; Gupta, B.B. A survey of phishing attack techniques, defense mechanisms, and open research challenges. *Enterp. Inf. Syst.* **2021**, *16*, 527–565. [CrossRef]
44. Khonji, M.; Iraqi, Y.; Jones, A. Phishing detection: A literature survey. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 2091–2121. [CrossRef]
45. Azeez, N.A.; Salaudeen, B.B.; Misra, S.; Damaševičius, R.; Maskeliūnas, R. Identifying phishing attacks in communication networks using URL consistency features. *Int. J. Electron. Secur. Digit. Forensics* **2020**, *12*, 200–213. [CrossRef]
46. Andryukhin, A.A.; Phishing, A. Preventions in Blockchain-Based Projects. In Proceedings of the International Conference on Engineering Technologies and Computer Science (EnT), Moscow, Russia, 26–27 March 2019. [CrossRef]
47. Alharbi, A.; Alosaimi, W.; Alyami, H.; Rauf, H.T.; Damaševičius, R. Botnet attack detection using local-global best bat algorithm for the industrial internet of things. *Electronics* **2021**, *10*, 1341. [CrossRef]
48. Toldinas, J.; Venčkauskas, A.; Damaševičius, R.; Grigaliūnas, Š.; Morkevičius, N.; Baranauskas, E. A novel approach for network intrusion detection using multistage deep learning image recognition. *Electronics* **2021**, *10*, 1854. [CrossRef]
49. Nisa, M.; Shah, J.H.; Kanwal, S.; Raza, M.; Khan, M.A.; Damaševičius, R.; Blažauskas, T. Hybrid malware classification method using segmentation-based fractal texture analysis and deep convolution neural network features. *Appl. Sci.* **2020**, *10*, 4966. [CrossRef]
50. Hemalatha, J.; Roseline, S.A.; Geetha, S.; Kadry, S.; Damaševičius, R. An efficient densenet-based deep learning model for malware detection. *Entropy* **2021**, *23*, 344. [CrossRef]
51. Awan, M.J.; Masood, O.A.; Mohammed, M.A.; Yasin, A.; Zain, A.M.; Damaševičius, R.; Abdulkareem, K.H. Image-based malware classification using vgg19 network and spatial convolutional attention. *Electronics* **2021**, *10*, 2444. [CrossRef]
52. Yong, B.; Wei, W.; Li, K.; Shen, J.; Zhou, Q.; Wozniak, M.; Damaševičius, R. Ensemble machine learning approaches for web shell detection in the internet of things environments. *Trans. Emerg. Telecommun. Technol.* **2020**, *33*, e4085. [CrossRef]
53. Damaševičius, R.; Venčkauskas, A.; Toldinas, J.; Grigaliūnas, Š. Ensemble-based classification using neural networks and machine learning models for windows pe malware detection. *Electronics* **2021**, *10*, 485. [CrossRef]
54. Azeez, N.A.; Odufuwa, O.E.; Misra, S.; Oluranti, J.; Damaševičius, R. Windows PE malware detection using ensemble learning. *Informatics* **2021**, *8*, 10. [CrossRef]
55. Awotunde, J.B.; Ogundokun, R.O.; Misra, S.; Adeniyi, E.A.; Sharma, M.M. Blockchain-Based Framework for Secure Transaction in Mobile Banking Platform. *Adv. Intell. Syst. Comput.* **2021**, *1375*, 525–534.
56. Buterin, V. A next-generation smart contract, and decentralized application platform. *White Pap.* **2014**, *3*, 37.
57. Eskandari, S.; Clark, J.; Barrera, D.; Stobert, E. A first look at the usability of bitcoin key management. *arXiv* **2018**, arXiv:1802.04351.
58. Sheng, S.; Broderick, L.; Koranda, C.A.; Hyland, J.J. Why Johnny Still Can't Encrypt: Evaluating the Usability of Email Encryption Software. Available online: https://cups.cs.cmu.edu/soups/2006/posters/shengposter_abstract.pdf (accessed on 25 December 2019).
59. Gaw, S.; Felten, E.W.; Fernandez-Kelly, P. Secrecy, flagging, and paranoia: Adoption criteria in an encrypted email. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Montréal, QC, Canada, 22–27 April 2006; pp. 591–600.
60. Schultz, E.E.; Proctor, R.W.; Lien, M.C.; Salvendy, G. Usability and security an appraisal of usability issues in information security methods. *Comput. Secure.* **2001**, *20*, 620–634. [CrossRef]
61. Garfinkel, S.L.; Margrave, D.; Schiller, J.I.; Nordlander, E.; Miller, R.C. How to make secure email easier to use. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Portland, OR, USA, 2–7 April 2005; pp. 701–710.
62. Ruoti, S.; Seamons, K. Johnny's Journey Toward Usable Secure Email. *IEEE Secure. Priv.* **2019**, *17*, 72–76. [CrossRef]
63. Pham, T.; Lee, S. Anomaly detection in bitcoin network using unsupervised learning methods. *arXiv* **2016**, arXiv:1611.03941.
64. Do, H.G.; Ng, W.K. Blockchain-based system for secure data storage with private keyword search. In Proceedings of the 2017 IEEE World Congress on Services (SERVICES), Honolulu, HI, USA, 25–30 June 2017; pp. 90–93.
65. Devi, K.; Paulraj, D.; Muthusenthil, B. Deep Learning Based Security Model for Cloud based Task Scheduling. *KSII Trans. Internet Inf. Syst.* **2020**, *14*, 3663–3679. [CrossRef]
66. Sermakani, A.M.; Paulraj, D. Effective Data Storage and Dynamic Data Auditing Scheme for Providing Distributed Services in Federated Cloud. *J. Circuits Syst. Comput.* **2020**, *29*, 2050259. [CrossRef]
67. Hariharan, B.; Paul Raj, D. WBAT Job Scheduler: A Multi-Objective Approach for Job Scheduling Problem on Cloud Computing. *J. Circuits Syst. Comput.* **2019**, *29*, 2050089. [CrossRef]
68. Perard, D.; Gicquel, L.; Lacan, J. BlockHouse: Blockchain-based Distributed Storehouse System. In Proceedings of the 2019 9th Latin-American Symposium on Dependable Computing (LADC), Natal, Brazil, 19–21 November 2019; pp. 1–4.
69. Michelin, R.A.; Dorri, A.; Steger, M.; Lunardi, R.C.; Kanhere, S.S.; Jurdak, R.; Zorzo, A.F. SpeedyChain: A framework for decoupling data from the blockchain for smart cities. In Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services, New York, NY, USA, 5–7 November 2018; pp. 145–154.

70. Hossain, E.; Khan, I.; Un-Noor, F.; Sikander, S.S.; Sunny, M.S.H. Application of big data and machine learning in smart grid, and associated security concerns: A review. *IEEE Access* **2019**, *7*, 13960–13988. [CrossRef]
71. Anthi, E.; Williams, L.; Słowińska, M.; Theodorakopoulos, G.; Burnap, P. A supervised intrusion detection system for smart home IoT devices. *IEEE Internet Things J.* **2019**, *6*, 9042–9053. [CrossRef]
72. Wei, L.; Luo, W.; Weng, J.; Zhong, Y.; Zhang, X.; Yan, Z. Machine learning-based malicious application detection of android. *IEEE Access* **2017**, *5*, 25591–25601. [CrossRef]
73. Restuccia, F.; D'Oro, S.; Melodia, T. Securing the internet of things in the age of machine learning and software-defined networking. *IEEE Internet Things J.* **2018**, *5*, 4829–4842. [CrossRef]
74. Mahdavinejad, M.S.; Rezvan, M.; Barekatain, M.; Adibi, P.; Barnaghi, P.; Sheth, A.P. Machine learning for Internet of Things data analysis: A survey. *Digit. Commun. Netw.* **2018**, *4*, 161–175. [CrossRef]
75. Kumar, N.; Singh, A.; Handa, A.; Shukla, S.K. Detecting Malicious Accounts on the Ethereum Blockchain with Supervised Learning. In Proceedings of the International Symposium on Cyber Security Cryptography and Machine Learning, Be'er Sheva, Israel, 8–9 July 2020; pp. 94–109.
76. Dalal, H.; Abulaish, M. A multilayer perceptron architecture for detecting deceptive cryptocurrencies in coin market capitalization data. In Proceedings of the 2019 IEEE/WIC/ACM International Conference on Web Intelligence, Thessaloniki, Greece, 14–17 October 2019; pp. 438–442.
77. Yuan, Z.; Yuan, Q.; Wu, J. Phishing Detection on Ethereum via Learning Representation of Transaction Subgraphs. In Proceedings of the International Conference on Blockchain and Trustworthy Systems, Dali, China, 6–7 August 2020; pp. 178–191.
78. Chen, L.; Peng, J.; Liu, Y.; Li, J.; Xie, F.; Zheng, Z. Phishing Scams Detection in Ethereum Transaction Network. *ACM Trans. Internet Technol.* **2021**, *21*, 1–16. [CrossRef]
79. Chen, W.; Zheng, Z.; Ngai, E.C.H.; Zheng, P.; Zhou, Y. Exploiting Blockchain Data to Detect Smart Ponzi Schemes on Ethereum. *IEEE Access* **2019**, *7*, 37575–37586. [CrossRef]
80. Chen, W.; Zheng, Z.; Cui, J.; Ngai, E.; Zheng, P.; Zhou, Y. Detecting Ponzi schemes on ethereum: Towards healthier blockchain technology. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018.
81. GitHub—MyEtherWallet/Ethereum-Lists: A Repository for Maintaining Lists of Things Like Malicious URLs, Fake Token Addresses, and so Forth. Available online: https://github.com/MyEtherWallet/ethereum-lists (accessed on 1 February 2023).