applied sciences

MDPI

Review

Synergy of Blockchain Technology and Data Mining Techniques for Anomaly Detection

Aida Kamišalić [1,*], Renata Kramberger [2] and Iztok Fister, Jr. [1]

[1] Faculty of Electrical Engineering and Computer Science, University of Maribor, Koroška Cesta 46, 2000 Maribor, Slovenia; iztok.fister1@um.si
[2] Department of Information Technology and Computing, Zagreb University of Applied Sciences, Vrbik 8, 10000 Zagreb, Croatia; renata.kramberger@tvz.hr
* Correspondence: aida.kamisalic@um.si

Abstract: Blockchain and Data Mining are not simply buzzwords, but rather concepts that are playing an important role in the modern Information Technology (IT) revolution. Blockchain has recently been popularized by the rise of cryptocurrencies, while data mining has already been present in IT for many decades. Data stored in a blockchain can also be considered to be big data, whereas data mining methods can be applied to extract knowledge hidden in the blockchain. In a nutshell, this paper presents the interplay of these two research areas. In this paper, we surveyed approaches for the data mining of blockchain data, yet show several real-world applications. Special attention was paid to anomaly detection and fraud detection, which were identified as the most prolific applications of applying data mining methods on blockchain data. The paper concludes with challenges for future investigations of this research area.

Keywords: anomaly detection; blockchain; distributed ledger; data mining; machine learning

Citation: Kamišalić, A.; Kramberger, R.; Fister, I., Jr. Synergy of Blockchain Technology and Data Mining Techniques for Anomaly Detection. *Appl. Sci.* **2021**, *11*, 7987. https://doi.org/10.3390/app11177987

Academic Editor: Gianluca Lax

Received: 27 July 2021
Accepted: 25 August 2021
Published: 29 August 2021

1. Introduction

Blockchain technology [1] might be considered one of the most disruptive technologies of the last decade, which can revolutionise business processes within the private and public sectors. It offers the means to secure processed transactions in distributed and decentralised environments, providing transparency and immutability [2,3]. Nevertheless, there are still several challenges associated with technology application related to security, scalability, interoperability, regulation. On the other hand, machine learning (ML) applications have emerged in recent years, due to the availability of vast amounts of data and the capacity of ML algorithms to provide systems with the ability to learn and improve automatically using past data [4,5]. Blockchain technology can benefit from the use of ML algorithms while taking advantage of their ability to provide an analysis for an enormous amount of data. It can enhance the security of such systems significantly [6–9].

The applications that benefit from blockchain technology and machine learning algorithms rise promptly within different domains, such as healthcare, fintech, and the energy sectors and for different purposes, such as anomaly, fraud and malicious activity detection, biometrics' monitoring and disease detection, etc. Several reviews have dealt with these topics in recent years. Some of them address the integration of blockchain technology and artificial intelligence among other technologies to achieve decentralised authentication [6], to enable different features within the 5G networks [8], or to achieve the de-anonymisation of bitcoin addresses or entity recognition within cryptocurrency transaction networks [10]. Other works provided have surveyed the use of one or both technologies separately within a specific domain, such as Healthcare [3,11–14], Agriculture [15], Construction Engineering and the built environment [16,17], or across several domains such as Data Management and IoT [3,18], Blockchain industrial applications [2,19], or addressing security and privacy issues [7,9,20]. A detailed insight into these reviews (presented in Section 3) reveals that

none of these provide a comprehensive review of all applications regardless of the domain where the blockchain technology and machine learning techniques are used to complement each other, offering complete solutions for anomaly and fraud detection. Therefore, we contribute to the body of knowledge by (1) Providing a comprehensive review of applications where the synergy of blockchain technology and machine learning algorithms is used to detect anomalies, (2) Discovering all main machine learning methods used and the types of data they exploit, (3) Offering a taxonomy of machine learning methods used to enhance blockchain technology, serving a specific purpose.

The structure of the paper is as follows. Section 2 provides fundamental information on blockchain technology. The research methodology used in the review, research questions and taxonomy, as well as an overview of existing reviews covering those topics are presented in Section 3. Sections 4 and 5 provide a detailed analysis of the machine learning methods used for intelligent data analysis and review of applications, respectively. A discussion can be found in Section 6, while Section 7 summarize the directions of the further development of the research field and issues to be addressed. Finally, conclusions and future trends are presented in Section 8.

2. Fundamentals of Blockchain Technology

In recent years, blockchain technology has been attracting much attention due to its features that complement storage technologies. With the expansion of cryptocurrencies it has become the most known representative of distributed ledger technologies. Blockchain technology enables the creation of a decentralised and distributed environment. It serves as a secure and immutable ledger, allowing transactions' to be processed without being controlled by a central authority. A ledger consists of a chain of blocks that store data chronologically. The chain is growing as new blocks are being appended to the end of the ledger (see Figure 1). Each new block holds a hash value reference to the content of the previous block which assures the immutability of saved data. Data are structured into transactions and sealed using cryptography, where a public key encryption mechanism is used to secure the content. The content is replicated, distributed and synchronised among nodes in a P2P network. Therefore, the consistency, immutability and transparency of the content is ensured using all the mentioned principles [21]. Since it is based on decentralisation principles, the consensus mechanism ensures a fault-tolerant system where nodes reach agreement on a single source of truth—the state of the ledger content. Its protocols ensure that nodes are synchronised and agree on transactions added to the ledger. The chosen mechanism influences the features of the blockchain directly, and its sustainable applicability in different domains. Some of the most known consensus mechanisms are: Proof-of-Work (PoW), Proof-of-Stake (PoS), Delegated Proof-of-Stake (DPoS), Proof-of-Authority (PoA), Proof-of-Activity, Proof-of-Identity, etc. [22–24].

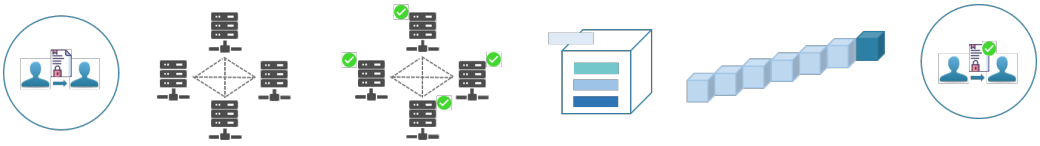

Figure 1. Blockchain transaction processing: (1) Transaction request, (2) Transaction broadcast to P2P network nodes, (3) Transaction verification using consensus algorithms, (4) Transaction packed into block with other transactions, (5) The block added to the blockchain, (6) Transaction completed

Essentially, there are four types of blockchain: public, private, consortium and hybrid. A public blockchain is permissionless, and it emphasises the public part-all the data are accessible and visible to the public. Typical representatives of this type of blockchains are cryptocurrencies [25]. On the other hand, a private blockchain is based on permissioned principles, therefore, only chosen nodes can join the closed network. Here, the network is still distributed but centralised, whereby the security, authorisations, permissions and accessibility are in the domain of a central authority—a single organisation. They are

usually used within one organisation or enterprise, typically for voting processes, digital identity, asset ownership, etc [18,21]. Consortium blockchains are partially decentralised, since more than one organisation manages the network. Only selected participants get full access. The typical use of this type of blockchain can be found in banks, government organisations, etc [18,21]. Finally, hybrid blockchains are a combination of private and public types of blockchains, where some instances are public and some restricted. Those networks are not open to everyone, but still offerthe main features of blockchain technology—integrity, transparency and security. They provide flexible control over the blockchain and are suitable for domains such as finance, supply chains, medical records, etc. [18,26,27].

The first application of blockchain technology emerged in 2009, with the cryptocurrency Bitcoin [1]. Since then, we have witnessed the evolution of blockchain technology and its expansion in a number of applications. However, the most known applications of blockchain technology are cryptocurrencies. Nowadays, there are over 10,000 different cryptocurrencies. Nevertheless, the huge potential of this technology can be seen through its usage within different domains, where new business models and efficiency efforts might be leveraged from the technology features. Several use cases that benefit from this technology are found in Healthcare, IoT, Supply chain management, Education, Electronic voting, Resource management, Transportation, Insurance, Energy, and Rights management [28,29].

3. Research Methodology

As stated in the Introduction, the main objective of this paper was to review the interplay of machine learning methods and blockchain data for anomalies and fraud detection. Therefore, this research was conducted by reviewing the present state of knowledge of applications using blockchain technology and machine learning methods.

While designing this research paper, we formed three research questions, which are addressed and answered in the paper. The research questions are as follows:

Research Question 1 (RQ1). *What are the core elements for defining the taxonomy of machine learning methods used for anomaly and fraud detection in blockchain systems?*

Research Question 2 (RQ2). *Which machine learning methods are used for the intelligent data analysis of anomalies using data stored in a blockchain?*

Research Question 3 (RQ3). *What type of applications emerge from the use of machine learning methods within a blockchain environment?*

Based on the research questions, the review was undertaken to address the following specific goals:

- To review machine learning methods used for the intelligent data analysis of anomalies using data saved in a blockchain environment.
- To synthesise a taxonomy of ML methods used for specific purposes.
- To review applications that benefit from blockchain technology and machine learning algorithms.

We started our study with an extensive literature search in several scientific abstract databases. In order to collect the required articles, the following search string was used:

```
("blockchain'' OR ''block chain'' OR ''distributed ledger'' OR
''smart contract'' OR ''cryptocurrency'') AND
(''data mining'' OR ''classification'' OR ''machine learning'' OR
''AI'' OR ''preprocessing'' OR ''deep learning'' OR
''neural network'' OR ''artificial intelligence'' OR
''anomaly detection'')
```

The search string was modified to meet the requirements and limitations of each selected search engine. Initially, the search was conducted so that the engines took the entire text of the articles into account. This led to a large number of results, and a large number of retrieved articles that were not relevant to our study. In order to fix this, the search was limited to include only abstracts and keywords. Additionally, the search was only limited to include results published within the last five years (2017 to 2021).

The search was conducted between the 11th and 14th of June, 2021. The following search engines and scientific databases were used: ACM Digital Library, IEEE Xplore, Science Direct, Springer Link, and Web of Science. Table 1 shows the number of results obtained from each database.

Table 1. Databases with the total number of search results.

Database Name	URL	No. Total Results
ACM Digital Library	dl.acm.org	119
IEEE Xplore	ieeexplore.ieee.org	982
Science Direct	sciencedirect.com	103
Springer Link	link.springer.com	2398
Web of Science	webofknowledge.com	1004
Total		4606

After the results had been collected, they were also checked to exclude duplicates, both within the databases and given by different databases. Table 2 shows the number of duplicates found among the databases. Given that Web of Science indexes articles that are hosted by other databases, it can be seen that most of the duplicates were found when comparing a specific database to Web of Science. Altogether, 18 duplicates were found within the databases, and 522 across each database pair. A total of 540 duplicates were excluded.

Table 2. Number of duplicates excluded, shown by each database pair.

	ACM	IEEE	Science Direct	Springer Link	WoS
ACM Digital Library	15	4	0	0	6
IEEE Xplore	4	1	0	0	425
Science Direct	0	0	0	0	45
Springer Link	0	0	0	0	46
Web of Science	6	425	45	46	2

We also defined several additional exclusion criteria that resulted in the removal of papers from our selection pool:

- Research not written in the English language,
- The full text of the article was not available, and
- The method, evaluation process, and results were not described.

After inspecting the title, keywords and the abstracts, 206 papers were initially selected for our research. However, one paper was not written in English, and we could not obtain the full text for nine papers. Therefore, these 10 papers were also excluded. A full text inspection was conducted on a total of 196 studies, and 130 were selected as relevant for our research.

As we can observe in Figure 2, there has been an increasing trend in using machine learning techniques in blockchain anomaly detection. Although it seems as though there is a decrease in the number of publications in the year 2021, note that we conducted the search in June 2021 and the 36 publications were published in the first six months. We can expect that more research within this area will be published until the end of the year, and will exceed the number of publications from the year 2020.

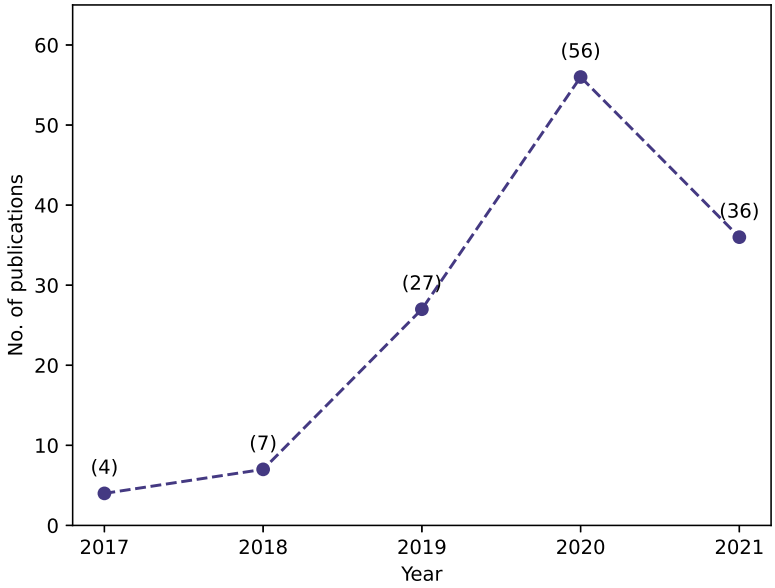

Figure 2. Number of publications per year included in the research.

Similar Review Papers

During the search process we were able to detect several review papers addressing topics on blockchain technology and artificial intelligence. Below we present those reviews and their contributions.

Mohsin et al. [6] published a review on integrating a blockchain technology with IoT, Telemedicine, Cloud computing and Artificial Intelligence among others, in order to achieve decentralised authentication. A state-of-the-art survey on the integration of blockchain with 5G networks was published by Nguyen et al. [8]. They detected several works combining machine learning with blockchain in 5G networks, to enable secure and intelligent resource management and orchestration, optimisation, secure computation offloading, and reliable network channel selection, etc. Wu et al. [10] presented a comprehensive review of the state-of-the-art literature on cryptocurrency transaction networks. They detected works using machine learning or deep learning for the de-anonymisation of bitcoin addresses or entity recognition. In 2020 Lezoche et al. [15] published a survey on new technologies (focusing on Big Data, AI, IoT, and Blockchain), and new supply chain methods that were analysed within the Agriculture domain. Azbeg et al. [11] offered a review of healthcare applications where the IoT and blockchain were integrated, Kouicem et al. [12] published a survey of security and privacy solutions in IoT, and the benefits that blockchain technology, among other things, might bring to security and privacy in terms of flexibility and scalability, while Mohd Aman et al. [13] reviewed architecture, applications, technologies and security developments made within the Internet of Medical Things (IoMT) in the COVID-19 period, providing an insight into the used technologies (i.e., blockchain, machine learning, big data) within the medical environment. Negro-Calduch et al. [14] performed a systematic review of systematic reviews to assess technological progress in the Electronic Health Record (EHR) and Personal Health Record (PHR) systems, whereby EHRs and PHRs are considered to be the primary beneficiaries of the implementation of blockchain technology and natural language processing techniques, (i.e., rule-based, machine learning, or deep learning-based) are considered useful for the extraction of information from clinical narratives and other unstructured data within EHRs and PHRs. A systematic literature review of blockchain-based applications across several domains, such as from supply chains, business, healthcare, IoT, privacy, and data management was published by Casino et al. [3]. Lu [18] published a review of the main applications based on the blockchain technology and studies of the blockchain and main components, blockchain-based IoT, blockchain-based security and blockchain-based data management. A survey of industrial blockchain, identifying

challenges and opportunities, and summarising the main obstacles of industrial blockchain, was conducted by Li et al. [2]. Hoffmann Souza et al. [19] published a survey on decision-making based on system reliability in the context of Industry 4.0. A systematic review presenting the current state of AI adoption in the context of Construction Engineering and Management (CEM), where Blockchain technology was detected as one of the key future research directions that would enable narrowing the gap between AI and CEM, was published by Pan and Zhang [16]. Nawari and Ravidran [17], on the other hand, presented an evaluation survey of Blockchain technology and its applications in the built environment. Peng et al. [20] analysed the characteristics of permissionless blockchain and summarised potential privacy threats. Valdovinos et al. [9] provided a systematic survey of the existing Distributed Denial of Service (DDoS) attacks detection and mitigation strategies in Software-Defined Networking (SDN). They provided a taxonomy of DDoS detection strategies (e.g., statistical, Machine Learning) and emerging approaches (e.g., network function virtualisation, blockchain, honeynet, network slicing, and moving target defence). Cryptographic techniques proposed to achieve authentication, privacy and other security features within Vehicular ad hoc networks (VANETs) were in the focus of a study published by Mudhe et al. [7].

4. Identified Data Mining Methods

Data stored in blockchain can definitely be considered to be Big Data [30,31]. For example, the full blockchain size of the most popular cryptocurrency Bitcoin is more than 300 gygabytes at the time of writing this paper (https://www.statista.com/statistics/647523/worldwide-bitcoin-blockchain-size/ (accessed on 15 August 2021)). Very sophisticated Data Mining tools and methods are usually used or developed for deep analysis of these data. There exists a wide pool of Data Mining methods nowadays, such as, for example, from Regression Analysis to more complex ML methods, Random Forest or Support Vector Machines [32]. Typically, these methods are used to cope with Association Rule Mining, Numerical Association Rule Mining, clustering, classification and others. Data Mining can also be viewed as a part of the KDD (Knowledge Discovery in Databases) [33] process, where the term KDD refers to the overall process of discovering useful knowledge from various data [33]. In this process, DM is actually an application of algorithms aimed to extract patterns from data. In line with this, KDD is a very complex process which consists of six phases [34,35]:

- Problem/application specification (a problem is introduced which plays the main role)
- Problem/application understanding (problem tries to become explainable),
- Data preprocessing, (data cleaning, data transformation, feature selection procedures take place)
- Data mining, (an actual model is built on preprocessed data and knowledge is extracted)
- Evaluation, (interpretation of results takes place)
- Result exploitation (visualisation of discovered knowledge, generation of reports, narrating stories).

Many research papers and practice have revealed that data preprocessing is probably the hardest step in the overall process of the KDD. If data are well prepared, cleaned and transformed, then the subsequent processes that follow occur more smoothly. Loosely speaking, the data preprocessing step, in most literature, is considered to be the most significant and most time consuming process in the whole pipeline [34]. Blockchain data are also very interesting from this perspective, because they usually involve a couple of additional steps that are not necessary for the data stored in spreadsheets, transaction databases and similar sequential-based formats. Blockchain data are, typically, stored in specially prescribed formats, which ensures data immutability. Data are structured into sets of valid transactions, which are packed into blocks. A block of transactions holds a reference in the form of a hash value to the content of the previous block. Each block is sealed cryptographically and appended to the end of the ledger. Therefore, retrieving data from a blockchain is not easy or effortless. It requires, firstly, to use specific parsers

for each blockchain, in order to extract raw data and make a systematic extraction and presentation of these data. Some examples of parsers can be found in the following links (https://github.com/alecalve/python-bitcoin-blockchain-parser (accessed on 15 August 2021)) (https://github.com/gcarq/rusty-blockparser (accessed on 15 August 2021)).

The following heatmap (Figure 3) presents all Data mining methods that were identified in our study. To improve the readability and clarity of the heatmap, we excluded all of the methods that appeared only once (AdaBoost, Adaptive Weighted Attribute Propagation, Bagging, Broad Learning System, Cascading Machine Learning, DBSCAN, Deep Hashing, Ensemble Learning, Generative Adversarial Network, k-Means Clustering, Link Mining, Linear Regression, and Logistic Regression), as well as some custom Supervised Learning methods. In our research we also encountered hybrid solutions that are based on statistical methods such as the Gaussian Graphical Model. As we can see from the heatmap, the authors utilised Computational Intelligence methods [36], conventional Machine Learning methods [5] as well as Deep Learning methods [37]. We can see that the most used methods for anomaly detection are Support Vector Machines, Artificial Neural Networks, and the Random Forest algorithm, while for detecting fraud, the best methods appeared to be the Random Forest algorithm and Gradient Boosting.

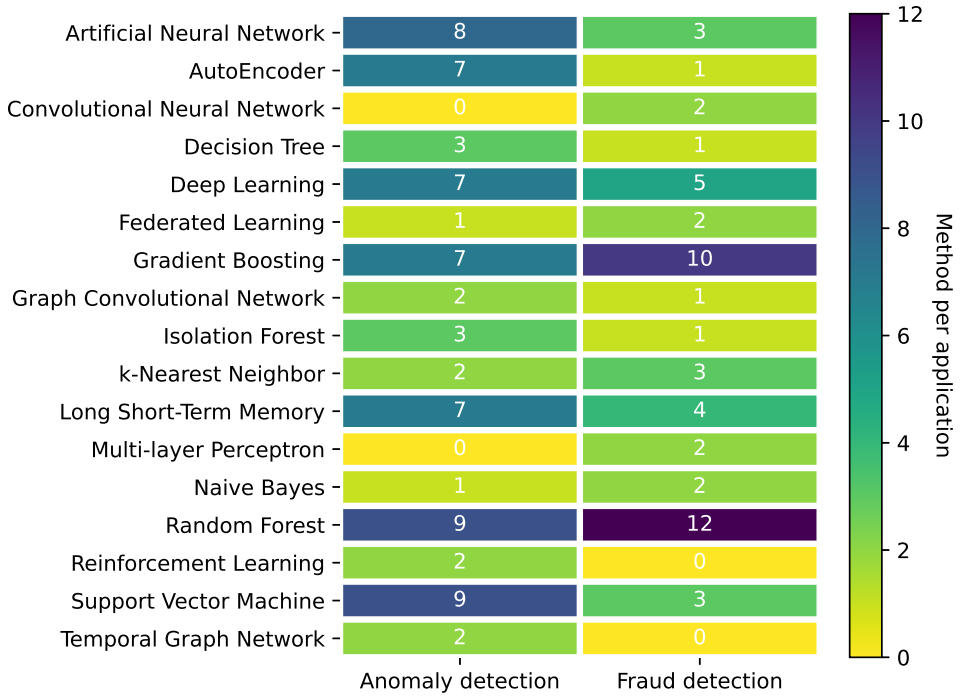

Figure 3. Heatmap of the most used methods per application.

5. Review of Applications

The performed literature review led to classification of publications regarding the addressed application. Publications were grouped into two categories of applications: Anomaly detection (see Tables 3–6) and Fraud detection (see Tables 7–10).

5.1. Anomaly Detection

Anomaly detection refers to the process of data processing and detection of behaviour patterns that may indicate a change in system operations [38]. The task of anomaly detection is searching for rare or suspicious events/items in data, which differ significantly from the whole dataset. In line with this, the process is also associated many times with the term outlier detection. Anomalies can be detected in practically most of the real-world datasets, but their practical use is arising in the detection of bank frauds, computer security, doping detection in sports, etc.

We divided the results further into four main groups associated with anomaly detection: Financial, Security, Data processing, and IoT.

5.1.1. Financial Anomaly Detection

One of the situations where detecting anomalies can be of use is money laundering. In their research, Alarab et al. [39] performed a comparative analysis of the use of multiple machine learning methods to detect money laundering in the Bitcoin blockchain. For the experiment, they used the elliptic data set (from which they excluded the time-step) and the aggregation features to enhance the performance. From the whole data set, unknown labels were excluded and only the licit and illicit remained. They used the Receiver Operating Characteristics (ROC) curve to present the visualisation of each of the tested techniques: Ensemble Learning, Random Forest, Extra Trees, Bagging, AdaBoost, Gradient Boosting, and k-Nearest Neighbours.

Graph Convolutional Networks assisted by linear layers were also used by Alarab et al. [40] to detect money laundering in the Bitcoin blockchain. The Elliptic data set was used for evaluation, and their method was then compared to the original GCN and Skip-GCN. The results that included precision, recall, F1 score, and accuracy, showed that their proposal reached the best score.

Table 3. Financial Anomaly Detection.

Method/Application	Money Laundering	Under-Priced DoS Attack	Credit Card Fraud	Pump and Dump	Anomalous Transactions	Transaction Signing	Honeypot Contracts	High Yield Investment Programme	Smart Contracts
Ensemble Learning	[39]								
GCN	[40]								
Decision Tree		[41]			[42]				
Random Forest		[41]	[43]	[44]				[45]	
XGBoost				[46]				[45]	
Isolation Forest						[47]			
LSTM					[48]				
LightGBM							[49]		
OCSVM					[50]				
Deep AutoEncoder									[51]

Eduardo et al. [41] tackled the problem of under-price DoS attacks in the Ethereum blockchain network. In under-price DoS attacks, malicious users perform denial of service attacks in order to exploit flaws in blockchain networks. One example of such exploits includes the Ethereum fee mechanism. In this scenario, the attackers pay a small fee for a large number of transactions. To test their network they created 2000 accounts and tested it with waves of normal transaction flows (2000 transactions), under-price DoS attack (10% of the transactions were malicious), and the Ethereum Boom (transactions from 20 December 2017). To test the flows they used real transactions from 5 May 2019. The results of the Machine Learning models show that the Decision Tree and Random Forest methods are most suitable for this type of task.

A system for credit card fraud prevention was presented by Balagolla et al. [43]. The system is trained to recognise anomalies within transactions stored on a blockchain. There were three data sets used to perform the tests: the credit card fraud detection data set (ULB), synthetic financials data set, and the German credit card fraud data set. Four Machine Learning algorithms were tested (Logistic Regression, SVM, XGBoost, and Random Forest), and the results showed that (based on the True Positive rate), the Random Forest algorithm had the best accuracy and Kappa value.

Cryptocurrency "pump and dump" schemes are a way of exploiting the blockchain market, where one first buys a cryptocurrency at a low price. Then, with the help of social media and similar platforms, they convince other investors to purchase the cryptocurrency, and, thus, increase its value. As others invest, the price of the cryptocurrency rises, and the organisers sell their shares at an (often) much higher price. Victor and Hagemann [46] presented a cryptocurrency pump and dump detection scheme. Quantification and detection is done based on data obtained from the Binance Exchange. Data were collected for the data set from 172 cryptocurrencies in one-second intervals, and filtered to contain only certain fields relevant for this analysis (timestamp, price of last trade, 24 h trading volume, and 24 h trade count). Also, 18 telegram channels were monitored to obtain timestamped messages associated with cryptocurrency pump and dump schemes. The data set was cleared of all small trade amounts, windows of 30 min were centered on ground-truth timestamps, and certain features (like entropy, stability, flat spots, etc.) were computed using the tsfeatures library. Finally XGBoost was used to detect pumps. Their timeline visualisation shows the comparison of pumps found by the model, and pumps from the ground truth.

Mirtaheri et al. [44] also created a system that detects pump and dump cryptocurrency manipulations by analysing social media and cryptocurrency market data. They collected the needed data from Telegram, Twitter, and CoinMarketCap.com. The data were then labelled as either pump/not-pump messages, and the Random Forest classifier was used to detect and predict pump and dump manipulations. Their approach was validated using an area under the receiver operating characteristic curve (ROC-AUC).

A system that detects suspicious activities in financial transactions and distributed ledgers was created by Camino et al. [42]. They conducted data preprocessing by first filtering out entries containing invalid values, then building a collection of vectors (grouped transactions by user) that will be analysed and filtered using the RFM (Recency, Frequency and Monetary) features. The Pearsons correlation coefficient was used to calculate the mutual influence of the features. Missing values from the data set were filled with a median value of that column, and the peak values were eliminated by subtracting the columns mean from it and then dividing it by its standard deviation. They used use cases to train decision trees, extract anomalies and detect anomalous accounts. They visualised their data using the t-SNE algorithm, as well as 2D and 3D scatter plots.

An LSTM network for anomaly detection and classification of Ethereum smart contracts was presented by Hu et al. [48]. They collected smart contracts from Ethereum, identified behaviour patterns manually, extracted features, and proposed a data slicing algorithm to slice the collected contracts. The proposed LSTM model was evaluated on the created data set, and the results were presented with the satisfactory precision, recall, and F-score.

An anomaly detection model for Bitcoin transactions was presented by Sayadi et al. [50]. The data used for the evaluation were obtained from the Bitcoin blockchain. They use One-Class SVM to detect outliers, and K-means clustering to gather similar attacks. The results of the evaluation were presented using confusion matrices, cluster frequencies, and detection results.

Podgorelec et al. [47] presented a machine learning based method for blockchain transaction signing and personalised anomaly detection. The data were collected from the Ethereum public main network. Isolation Forest was used to detect anomalous transactions, while Random Forest was used to determine the feature importance.

Honeypot contracts represent malicious contracts that are designed so that they have certain obvious flaws and attract other malicious users that wish to profit off of them. Of course, the flaws are carefully used to mask traps, and the only one that will ultimately profit is the creator of the honeypot contract [49]. Chen et al. [49] created a system that detects these kinds of contracts. To collect the data needed for the data set, they extracted honeypot contracts from the HONEYBADGER project and analysed each of the entries whether it was a honeypot or not. The results were then categorised by the used technology,

and the Ethereum ledger is downloaded into the data set. Feature extraction was conducted by converting the bytecode into opcodes, analysing the opcode frequency and using the bi-gram features to determine the opcode combination. Classification was done with the use of the LightGBM algorithm. The evaluation metrics used to present their results were precision, recall, AUC, and F1.

A transaction pattern analysis was performed by Toyoda et al. [45] to identify High Yield Investment Programmes (HYIP) . HYIP presents a fraudulent act where scammers offer high interest payments with minimal risk to potential investors. In the end, the HYIP collapses and the scammers collect the earned interest. To create a data set for the pattern analysis they collected both HYIP and non-HYIP related data, grouped the transactions by Bitcoin address, and finally conducted feature extraction to remove change of transactions and to calculate the transaction pattern. The classification was performed using XGBoost and Random Forest. The evaluation was conducted by the True Positive Ratio, False Positive Ratio, and F1 score.

Demertzis et al. [51] presented another anomaly detection framework. They used deep autoencoders to detect anomalous behaviours within a blockchain network. Their data set consisted of network transaction data that had their lower layer transmission data removed. For the data set, the Optimal Dataset Threshold (ODT) was determined, and the data were normalised. The evaluation results for the proposed method and the comparison to other methods, i.e., OCSVM, Isolation Forest, and Minimum Covariance Determinant, are depicted using RMSE, precision, recall, F1-score, and AUC.

5.1.2. Cryptojacking, Malware, and Security

Desai et al. [52] created BlockFLA, an accountable federated learning framework based on the Hyperledger Fabric blockchain. The main goal of their work is to detect backdoor attacks. BlockFLA's performance was tested by using trojan patterns on the CIFAR10 data set. An agent who corrupts the data set is considered an adversary. They presented their results by sampling from a Dirichlet distribution.

ContractWard [53] is a system that detects vulnerabilities within Ethereum smart contracts. They use a combination of the Synthetic Minority Oversampling Technique (SMOTE) and TomekLinks to deal with the class-imbalance problem in the training sets. The classification was conducted using five supervised learning techniques (eXtreme Gradient Boosting, Adaptive Boosting, Random Forest, Support Vector Machine, and k-Nearest Neighbor). Micro-F1 and Macro-F1 are used to present the measurements of the conducted tests.

Table 4. Security.

Method/Application	Backdoor Attacks	Vulnerability Det.	Cryptojacking	Security Analysis	Malicious user Det.	Botnet	Privacy Protection	Security Framework	Cyber Threat	Malicious Behaviour
Federated Learning	[52]									
XGBoost		[53]			[54]					
Neural Network		[55]		[56]						
Deep Learning		[57]						[58]		
Naive Bayes			[59]							
LSTM			[60]				[61]			
SVM				[56]	[62]					
Decision Tree				[56]						
Random Forest		[63]	[64]	[56]						
KNN					[62]					
Temporal Graph					[65]					
OCSVM						[66]				
Deep AutoEncoder									[67]	[68]

ALICIA, applied intelligence in the blockchain-based VANET, was presented by Maskey et al. [55]. They designed the system to provide vulnerability detection using

neural networks. An existing data set was used that contained car trajectories and vehicle telemetry (such as speed, acceleration, heading, etc.). The data set was additionally supplemented with simulated accident events. The results of their evaluation are shown on a graph containing the relationship between accuracy and the false positive rate.

Ashizawa et al. [57] presented Eth2Vec—a deep learning vulnerability detection system for Ethereum smart contracts. To evaluate their proposed system, they compared it to SVM. They collected smart contracts from Etherscan.io to create the data set needed for the evaluation. The evaluation was presented using precision, recall, and an F1-score for each, and it can be observed that Eth2Vec offers a better average performance.

Another vulnerability detection model for Ethereum smart contracts was presented by Song et al. [63]. To create a data set they collected source codes of smart contracts from the official Ethereum website, which were then labelled using Oyente. The feature extraction was preformed using the n-gram algorithm. To reduce the dimension of the data set, the opcodes were simplified by removing operands and classifying similar opcodes. The test was conducted on Random Forest, SVM, and KNN. The results are presented by calculating the F1-score, Micro-F1, and Macro-F1. The ROC curves of their models were presented additionally.

Cryptojacking is the process of adding covert malware that performs cryptocurrency mining on one's computer. The attackers use the victims' resources to collect mining rewards [59,60]. Liu et al. [60] proposed an approach to detect cryptojacking using Recurrent Neural Networks (LSTM). This approach uses the browser header data to detect malicious behaviour. The collected data are pre-proccesed so that the function features, suspicious data, and the function calling sequence are extracted and categorised (replaced by predetermined symbols). The evaluation results are shown, presenting the precision, recall, and F1-score with the addition of the hardware performance test results.

Another system used for detecting cryptocurrency miners using the NetFlow/IPFIX protocol was designed by Munoz et al. [59]. Data were first gathered from the Stratum traffic generator and analysed to identify the flows coming from mining traffic. Then, to create a training data set, traffic was captured from a large university campus network and flows were matched to those gathered from Stratum. The model was then trained with 677 samples on multiple Machine Learning algorithms (SVM, CART, C4,5 Decision Tree, and Naive Bayes). The results of the training were presented using accuracy, precision, recall, and an average F score. Additionally, the results of the method with the highest accuracy (Naive Bayes) were presented using a confusion matrix.

The dangers and descriptions of several types of malicious applications that affect Android devices were presented by Suleman et al. [69]. One of the listed malicious applications are cryptojacking applications. This means that Android devices, such as mobile phones and tablets, can also be affected by this type of malware software. Soviany et al. [70] presented the importance of the methodology when working on the detection of crypto-mining malware. They defined the exact steps in creating and testing such systems and the methodology for the presentation of the results. BrenntDroid, a tool that detects mining on Android devices was presented by Dashevskyi et al. [64]. They created a dataset by collecting potential Android mining applications and analysed their behaviours to determine whether they are truly miners. The dataset was filtered using scikit-learn library and the entries containing low variance were excluded. The features with high correlations were detected using the Pearson correlation coefficient and also excluded. The results of the Random Forest classifier were presented using ROC and AUC.

A machine learning model for smart contracts' security analysis was presented by Momeni et al. [56]. They collected the data for their data set from Etherscan, where they collected smart contract source codes. The source codes then had to be compiled, and the source codes were removed that were not version 0.4.18. After Feature Extraction, the data set was processed using four machine learning methods i.e., SVM, NN, RF, DT. For the presentation of their results they calculated the accuracy, precision, recall, and F1. The results were grouped by security problem and the method with the best results was shown

for each of them. They concluded that each specific problem yields a specific method and that there is no method that would be best for all scenarios.

Another method of anomaly detection was presented by Huang et al. [62]. Their main goal was to detect malicious nodes in the blockchain network. The data collected for this experiment consisted of time intervals of prepare and commit phases of nodes in different situations. They labelled the normal and abnormal data, and conducted the experiment using KNN, CNN, SVM, Gaussian model, and the Bernoulli model. The results are shown by displaying the relation between the accuracy and delay on several graphs.

Temporal graph properties to detect malicious accounts in permissionless blockchains were used by Agarwal et al. [65]. They performed an evaluation using the data obtained from the Etherscan API. Additionally, they used other sources to identify and label malicious accounts within the data set. To present the results of their experiment they calculated the precision, recall, F1 score, and MCC score. Additionally, they provided cosine similarity graphs to show the correlation between old and new malicious accounts, and the similarity between malicious and benign accounts.

Kumar et al. [54] created a system for detecting malicious accounts on the Ethereum blockchain. The used data set consisted of malicious and non-malicious addresses. A string comparison was used to filter out duplicate addresses, regardless of the case sensitivity. Additionally, addresses containing null transactions were also eliminated from the data set. Their method compares multiple supervised learning techniques: KNN, Decision Tree, Random Forest, and XGBoost.

A One Class Support Vector Machine classifier was used by Zarpelao et al. [66] to detect Bitcoin-based botnets. For the evaluation of their proposal they used an instance of the ZombieCoin botnet with six nodes. The botnet network was executed for an estimated two hours to collect the needed data. Additionally, legitimate data were collected from blocks appended to the main Bitcoin blockchain in three days. The obtained data were used to construct multiple experimental scenarios. The performance was measured using TPR, FPR, and AUC.

Adversarial Machine Learning was used by Yilmaz et al. [61] to enhance the privacy protection of grid users data. Their chosen method is based on the Long Short Term Memory (LSTM) model. The data set used in their evaluation was the Electricity Consumption and Occupancy data set, that contains power consumption readings and ground-truth occupancy information.

A security framework for IoT-based on deep learning and blockchain was proposed by Rathore et al. [58]. The MS COCO data set was used to evaluate their framework. They presented the results of their experiments by providing F-Score, accuracy, MCC, and AUC.

A system for the detection of cyber threats and situational awareness was proposed by Graf and King [67]. They used the deep autoencoder. The data were collected from the open source intelligence sources. The performance of the proposed model is shown by presenting a graph with the loss, accuracy, validation loss, validation accuracy, as well as the ROC space plot.

A deep autoencoder system for anomaly detection on the Ehtereum blockchain was proposed by Scicchitano et al. [68]. To evaluate their proposal, they used one synthetic and one real Ethereum data set. They presented the outlierness score on the real and synthetic data sets.

5.1.3. Data Processing

Social media behaviour detection is the focus of the study published by Liu et al. [71]. They trained the Isolation Forest algorithm on user login data from Carnegie Mellon University and focused mostly on the user's login time. Principal Component Analysis (PCA) was used to process the data and obtain the outliers.

Lin et al. [72] proposed a system that creates an address classification. They used a data set with 26,313 addresses to perform this classification. The transaction history was pruned, and only the relevant (direct) transactions of an address were added to the transaction

history summary. They tested multiple machine learning methods, and their tests show that LightGBM offers the best results. A confusion matrix was used for data representation.

Kanemura et al. [73] performed an identification of Bitcoin addresses by using a voting based classification method. Their goal was to detect which addresses are used by darknet market operators. To create a data set they extracted addresses, related and non-related to the darknet market, from multiple forums and sites. They expanded the collected addresses with the use of the address clustering heuristic, and compared the voting and non-voting identification methods. Data were presented by calculating the recall, precision, and F1 score of the results.

Table 5. Anomaly detection in data processing.

Method/Application	Social Media	Address Identification	Network Traffic	Performance Testing	Data Analysis	Behavioral Patterns	Transaction Clustering	Blockchain Simulator
Isolation Forest	[71]							
LightGBM		[72]						
Random Forest		[73]	[74]					
AWAP		[75]						
OCSVM				[76]	[77]			
AutoEncoder				[76]				
KMC					[77]			
Deep Learning					[78]			
Temporal Graph					[79]			
CML						[80]		
VGAE							[81]	
Neural Network								[82]
XGBoost								[82]

Wang et al. [75] proposed a detection model from Bitcoin address de-anonymising. They used a labelled data set, specifically: a five categories bitcoin address data set. Feature vectors were extracted from the historical transactions with using a parser. The tested algorithms were Logistic Regression, LightGBM, BAGC, CP, and AWAP. Their results were presented with the calculation of the accuracy, precision, F1 score, Jaccard, and NMI. The results were, additionally, summarised in four graphs.

Li et al. [74] performed an Ethereum behaviour analysis using multiple Machine Learning algorithms (Logistic Regression, SVM, KNN, C4.5 Decision Tree, AdaBoost, and Random Forest). They gathered the data with NetFlow traffic and obtained real communication relationships between the nodes in the network. The obtained data were later analysed using the passive traffic association analysis method. Two sets of features were extracted from the training set: features based on the statistical information, and features based on the graph information. The node2vec algorithm was also used for the graph representation. The precision rate and recall rate were used for the evaluation of the aforementioned machine learning algorithms. The results show that the best performance was given by the Random Forest algorithm.

Fan et al. [76] conducted performance analyses of machine learning methods on Bitcoin miners. They use multiple algorithms (LR, GB, RF, SVM, DNN, OCSVM, AE) and deployed them on real Bitcoin node implementations to test their training and testing latency. They used a data set for security detection, which consisted of normal and abnormal behavioural data. They concluded that the LR would be best for signature-based detection, but OC-SVM and AE would be best for anomaly detection.

Brinckman et al. [77] presented techniques and applications for crawling, ingesting and analysing data obtained from a blockchain. They used multiple machine learning methods to detect anomalies in the transaction behaviour, i.e., SVM, OCSVM, and K-Means clustering. The data were collected from websites that identify rogue accounts. Transactions for each account were clustered and the account features were extracted.

Patel et al. [78] presented a one-class graph deep learning framework for anomaly detection on the Ethereum blockchain. To create a data set, they collected the external transactions from the Ethereum blockchain, marked the anomalies manually, and extracted the needed features. They evaluated their system by comparing the results to the OCSVM and Isolation Forest models. The results were presented by showing the accuracy and F1-score for each of them.

Zhao et al. [79] made a temporal analysis for the Ethereum blockchain using Temporal Graph algorithms. To construct a data set, they extracted relevant data from the ethereum_blockchain data set that can be found in the BigqueryPublicData Repository. In addition to presenting the accuracy of the Random Forest and Logistic Regression, they also provided a visualisation of the temporal evaluation of the collected data.

Zola et al. [80] used Cascading Machine Learning to detect changes in entity behavioural patterns. The used two data sets: one from WalletExplorer, and the other contained Bitcoin data downloaded from the mainnet (from the last 3 years). The data set was cleared of unneeded types of transactions (lending), unlabelled and unusable data. The F1-scores were computed using k-fold cross-testing, and were shown together with their standard deviation. Multiple graphs were presented to show the different F1-score (bar charts) with regards to the batch size. Radar graphs show the precision, recall, F1-score, and number of samples for each batch size. Heatmaps show the F1-score of each test per batch size.

A variational graph autoencoder was used by Shah et al. [81] for transaction clustering and embedding generation. For their analytic framework they collected data from the Bitcoin blockchain (full node) and stored it in multiple NoSQL databases for easier processing. They presented the self-organising map output and explainable clustering for the retrieved data. The evaluation was conducted on the graph autoencoder, structural deep network embedding, and variational graph autoencoder. The results show that VGAE had the best result for both ROC and average precision.

Gouda et al. [82] presented BlockEval—a blockchain simulator where blocks are generated using Deep Learning techniques. Two methods were evaluated: Artificial Neural Networks and XGBoost. They were compared, and the performance of their models is presented in the form of their median transaction value, median fee, block size, and block count.

5.1.4. IoT and Sensors

An anomaly detection system for wastewater reuse was presented by Iyer et al. [83]. Hyperledger Fabric and multiple Machine Learning methods (polynomial regression, DBSCAN, autoencoders, and LSTM) were used to detect anomalies associated with water meter tampering. The blockchain is used to store all of the data obtained from the 2030 Wastewater Resources Group sensors that report data every hour. The data are then labelled as either anomalous or non-anomalous by each of the methods.

Belhadi et al. [84] used Reinforcement Learning to detect anomalies and faults in the smart grid. ITSA (Intelligent Time Series Anomaly detection) uses the CASAS (Center of Advanced Studies in Adaptive Systems) and OPSD (Open Power System Data) data sets injected with complex anomalous patterns to train the Reinforcement Learning models.

Table 6. Anomaly Detection in IoT.

Method/Application	Water Network	Electricity Network	IoT	Sensor Data	Transportation	Intrusion Detection	Manufacturing	Battery Health	Workflow Pattern	Health
DBSCAN	[83]									
Reinforcement Learning		[84]			[85]					
KNN		[86]								
LSTM		[87]							[88]	
Transfer Learning			[89]							
Deep Neural Network			[90]			[91]	[92]			
SVM			[93,94]							
BiLSTM			[95]							
AutoEncoder			[96]			[97]				
Deep Learning			[98]		[99]	[100]				[101]
GGM				[102]						
SLSTM					[103]					
GNN							[104]			
Isolation Forest								[105]		
GCN										[106]

An anomaly detection system for electricity consumption in smart grids was presented by Li et al. [86]. They use the k-Nearest neighbors algorithm with the combination of a data set collected from sensors. Their system is compared to DRAD (Distributed Real-Time Anomaly Detection in the networked industrial sensing systems) and ADSM (Anomaly Detection using Smart Meter data in the smart grid) and shows their successful detection rate regarding the anomaly occurrence rate.

LSTM-based privacy preserving framework for smart power networks was proposed by Keshk et al. [87]. They used two data sets: ICS Power Systems and UNSW-NB15 for evaluation purposes. They presented the accuracy vs. loss graphs and the accuracy after the application of their privacy preservation for both data sets.

A platform that manages crop growth and monitors crop diseases with the use of Blockchain technology and machine learning was developed by Pranav et al. [89]. The data set used for the training of the transfer learning model was the plant disease dataset. They presented the evaluation of their model using training and validation accuracy.

Liang et al. [90] proposed a Deep Learning based intrusion detection system for the IoT. They used the NSL-KDD data set, that contains different attack scenarios and classes. They evaluated the proposed system by using multiple settings (optimiser, init_mode, activation function), and presented the results through accuracy, average precision, average recall, and average F1-score.

Ethereum blockchain was used by Cheema et al. [93] to create an SVM based intrusion detection system for the IoT. The Bot-IoT data set was used to test the performance of the system given two scenarios: one using 10 features, and the other using 34 features. The results are presented with the ROC curves, accuracy, precision, recall, F1-score, and fall out.

Alkadi et al. [95] created an intrusion detection framework for the protection of IoT and cloud networks. The intrusion is detected by using the Bidirectional Long Short-Term Memory (BiLSTM) algorithm. They evaluated the framework by using the BoT-IoT and UNSW-NB15 data sets. The results were presented using the accuracy, training times, and testing times. Additionally, they compared the results with other Machine Learning techniques (SVM, RF, NB, MLO) by using both data sets.

Ngo et al. [94] created an IoT Botnet detection system based on the integration of static and dynamic vector features. Their data set consisted of both botnet and benign samples. Feature Extraction was used to reduce the data dimensions. The data were also standardised before they were considered to be ready for training and testing. The evaluation criteria used to provide the results of their experiment were: accuracy, precision, and F1. Additionally, they provided an ROC curve of the classifiers, PSI graph, and SCG.

Ali et al. [96] proposed a trust zone measurement architecture for blockchain based IoT systems. Their data set is comprised of various sensors in an IoT network taken from

the UCI Machine Learning Repository. This data set contains both malicious and benign data. The proposed work was tested, and they evaluated four machine learning algorithms: Autoencoder (their solution), Isolation Forest, SVM, and Local Outlier Factor. The results, presented by the accuracy and detection time, showed that their autoencoder method functioned best.

Sharma et al. [98] proposed a system that uses Deep Learning and blockchains to enable security in the industrial IoT. They used the Bot-IoT data set that contains labelled data of smart devices and multiple test scenarios. The evaluation of the model was performed using recall, precision, overall accuracy, and average accuracy.

Ide [102] created a collaborative anomaly detection system that focuses on noisy sensor data. The proposed system has multiple clients, and each of them has an individual data set that is a result of repeated measurements. The data were processed so that the data set was split into three equal blocks and each variable was standardised. Additionally, outlying samples of data were removed from the data set. A variable-wise anomaly score was calculated for each sample. The results were presented using a simple line graph.

Belhadi et al. [85] presented SS-ITS—a secure scalable intelligent transportation system. The system gathers and performs Feature Extraction on urban traffic data. The collected data are divided into different windows and processed accordingly. They used the local outlier factor to identify anomalies and extract the training data. For experimental results they used two urban data sets: ECML PKDD 2015 and HUMBI. They compared the proposed solution to baseline anomaly detection solutions (DILOF and MSCRED), baseline blockchain learning solutions (DRL and DUeling DQL), and baseline high-performance computing solutions (LoTAD and FUAD).

Deep learning and blockchain were used by Kim et al. [99] for secure and private dashcam video sharing. Their data set contained 519 images and 1093 sounds. The evaluation results were presented by calculating the accuracy, F-measure, precision, and recall. Optimal thresholds, image and sound detection comparison graphs, and overhead comparisons are visualised additionally.

Preuveneers et al. [97] created a chained anomaly detection model that uses deep learning (autoencoder). To evaluate their system, they used the CICIDS2017 data set. The data set contains network traffic information and common attacks. The evaluation was conducted on a real network where the data set was distributed among the nodes. They presented the results of the accuracy, loss, validation accuracy, and validation loss. They also presented the average epoch time comparison when using blockchain as a way of storing the weight updates and revised models, and when using the classical method of storage. The results show that when using blockchain, the latency is bigger.

Ferrag et al. [100] created DeepCoin, a deep learning (RNN) blockchain-based energy exchange framework. The evaluation was conducted by using multiple data sets: the CICIDS2017 data set, a power system data set and a web robot (Bot)-Internet of Things (IoT) data set. They presented the results in the form of accuracy, false alarm rates, detection rates, training time and test time. They also compared their proposition with SVM, Random Forest, and Naive Bayes.

Chen et al. [104] provided a GNN anomaly detection method on industrial time-series logs. They used a data set provided by SWaT, and represented the accuracy of their system proposal.

Liang et al. [91] proposed a deep learning based collaborative anomaly intrusion detection system. They used the KDD CUP1999 data set (the iris, lymphographic, vehicle, and glass data sets), and compared their results to other similar propositions. They presented the accuracy, validation time, average filtration efficiency, TPR, FPR, and the relationship between attack intensity and average hiding probability.

Jin et al. [105] proposed a blockchain-based data collection and anomaly detection for the estimation of battery state-of-health. The proposed system was evaluated using battery charging data provided by NASA. The collected data were processed so that the units and magnitudes became normalised. The normalised data were enlarged 100 times

and the collected data had to be in a range from 0 to 100. They use Isolation Forest to detect anomalies within the data, and showed the evaluation results using F1 and F2 scores. The results were, additionally, compared to several different algorithms, i.e., K-means, FCM, and PSO+FCM.

Jadidi et al. [92] presented an MS-DNN (Multi-Source Deep Neural Network) framework that detects anomalies within manufacturing systems. They validated the framework using two data sets: a factory automation data set and a SWaT (Secure Water Treatment) data set. The evaluation results are presented using precision, recall, F1, and accuracy.

An LSTM based anomaly detection framework was proposed by Xie et al. [88]. They used three data sets for evaluation purposes: the HDFS Benchmark data set, the HDFS data set and the oil industry data set. The performance of their model was evaluated presenting the accuracy, precision, recall and F1-score for each data set.

SP2F, an SLSTM (Stacked Long-Short Term Memory) privacy-preserving framework for agricultural unmanned aerial vehicles was presented by Kumar et al. [103]. Two IoT data sets, ToN-IoT and IoT Botnet, were used for evaluation purposes. The results were compared to two scenarios, one before the two-level privacy was applied to the data sets and the other after it was applied to the data sets. The results were compared to the Random Forest, Decision Tree, and Naive Bayes, and represented with accuracy, detection rate, precision, F1-Score, execution time analysis, confusion matrices, and ROC curves.

Drungilas et al. [101] evaluated two different implementations that are used for model validation. One implementation is based on the chaincode, and the other is a combination of chaincode and an Oracle web service component. They used two data sets for the evaluation of their proposed system, a generated synthetic 2D data set and an EEG eye state data set. Since the EEG data set was smaller than the synthetic 2D data set, it was expanded by bootstrapping the original data. The data on both data sets were indexed to improve the evaluation speed. Their model was evaluated on both data sets, and the results presented with the model inference runtime, distribution of runtime, and the overall overheads.

Wang et al. [106] proposed GuardHealth, a data management and graph convolutional network enabled anomaly detection system for healthcare. They evaluated the proposal by simulating malicious and benign nodes. Their model was compared to logistic regression and multilayer perceptron, and showed the average trust value and precision.

5.2. Fraud Detection

Fraud detection can be viewed as a subset of anomaly detection. Loosely speaking, fraud can be considered as a criminal activity with the intention of acquiring financial or any other gain [107]. We divided the results further into four main groups associated with fraud detection: financial, security, data processing, and IoT.

5.2.1. Financial Fraud Detection

KaRuNa is a blockchain-based framework for fraud cryptocurrency schemes created by Sureshbhai et al. [108]. Their model was based on the LSTM classifier. They used the Elliptic data set to evaluate the performance. This data set was enhanced by adding a classification score for the reduced raw cryptocurrency data obtained from social media, newsapi, and other web sources. The results are visualised with graphs depicting the analysis of tweets and a fraud scheme classification confusion matrix. The precision, recall, and F-score were also provided.

A multilayer perceptron architecture to detect cryptocurrency deception was presented by Dalal and Abulaish [109]. The data set for their evaluation was collected from the CMC website and labelled either legitimate or deceptive. The evaluation was conducted using Linear Regression, Softmax Regression, SVM, and MLP. The accuracy, precision, TPR, FNR, TNR, and FPR were presented, and it is observed that MLP performed the best.

Table 7. Financial fraud detection.

Method/Application	Cryptocurrency Deception	Phishing	Financial Fraud	Ponzi Schemes	Money Laundering	Poisoning Attacks	Scam Detection	Fraudulent Transactions	High Yield Investment Programmes	Eclipse Attacks
LSTM	[108]									
Multi-layer Perceptron	[109]						[110]			
Graph2Vec		[111]								
GCN		[112]								
AutoEncoder		[112]								
AdaBoost		[112,113]								
CNN			[114]	[115]						
Node2Vec			[116]							
Ordered Boosting				[117]						
XGBoost				[118]						
Random Forest				[119,120]				[121,122]	[123]	[124]
KNN					[125]					
Federated Learning						[126]				
SVC							[110]			
Logistic Regression								[127]		

An improved graph classification algorithm (Graph2Vec) for phishing detection on the Ethereum blockchain was proposed by Yuan et al. [111]. To create a data set they gathered phishing addresses from etherscan.io and also added the same number of normal addresses. They gathered the transactions for every address, removed the redundant data, and also removed the addresses with less than 10 transactions and more 300 transactions. They presented the evaluation of their algorithm by calculating the precision, recall, and F1-score. They also compared the performance to several other methods, such as node2vec, WL-kernel, and Graph2Vec.

A phishing scam detection system for Ethereum blockchain was presented by Chen et al. [112]. They used a graph convolutional network and autoencoder to detect phishing accounts. As a data set they used the Ethereum transaction history. They provided a performance comparison of their GCN method, Deep Walk, Node2Vec, and LINE. They showed the results of their AUC, recall, precision, and F1-score.

Zhou et al. [114] proposed a financial fraud detection method using deep learning (a Convolutional Neural Network). They gathered the data from a large O2O supply chain management platform to create the data set, and calculated the precision, recall, and F1-score of the experimental evaluation. Additionally, they compared their proposition to SVM and a decision tree.

Zhou et al. [116] proposed a financial fraud detection system by using Node2vec. To evaluate their proposal they used a data set provided from an Internet financial service provider in China. They compared Node2Vec, DeepWalk, and SVM, and presented their results by showing the calculated precision, recall, F1-score and F2-score.

Lou et al. [115] created an improved Convolutional Neural Network to detect Ponzi contracts. They obtained the data for the data set from etherscan.io. They collected the contracts and converted the hexadecimal bytecodes to the corresponding decimal number. Additionally, they standardised the bycodes. They performed the evaluation on their algorithm and several others (Decision Tree, SVM, XGBoost, OCSVM, Isolation Forest, Random Forest), and presented their corresponding precision, recall, and F-scores.

A Ponzi scheme is not a novel fraud. It is an investment fraud where the scammer pays the old investment clients revenue by using the investments of new clients rather than through legitimate business actions. In a blockchain environment this is done by using smart contracts [120]. Fan et al. [117] proposed a Ponzi scheme detection method. To create

a data set, they collected Ponzi and non-Ponzi scheme contracts from multiple websites. The contracts were converted from bytecode to opcode using the pyevmasm library and removed the operands. The opcodes were additionally converted to eigenvectors (using Bag Of Words - BOW) to conduct feature extraction utilising n-grams. BOW allows the definition of stop words, so that frequent operators can be removed from the opcode. They compared their method to multiple others by presenting the precision, recall, and F-score.

Machine learning was used by Chen et al. [120] to detect Ponzi schemes on the Ethereum blockchain. To create a data set, they collected smart contract source code from etherscan.io. and checked whether they were Ponzi scheme contracts manually. The features were then extracted without the course code, all the related transactions were collected and unsuccessful transactions were removed. Next, the contracts were converted from bytecode to opcode, the features were classified, and feature extraction was performed. Multiple algorithms were evaluated and combined, and their performance was presented using precision, recall and F-scores.

Chen et al. [118] used XGBoost to detect Ponzi schemes on the Ethereum blockchain. To test their system, they collected smart contracts from etherscan.io. The bytecodes were converted to opcodes and their frequency calculated. The contracts were labelled as Ponzi or non-Ponzi. The results were presented by calculating the precision, recall, and F-score.

Machine learning methods were used by Bartoletti et al. [119] to detect Ponzi schemes on the bitcoin blockchain. To create a data set they collected bitcoin addresses related to Ponzi schemes and their respected transactions. They extracted features that could be useful to detect Ponzi schemes. Additionally, the data set was also filled with a number of addresses not connected to Ponzi schemes. To create an evaluation, they selected several machine learning classifiers: RIPPER, Bayes Network, and Random Forest. They calculated their accuracy, specificity, sensitivity, precision, F-measure, G-mean, and AUC. The results were visualised using confusion matrices.

Baek et al. [125] proposed the detection of money laundering with Ethereum cryptocurrency transactions. To create a data set they collected wallets from etherscan.io and extracted the wallets with the largest trading volumes. For the minimisation of data, they chose the expectation maximisation algorithm, and the k-means algorithm for the clustering and weight defining. To present the results they calculated the accuracy, precision, F-measure, and True Negative Rate. A ROC curve and Precision Recall Curves were used for visualisation purposes.

A federated learning framework was used by Liu et al. [126] to detect poisoning attacks. For the evaluation they used the MNIST and CIFAR-10 data sets. The performance of their model was presented by calculating accuracy for different numbers of participants, and the percentage of modified labels that indicate the strength of the poisoning attack.

Badawi et al. [110] used machine learning classifiers to detect scams within a bitcoin blockchain. They searched for bitcoin generator scams with multiple search engines: Google, Bitcoin.fr, CuteStat.com, and the Internet Archive. They included multiple classifiers for evaluation purposes. The results were presented by calculating precision, recall, and the F1 score. It shows that SVC and MLP provided the best performance.

Bhowmik et al. [127] presented a comparative study of machine learning algorithms used for fraud detection in blockchain networks. They used the node2vec algorithm to collect data for the data set. Features were then extracted from the collected data and stored in a CSV file. The CSV was then converted into a dictionary using the node2vec algorithm. A network edge list file was created and the embedding dimensionality reduced. Additionally, the features had to be normalised (the value 1 was assigned to fraudulent transactions, and 0 for the others), the mean and standard deviation were calculated. The results are shown by the achieved accuracy of each algorithm, and it was observed that logistic regression performed the best.

A security enhancement to financial transactions in the bitcoin blockchain was offered by Boughaci and Alkhawaldeh [121] using machine learning. They used the Elliptic data set and the k-means clustering technique to partition unlabelled data. The measurements

were made by using four machine learning algorithms: the Naive Bayes, Bayes Network, AdaBoost, and Random Forest. The precision, recall, TP rate, FP rate, PRC, and area under the ROC curve were calculated. The results showed that Random Forest had the best performance out of the selected algorithms.

Lee et al. [122] used machine learning to detect illegal transactions on the bitcoin blockchain. They collected hash lists of legal and illegal transactions from multiple websites (such as Silk Road and Blockchain Explorer) to create their data set. The testing was conducted on the artificial neural network and random forest classifier. The F1-scores of these two methods show that random forest was a better fit for this type of detection.

Wen et al. [113] proposed a framework used for the detection of phishing scams on the Ethereum blockchain. They collected data from Etherscan and added three filer rules to remove accounts with low activity levels, i.e., removing the smart contracts accounts, removing accounts with less than 5 transactions and transfer-in transactions with less than four, and removing accounts whose greatest balance was less than five. The testing was conducted on multiple Machine Learning models including SVM, KNN, and AdaBoost. For each model the precision, recall, F1-score and AUC were presented, and it was concluded that AdaBoost performed best.

A novel methodology for the detection of high yield investment programmes Bitcoin addresses was proposed by Toyoda et al. [123]. The data were collected by searching for HYIP addresses and collecting their transactions. Feature extraction was then conducted, unneeded parts of the transaction were removed, the BTC was converted to USD, and the transactions were labelled as spent, received, or Coinbase. The evaluation was conducted on multiple algorithms (RF, XGBoost, Neural Network, SVM, k-NN) and the results were shown as TPR and FPR. The best result was provided by Random Forest.

Xu et al. [124] used the Random Forest classifier to create a detector for eclipse attacks for the Ethereum blockchain. Eclipse attacks are used to isolate a certain user from a network by controlling their outgoing connections. In order to collect data for the data set, they collected the UDP packets from normal and unsolicited nodes. The data were then converted into a readable format using the Ethereum UDP packet dissector and added into the data set. They evaluated their proposition and presented the results for the Random Forest classifier in the form of its precision, recall, F-score, and support.

5.2.2. Cryptojacking, Malware, and Security

Abdulqadder et al. [128] created an intrusion detection system to mitigate attacks in an SDN/NFV enabled cloud. Their method used a Recurrent Neural Network to detect flow features. They used a network simulator (Ns3) and compared their proposed model to the k-nearest neighbors algorithm by calculating the precision, recall, accuracy, detection rate, and processing time.

Liu et al. [129] provided a classification and sharing method of malware that uses threat intelligence. Their method is based on the Broad Learning network. The Kaggle's malware classification data set was used for evaluation. Data were preprocessed in order to convert the malware data from binary to hexadecimal, and then convert the hexadecimal values into a matrix to create a grey scale image. They compared the proposed algorithm to several other algorithms (k-nearest neighbor, Random Forest, and a Convolutional Neural Network) using accuracy and duration dependent on the image size.

A decentralized firewall that uses Deep Belief Neural Networks to detect malware was proposed by Raje et al. [130]. The data set used for evaluation was a combination of the MALIMG data set (for malicious data) and vanilla windows installations (for the benign data). They presented the results by showing the accuracy and TPR.

Deep Recurrent Neural Networks (LSTM) were used by Yazdinejad et al. [131] to detect cryptocurrency malware. Their data set is comprised of real-world cryptocurrency malware samples and benign samples. They extracted the scripts of each file and created samples of the original code. The operators, operands, and memory addresses were removed from the data set. They conducted the evaluation of different LSTM configurations

and provided their accuracy, and comparison to other ML classifiers (SVM, Naive Bayes, Decision Tree, KNN, MLP, AdaBoost, Random Forest).

Table 8. Security.

Method/Application	Cloud	Malware	Cryptojacking	Miner Detection	Intrusion Detection	Privacy Protection	Malicious User Detection	Malicious Activity	Video Copyright
RNN	[128]								
BLS		[129]							
DBNN		[130]							
LSTM		[131]			[132]				
Deep Learning		[133]							[134]
Naive Bayes		[135]					[136]		
ATT-LSTM			[137]						
Random Forest			[138–141]				[142]		
SVM			[139,143,144]		[145]				
KNN			[146]						
Gradient Boosting			[141,147]				[148]	[149]	
Neural Network			[141]						
Decision Tree				[150]					
XGBoost						[151]	[152]		
Supervised Learning							[153]		
T-DSNE							[154]		
Bagging							[148]		

A deep learning model for the detection of malware on the Quorum chain was presented by Gao et al. [133]. They compared their new model to other algorithms, such as Decision Tree, k-NN, Logistic Regression and SVM. The results were presented using their precision, recall, F1-score, and z-values.

Kumar et al. [135] proposed a system for malware detection on Android IoT devices. They used a data set composed of both benign and malware applications. The data were collected from the Google Play Store and Chinese App store. They conducted the evaluation on several machine learning algorithms, i.e., Improved Naive Bayer, SVM, KNN, Naive Bayes, and DBN. The results were presented using TPR, FPR, and accuracy. The best results were given by the Improved Naive Bayes algorithm.

Vesely and Žadnik [150] focused their work on the detection of cryptocurrency miners. They used a data set that was collected in the Czech National Research and Educational Network, and subnets of three major institutions. The data set contained mining and non-mining clients, and was annotated accordingly. The results were presented using cumulative normalised distribution functions and confusion matrices.

A deep learning approach for detecting cryptomining malware was presented by Databian et al. [137]. They evaluated LSTM, attention-based LSTM and Convolutional Neural Networks. In order to create their data set they collected the cryptominer samples from virustotal.com and removed all the inactive samples. The evaluation of the aforementioned methods is shown by presenting their accuracy, precision, recall, F-measure, MCC, and FPR. The best results were given by ATT-LSTM.

Machine learning was used by Caprolu et al. [138] to detect cryptojacking. The Random Forest algorithm was selected as the most appropriate for this task. They tested the proposed method on multiple scenarios: a baseline example that simply monitors the traffic on the client, the detection of full nodes, detection of miners, and sponge-attack detection. All results were presented by calculating the F1-score and using AUC curves.

Gangwal et al. [139] proposed a machine learning based system for the detection of covert cryptomining. They collected events and information about the performance of computers (processor events, hardware events, software events, and hardware cache events). In the case of missing values, they were replaced with the mean of the associated event. They trained and evaluated two machine learning methods, i.e., Random Forest and

SVM. The testing was conducted on multiple scenarios, and the results were presented using accuracy, precision, recall, F1, and confusion matrices.

A solution to detect cryptojacking using magnetic side-channels and machine learning was presented by Gangwal and Conti [146]. They used two different laptops to collect the data for the data set. They used laptops to conduct cryptominning and profiled the events. In addition to the hardware and software measurements, they also measured the generated magnetic fields. Before the data could be used for training and testing, a scaling function had to be used to normalise the input data. They tested the KNN classifier, and presented their results using confusion matrices, full-stack classifications, accuracy, precision, recall, and F1-score.

Mansor et al. [147] compared the use of machine learning algorithms to detect cryptojacking. They tested the performance of Random Forest and Gradient Boost on a data set with both malicious and benign applications. Their results showed the confusion matrices and TP/FP rates for both algorithms.

A system that detects cryptomining malware using machine learning and deep learning was proposed by Pastor et al. [140]. They used Mouseworld to generate the needed data. Additionally, they used the DS1 data set. Multiple machine learning models were evaluated (FCNN, Random Forest, Logistic Regression, CART, and C4.5). After presenting their F1, precision, recall, accuracy, AUC ROC, AUC P-R and confusion matrices, it was observed that RF, C4.5, and FCNN performed well.

MineCap: An incremental learning method for cryptojacking detection was presented by Neto et al. [141]. They used mining pools running on specific TCP ports to collect the data needed for the data set. After the data were collected, unnecessary information was removed (source IP, destination IP, source port, destination port, transport protocol). They evaluated multiple classification algorithms, i.e., Random Forest, Logistic Regression, Gradient Boosted Tree, Naive Bayes. The results were presented using a graph with the ROC curve, and a graph containing the precision, sensibility, and specificity. Additionally, more graphs were presented that showed the accuracy of the ML algorithms.

Kharraz et al. [144] created OUTGUARD—a system that detects in-browser covert cryptomining. To construct their data set they collected the blacklist pattern information from CoinBlockerList, NoCoin, and minerBlock. They then gathered websites that contained JavaScript libraries matching the blacklist patterns. They used Wappalyzer to label the cryptojacking libraries and also added non-cyprojacking websites to the data set. Lastly a set of features was extracted including: JavaScript execution time, JavaScript compilation time, garbage collection, Iframe resource loads, CPU usage, etc. To evaluate the proposed system, they presented the score ratio based on the feature, and TPR and FPR ratio graph.

Yang et al. [132] proposed a spam transaction attack detection model that is based on Deep Learning and LSTM (GRU and WGAN-div). The data set was created by using the bitcoin sound code and simulating the needed environment. The results were presented with an accuracy and false alarm rate, and compared to ADvISE, SVDD, and OC-SVM.

Deebak and Al-Turjman [151] used machine learning to measure privacy protection and cyber risks. Multiple machine learning algorithms, i.e., XGBoost, Nearest Neighbor, SVM, and Decision-Tree were used to detect fraudulent behaviour. The data set used for testing purposes was collected from an insurance company. The detection was focused on whether the claims were fraudulent or not. The performance was measured using accuracy, precision, recall, F1-score and training time.

A supervised learning model that can be used to identify illegal activities in the bitcoin blockchain was created by Nerurkar [153] et al. The data set was collected from the VJTI Blockchain lab, and the raw data were converted to CSV files. The necessary features were extracted and multiple hash addresses (from a single entity) were grouped by using multi-input heuristic clustering. The experimental study of their approach was conducted comparing the performance of SVM, LogReg, XGBoost, Random Forest and their custom proposed model. The results were presented by calculating the precision, recall, and F-

score, and by multiple graphs showing the scalability, learning curves, and performance of each method.

A method for the detection of intrusion and DoS attacks on E-voting systems was presented by Cheema et al. [145]. They used the UNSW-NB15 data set to train and test two SVM classifier models (Gaussian and Linear). The evaluation was made using accuracy, area under the curve, and prediction speed.

A cryptojacking detection method using machine learning was presented by Nukala [143]. He tested KNN, Random Forest, Decision Trees, SVM, and Naive Bayes. The data set consisted of cache hits and misses, and the performance was presented using the models accuracy, precision, recall, and F1-score. The best F1 score was given by SVM.

The T-distributed stochastic neighbour embedding was used by Sun et al. [154] to detect malicious user activity on Ethereum. They used an existing data set, and extracted the ones that could be associated with malicious behaviour. Node clustering was employed to detect such behaviour. The performed work was presented using Eigenvector visualisation.

Supervised machine learning was used by Ostapowicz and Zbikowski [142] to detect fraudulent accounts on the Ethereum blockchain. Data were obtained from Etherscan.io, and the empty wallets were removed (the ones with no transactions). The evaluation included three machine learning classifiers (Random Forest, SVM, and XGBoost). The probability specificity, recall, precision, FPR, F1, and confusion matrices were presented for each of the evaluated methods. Random Forest obtained the best results.

Farrugia et al. [152] presented the detection of illicit accounts on the Ethereum blockchain by using XGBoost. They created the data set by collecting the data from the Etherscamdb and a local Geth client. They collected both normal accounts and those labelled as illicit. The data were filtered by removing the duplicate accounts, their transactions were gathered using Etherscan API, and removing unsuccessful transactions. The data were visualised utilising a 2D and 3D t-SNE scatter plot. To evaluate their proposal, they calculated the accuracy, sensitivity, specificity, F1-score, and AUC for multiple scenarios. They also provide a graph with the average logarithmic loss, classification error, and a confusion matrix.

A method for the detection of suspicious users was proposed by Mittal and Bhatia [136]. They used two data sets to evaluate their system: Bitcoin-OTC and Bitcoin-Alpha. Multiple machine learning techniques were evaluated, such as SVM, Naive Bayes, Decision Tree, and Neural Networks. They presented the results of the evaluation providing the precision, recall, F1-score, support, and accuracy from each machine learning algorithm, and for each data set.

A supervised learning model to identify illegal activities within the bitcoin blockchain was presented by Nerurkar et al. [149]. The data set was taken from the VJTI Blockchain lab and converted to CSV files. They evaluated the proposed model on multiple classifiers (SVM, Logistic Regression, XGBoost, and Random Forest). The results of the valuation were presented with a several performance variables (like AUc, accuracy, sensitivity, detection rate, kappa, P-value, etc.), confusion matrices, CPU and RAM utilisation, learning curves, scalability graphs of the models, and performance graphs of the models.

An estimate of the proportion of malicious entities in the bitcoin system was proposed by Sun Yin and Vatrapu [148]. They used supervised machine learning. The data set consisted of categorised and uncategorised data for every cluster in the blockchain environment. Data were cleaned from all of the empty cells (values depending on the cell type were inserted in the empty cells—0 for integers, 0.0 for float, and the string values depended on the column). Manual feature extraction and feature engineering was conducted after the data set was cleared of missing values. Multiple classifiers were tested and presented using mean CV-Accuracy and SD. Gradient boosting and bagging proved to have the best performance, so they were chosen for further research.

Chen et al. [134] created a decentralised autonomous video copyright protection system based on blockchain. They evaluated their system using the VCDB data set, and presented the dimension, recall, and query speed.

5.2.3. Data Processing

The LightGBM algorithm was used by Jourdan et al. [155] to characterize entities in the bitcoin blockchain. For testing purposes, they gathered addresses and their labels from WalletExplorer. Additionally, they applied common spending heuristics and transitive closure operations to the labelled data set. They evaluated their decision tree algorithm, and compared the results (accuracy, F-1, and precision) to the logistic regression algorithm.

Jan et al. [156] used deep learning for integrity verification and behavioral classification. They created a data set by downloading benign applications from the Google Play Store and malicious applications from VirusTotal. They captured their behaviour logs and labelled the data in the data set. The results of the evaluation are presented with the accuracy, precision, recall, F1-score, and ROC curves.

Linoy et al. [157] used machine learning for the deanonymisation of addresses within the Ethereum blockchain. They collected verified contract data from etherscan.io. The main focus was on contracts written in Solidity. For easier parsing, they converted the bytecodes into opcode. Each contract was split into its individual components and refined before the feature extraction.

Table 9. Data processing.

Method/Application	Entity Characterisation	Integrity Verification	Anonymity	Fake News	Miner Reputation
LightGBM	[155]				
Deep Learning		[156]			
Random Forest			[157]		
Graph Embedding				[158]	
Linear Regression					[159]

Hamdi et al. [158] used graph embedding to detect fake news on Twitter. They combined multiple sources (ego-Twitter, Twitter API, CREDBANK) to create their own data set. After the data were combined, they used NetworkX to a graph that could be used to train the classification model. The results were shown by Micro-F1 and Macro-F1 graphs, SBM visualization using t-SNE, accuracy, precision, recall, F1-score, and AUC ROC.

Kaci and Rachedi [159] proposed a machine learning method to manage a miner's reputation. To evaluate their proposed solution, they created a data set that is composed of mining history information. The evaluation of the proposal was compared to linear regression, SVR and MLP. The results were presented using the accuracy and training time.

5.2.4. IoT and Sensors

Ding et al. [160] proposed a multiple object tracking system using HashNet from deep hash extraction. They used the MOT15 data set for the evaluation and acquired multiple results (mostly tracked agents, mostly lost agents, False Positives, False Negatives, identity switches, multi-object tracking accuracy and multi-object tracking precision).

AIT is an deep learning based trust management system for vehicular networks proposed by Zhang et al. [161]. To create a data set, they used SUMO (Simulator of Urban MObility) to generate maps and vehicular network simulations. Their model is based on the Feedforward Neural Network, and for the evaluation (precision, recall, percentage of malicious nodes, and accuracy) it was compared to the Recurrent Neural Network and Convolutional Neural Network.

Table 10. IoT fraud detection.

Application/Method	Object Tracking	Vehicular Network	Target Detection	IoT	Outlier Detection	Sybil Attacks	Insurance Systems
Deep Hashing	[160]						
Deep Learning		[161]	[162]				
Federated Learning		[163]					
GAN				[164]			
Supervised Learning					[165]		
Isolation Forest		[166]			[165]		
KNN						[167]	
XGBoost							[168]

Liu et al. [163] used blockhain and Federated Learning for intrusion detection in vehicular edge computing. They used the KDD Cup99 data sets of edge vehicles to test the proposed system and represent the precision rate, recall rate, and accuracy rate changes with respect to data size.

Zhang et al. [162] proposed a target detection and automatic monitor scheme based on blockchain and deep learning models. They used the CIFER-10 and Mnist data set to conduct the performance evaluation of the proposed model. The results showed the training accuracy and loss.

Hao et al. [164] used Generative Adversarial Neural Networks to detect fraudulent behaviour in the IoT. They prepared two sets of data: one set for the digital signature frauds (containing messages, private keys, and public keys), and another data set for asymmetric encryption frauds (plaintext, private keys, and public keys).

Supervised machine learning for outlier detection was used by Salimitari et al. [165]. They created a simulation of an IoT network with 100 sensors and collected their data. The performance was presented using fault tolerance and accuracy.

BITS: A blockchain based intelligent transportation system was proposed by Maskey et al. [166]. They used machine learning to detect outliers within the system. Simulated data were used from multiple data and randomly injected 10% outlier values. They presented the outcome of the Isolation Forest model using a graph that included the accuracy and false positive rate.

A multi-level trust mechanism against Sybil attacks in vehicular networks was presented by Haddaji et al. [167]. They tested the system with three different machine learning algorithms: SVM, KNN, and Random Forest. The algorithms were tested using the VeReMi data set that contains multiple types of attacks: Constant attack, Constant offset attack, Random attack, Random offset attack, and Eventual stop attack. They presented the accuracy and time consumed per test for each of the selected algorithms, and showed that KNN gave the best ratio of accuracy and consumed time.

Dhieb et al. [168] presented a system for fraud detection and risk measurement in the Insurance sector. For their experiment they used four machine learning classifiers (Decision Tree, SVM, Nearest Neighbor, and XGBoost) on a data set obtained from an insurance company. They calculated the accuracy, recall, precision, and F1-score, and showed that XGBoost performed the best. Additionally, they provided the normalised confusion matrix for XGBoost.

6. Discussion

Following the previous section which depicted the features of applications considered in our study systematically, we provide an overall discussion and extraction of key elements that define the taxonomy of data mining methods used for analysing blockchain data. We defined the following levels:

- Level 1: Data extraction,
- Level 2: Data preprocessing,

- Level 3: Data mining,
- Level 4: Evaluation and visualisation of results.

The first level is tailored to the raw data that are stored in blockchain. Here, we are confronted with data retrieval from blockchain. As we have already mentioned, blockchain technology, where a set of valid transactions form a block and a set of blocks that satisfy the consensus protocol that are added to the ledger, brings the benefits of transparency, immutability and consistency of data [21]. Nevertheless, the features that offer these benefits are the ones that include several challenges with regard to data management. Searching and retrieving data in blockchain-based systems is not straightforward. It is time and money consuming, since it requires additional programming efforts. Blockchain is optimised for storage and not for searching and retrieving data as is the case with traditional databases. Therefore, the biggest obstacles to enabling the efficient retrieval are: decentralisation and data distribution, lack of query language, data confusion and entanglement and limited APIs [21,169]. At the moment there are several efforts underway to try to provide more efficient and reliable data access, such as supporting faster querying using a centralised indexing server to copy blockchain data (e.g., Etherscan), or proposing an SQL-like query language (e.g., Ethereum Query Language (EQL)) to provide general purpose querying [21]. Let us mention that many of the research papers skipped the step of data retrieval, due to using publicly available datasets where this step had already been performed.

In addition to the raw data stored in the blockchain, in some studies [74,124,141] raw network traffic was collected to detect certain anomalies within the blockchain network. Specifically, the needed traffic information was added to the data set and, if necessary, converted into different formats. On the other hand, some created network and scenario simulations to generate the needed data [59,66,76,106,128,132,140,150,161,165,166]. The simulations included malicious nodes, traffic flows, etc. Smart contracts were obtained from sources such as HONEYBADGER [49], Ethereum [48] and Etherscan [56,57]. The collected smart contracts are mostly in bytecode, which is converted into opcode for further processing [49,117,118,157].

The data are stored mostly on the blockchain, or collected from the blockchain or its network. Some authors [83,97] used blockchain to store the data collected from a data set or sensors, while Preuvenees et al. [97] also tested the storage of data on the blokchain versus using a classical method of storage, and their results showed more significant latency when using blockchains. Additionally, NoSQL databases were used to store the data retrieved from the blockchain for easier processing [81], while Drungilas et al.[101] tested whether it was better to keep all the data on the chaincode, or to combine the chaincode with the Oracle web service. Another way to store data is using a CSV file [127,149,153] because of its simplicity and ease of further processing.

The second level deals with the preprocessing of the retrieved data. Usually, data preparation is one of the most complex processes, that involves data cleaning, missing data estimation, feature selection, and several data transformations.

Data cleaning involves the actions of filtering and excluding data that cannot be used or is irrelevant. Data that are excluded can include irrelevant fields [45,46,72,80,141,152], invalid fields [42,120], inactive accounts [46,111,113,142], duplicate addresses and data [54,111,152], outliers [102], and data with unknown labels [40,80]. Additionally, missing values can be filled with a median value of the column [42,139] or with certain default values [148]. Peak values can be eliminated by subtracting the column mean and dividing it with the standard deviation [42]. Data can be further normalised [51,105,127,146] and, when working with smart contracts, the operators and operands can be removed from the opcodes [63,117,131]. When working with transactions, if needed, the data can be grouped by user or address [42,77,111,153]. If the data set seems to be too small, it can be expanded by bootstrapping the original data [101].

To be able to work with the collected data, they should be labelled [118,123,148,150,156]. The labelling can either be done manually [48,78,120] or automatically [44,59] by using certain tools, like Oyente [63] or Wappalyzer [144].

Feature extraction can be done by using several tools, like the bi-gram features [49], RFM features [42], the tsfeatures library [46], or the n-gram algorithm [63,117]. The mutual influence of features can be calculated using Pearson's correlation [42].

The third level is devoted to the selection of the data mining method. The selection of data mining is done mostly by conducting literature reviews and research. Most of the articles included in our study also evaluated multiple methods. The methods were either selected to show the performance of their custom solution, or were evaluated to choose the best method for a specific problem. Figure 4 shows the trends of using various methods over the past five years. In the first years that were considered in this study, the authors utilised mostly conventional machine learning methods, e.g., Random Forests. However, a very interesting trend appeared in recent years, where the use of deep learning methods was in the majority, which is not surprising due to the popularity of deep learning [37].

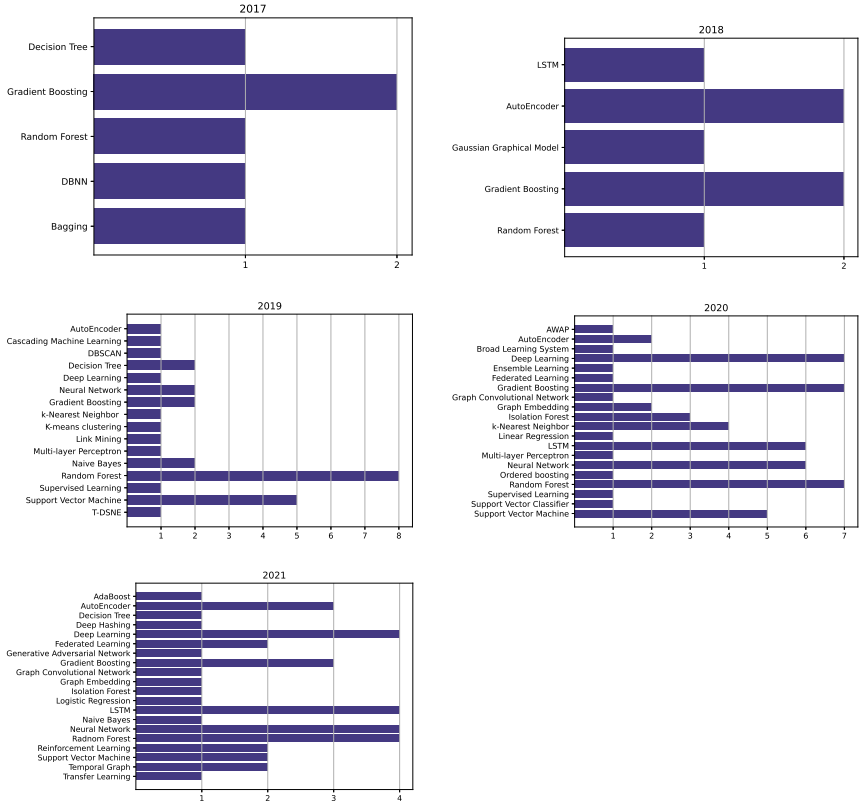

Figure 4. Distribution of used methods per year.

The taxonomy is concluded with an evaluation of results, as well as a visualisation of the obtained results in level 4.

Most of the evaluation results given by the reviewed publication were given by presenting the precision, recall, F-score and accuracy. Other metrics include the processing and training time, the Micro and Macro F-score [53,63], fall out [93], loss [67,87,97,103,162], Jaccard, NMI [75], True Positive Rate (TPR) and False Positive Rate (FPR) [45,66,91,100,121,123,135,144,147], Kappa value [43], and MCC score [58,65,137].

The visualization was done by presenting the ROC curves [40,51,63,67,81,93,94,103, 121,125,140,141,156], AUC [49,58,66,93,112,113,138,140,152], ROC-AUC [44,158], the t-SNE algorithm [42,158], 2D and 3D scatter plots [42], confusion matrix [50,59,72,103,108,119,139, 140,142,146,147,149,150,152,168], detection frequencies [50], PSI graph and SCG [94], the

Dirichlet distribution [52], Precision Recall Curves [125], Radar graphs and Heatmaps [80], and other suitable graphs.

7. Where Are We Now, and What Follows?

A review of the papers published in the last five years revealed the trends and facts of synergy between data mining and blockchain technology. According to the study, numerous methods were proposed, used and utilised to intelligently analyse data stored in blockchain, focusing on anomaly detection, implying the popularity and importance of this field and research that will further explore the potential of this synergy.

Based on the facts presented so far, here, we summarise what are to be the directions of the further development of this research field, to what part of this synergy researchers should focus their research investigation, as well as what the issues and challenges yet to be explored are.

7.1. The Importance of Synergy

With the implementation of blockchain solutions in different application domains, different systems will need to be developed for controlling the content. Researchers will have a lot of opportunities to develop methods for the analysis of data stored in blockchain, since cryptocurrencies will be inevitable in the future. Countries and different agencies will have to control this aspect, including money flows, preventing money laundering [170], as well as controlling potential terrorism-sponsoring [171]. Another important aspect lies in smart contracts, which are already being implemented in different domains [172,173], such as the insurance industry, healthcare, land registry. We expect that there will be an expansion of its usage in the future. Therefore, it will be essential to detect anomalies in these contracts and avoid potential fraudulent behaviour [174]. Another approach that might be decisive for Industry 4.0 lies in using data mining methods as an active structural component of the blockchain. This will strengthen blockchain networks and address security issues, which emerge in those environments such as information protection and industrial confidentiality [19,51]. Those approaches will offer a timely behaviour prediction and optimal decision-making in dynamic environments.

7.2. Challenges

Research of papers, on the one hand, revealed interesting trends which suggest that most of the solutions are prototypes and proofs-of-concept. Some research papers also proposed several solutions which are basically only ideas without their practical evaluation using proof-of-concept. Therefore, there is still a long road to ensure quick flow or transition from prototypes to real applications.

On the other hand, research exposed several challenges where much more devotion should be given in the future, especially in the design of datasets, experiments, test cases or scenarios and implementation of algorithms. Researchers should also explore further ways for automatization of some preprocessing steps [148], while expanding and enlarging the datasets [56,90,106,109,118]. This aspect should also be at the centre of the research, since more complex and expanded datasets should definitely contribute to more accurate anomaly detection and potentially result in faster decisions. Also, the datasets should be kept up to date to include new frauds and types of attack. Some researchers reported that the developed methods also use a lot of computational power [126] or communication cost of propagating a new block to all participants in the network [95]. The need for simulation in real world scenarios was also reported in paper [90], as well as the creation of realistic test cases and experiments [140]. Additionally, enhancing methods with different types of fraud detection is also one important research direction [117]. Finally, using additional optimisation methods, i.e., metaheuristics [121], should also be a fruitful direction in improving the existing algorithms.

8. Conclusions

In this paper, we have reviewed recent studies that explore the synergies of blockchain technology and data mining techniques for anomaly and fraud detection. These two applications were detected as the most fruitful ones for possibly applying data mining methods on blockchain data. The aim of this review was to analyse the current trends in exploiting the synergies of blockchain technology and data mining techniques for anomaly detection, while discovering all the main machine learning methods and constructing a taxonomy of those methods used to enhance the blockchain technology for specific purposes.

A review of the data mining methods used during the last five years revealed a tendency in this research area. In the first two years the dominant method used was Gradient Boosting. SVM and Random Forest are two methods used consistently in the studies throughout this five year period. Nevertheless, we can observe that these two methods were offering the best results predominantly among studies published in 2019 and 2020, whereby Random Forest is also predominant in 2021. Nevertheless, we can see a new tendency in the last two years going towards the use of Neural Networks, Gradient Boosting, Deep Learning and LSTM. There are also some future challenges in this domain. It would be interesting to explore the maturity of the proposed ideas and the flow of knowledge from research papers to real-world applications. Additionally, researching the opportunities of Automated Machine Learning (AutoML) in this domain may also be a fruitful direction.

Author Contributions: Conceptualization, A.K. and I.F.J.; methodology, A.K. and I.F.J.; validation, A.K.; data curation, R.K.; literature search R.K., literature review, A.K. and R.K.; writing A.K., R.K. and I.F.J.; visualisation, A.K. and R.K.; supervision, A.K. and I.F.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Slovenian Research Agency (Research Core Funding No. P2-0057) and the European Union's Horizon 2020 Research and Innovation Program under the Cybersecurity CONCORDIA project (GA No. 830927).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Acknowledgments: The authors acknowledge the financial support from the Slovenian Research Agency (Research Core Funding No. P2-0057) and the European Union's Horizon 2020 Research and Innovation Program under the Cybersecurity CONCORDIA project (GA No. 830927).

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

Abbreviations

The following abbreviations are used in this manuscript:

AE	AutoEncoder
AWAP	Adaptive Weighted Attribute Propagation
BLS	Broad Learning System
CML	Cascading Machine Learning
DBNN	Deep Belief Neural Network
DNN	Deep Neural Network
GAN	Generative Adversarial Network
GB	Gradient Boosting
GCN	Graph Convolutional Network
GGM	Gaussian Graphical Model
GNN	Graph Neural Network
KMC	k-Means Clustering
KNN	k-Nearest Neighbor
LR	Logistic Regression
MLP	Multi-layer Perceptron

NN	Neural Network
OCSVM	One-class Support Vector Machine
RF	Random Forest
RNN	Recurrent Neural Network
SVM	Support Vector Machine
T-DSNE	T-Distributed Stochastic Neighbor Embedding
VGAE	Variational Graph AutoEncoder
XGBoost	eXtreme Gradient Boosting

References

1. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 13 July 2021).
2. Li, Z.; Zhong, R.Y.; Tian, Z.G.; Dai, H.N.; Barenji, A.V.; Huang, G.Q. Industrial Blockchain: A state-of-the-art Survey. *Robot. Comput. Integr. Manuf.* **2021**, *70*, 102124. [CrossRef]
3. Casino, F.; Dasaklis, T.K.; Patsakis, C. A systematic literature review of blockchain-based applications: Current status, classification and open issues. *Telemat. Inform.* **2019**, *36*, 55–81. [CrossRef]
4. Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. *Science* **2015**, *349*, 255–260. [CrossRef]
5. Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*, 4th ed.; Pearson Education, Inc.: Upper Saddle River, NJ, USA, 2002. Available online: http://aima.cs.berkeley.edu/ (accessed on 13 July 2021).
6. Mohsin, A.H.; Zaidan, A.A.; Zaidan, B.B.; Albahri, O.S.; Albahri, A.S.; Alsalem, M.A.; Mohammed, K.I. Blockchain authentication of network applications: Taxonomy, classification, capabilities, open challenges, motivations, recommendations and future directions. *Comput. Stand. Interfaces* **2019**, *64*, 41–60. [CrossRef]
7. Mundhe, P.; Verma, S.; Venkatesan, S. A comprehensive survey on authentication and privacy-preserving schemes in VANETs. *Comput. Sci. Rev.* **2021**, *41*, 100411. [CrossRef]
8. Nguyen, D.C.; Pathirana, P.N.; Ding, M.; Seneviratne, A. Blockchain for 5G and beyond networks: A state of the art survey. *J. Netw. Comput. Appl.* **2020**, *166*, 102693. [CrossRef]
9. Valdovinos, I.A.; Pérez-Díaz, J.A.; Choo, K.-K.R.; Botero, J.F. Emerging DDoS attack detection and mitigation strategies in software-defined networks: Taxonomy, challenges and future directions. *J. Netw. Comput. Appl.* **2021**, *187*, 103093. [CrossRef]
10. Wu, J.; Liu, J.; Zhao, Y.; Zheng, Z. Analysis of cryptocurrency transactions from a network perspective: An overview. *J. Netw. Comput. Appl.* **2021**, *190*, 103139. [CrossRef]
11. Azbeg, K.; Ouchetto, O.; Andaloussi, S.J.; Fetjah, L. A Taxonomic Review of the Use of IoT and Blockchain in Healthcare Applications. *IRBM* **2021**. [CrossRef]
12. Kouicem, D.E.; Bouabdallah, A.; Lakhlef, H. Internet of things security: A top-down survey. *Comput. Netw.* **2018**, *141*, 199–221. [CrossRef]
13. Mohd Aman, A.H.; Hassan, W.H.; Sameen, S.; Attarbashi, Z.S.; Alizadeh, M.; Latiff, L.A. IoMT amid COVID-19 pandemic: Application, architecture, technology, and security. *J. Netw. Comput. Appl.* **2021**, *174*, 102886. [CrossRef]
14. Negro-Calduch, E.; Azzopardi-Muscat, N.; Krishnamurthy, R.S.; Novillo-Ortiz, D. Technological progress in electronic health record system optimization: Systematic review of systematic literature reviews. *Int. J. Med. Inform.* **2021**, *152*, 104507. [CrossRef]
15. Lezoche, M.; Panetto, H.; Kacprzyk, J.; Hernandez, J.E.; Alemany Díaz, M.M.E. Agri-food 4.0: A survey of the supply chains and technologies for the future agriculture. *Comput. Ind.* **2020**, *117*, 103187. [CrossRef]
16. Pan, Y.; Zhang, L. Roles of artificial intelligence in construction engineering and management: A critical review and future trends. *Autom. Constr.* **2021**, *122*, 103517. [CrossRef]
17. Nawari, N.O.; Ravindran, S. Blockchain and the built environment: Potentials and limitations. *J. Build. Eng.* **2019**, *25*, 100832. [CrossRef]
18. Lu, Y. The blockchain: State-of-the-art and research challenges. *J. Ind. Inf. Integr.* **2019**, *15*, 80–90. [CrossRef]
19. Hoffmann Souza, M.L.; da Costa, C.A.; de Oliveira Ramos, G.; da Rosa Righi, R. A survey on decision-making based on system reliability in the context of Industry 4.0. *J. Manuf. Syst.* **2020**, *56*, 133–156. [CrossRef]
20. Peng, L.; Feng, W.; Yan, Z.; Li, Y.; Zhou, X.; Shimizu, S. Privacy preservation in permissionless blockchain: A survey. *Digit. Commun. Netw.* **2020**, *7*, 295–307. [CrossRef]
21. Paik, H.Y.; Xu, X.; Bandara, H.M.N.D.; Lee, S.U.; Lo, S.K. Analysis of data management in blockchain-based systems: From architecture to governance. *IEEE Access.* **2019**, *7*, 186091–186107. [CrossRef]
22. Turkanović, M.; Hölbl, M.; Košič, K.; Heričko, M.; Kamišalić, A. EduCTX: A Blockchain-Based Higher Education Credit Platform. *IEEE Access* **2018**, *6*, 5112–5127. [CrossRef]
23. Aste, T.; Tasca, P.; Di Matteo, T. Blockchain technologies: The foreseeable impact on society and industry. *Computer* **2017**, *50*, 18–28. [CrossRef]
24. Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H. An overview of blockchain technology: Architecture consensus and future trends. In Proceedings of the 2017 IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, USA, 25–30 June 2017.
25. Phillip, A.; Chan, J.; Peiris, S. A new look at Cryptocurrencies. *Econ. Lett.* **2018**, *163*, 6–9. Available online: https://ieeexplore.ieee.org/document/8029379 (accessed on 13 July 2021). [CrossRef]

26. Banerjee, S.; Bouzefrane, S.; Abane, A. Identity Management with Hybrid Blockchain Approach: A Deliberate Extension with Federated-Inverse-Reinforcement Learning. In Proceedings of the 2021 IEEE 22nd International Conference on High Performance Switching and Routing (HPSR), Paris, France, 7–10 June 2021; pp. 1–6.
27. Zhu, S.; Hu, H.; Li, Y.; Li, W. Hybrid blockchain design for privacy preserving crowdsourcing platform. In Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 14–17 July 2019; pp. 26–33. [CrossRef]
28. Jaoude, J.A.; Saade, R.G. Blockchain Applications—Usage in Different Domains. *IEEE Access* **2019**, *7*, 45360–45381. [CrossRef]
29. Di Francesco Maesa, D.; Mori, P. Blockchain 3.0 applications survey. *J. Parallel Distrib. Comput.* **2020**, *138*, 99–114. [CrossRef]
30. Karafiloski, E.; Mishev, A. Blockchain solutions for big data challenges: A literature review. In Proceedings of the IEEE EUROCON 2017—17th International Conference on Smart Technologies, Ohrid, North Macedonia, 6–8 July 2017; pp. 763–768. [CrossRef]
31. Deepa, N.; Pham, Q.-V.; Nguyen, D.C.; Bhattacharya, S.; Prabadevi, B.; Gadekallu, T.R.; Maddikunta, P.K.R.; Fang, F.; Pathirana, P.N. A Survey on Blockchain for Big Data: Approaches, Opportunities, and Future Directions. *arXiv* **2020**, arXiv:2009.00858.
32. Hackeling, G. *Mastering Machine Learning with Scikit-Learn*; Packt Publishing Ltd.: Birmingham, UK, 2017.
33. Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P. From Data Mining to Knowledge Discovery in Databases. *AI Mag.* **1996**, *17*, 37–37. [CrossRef]
34. García, S.; Luengo, J.; Herrera, F. *Data Preprocessing in Data Mining*; Springer International Publishing: Cham, Switzerland, 2015; Volume 72.
35. Fister, D.; Fister, I.; Jagrič, T.; Fister, I.; Brest, J. Wrapper-Based Feature Selection Using Self-adaptive Differential Evolution. In *Swarm, Evolutionary, and Memetic Computing and Fuzzy and Neural Computing*; Springer: Cham, Switzerland, 2019; pp. 135–154.
36. Engelbrecht, A.P. *Computational Intelligence: An Introduction*; Wiley: West Sussex, UK, 2007.
37. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
38. Prasad, N.R.; Almanza-Garcia, S.; Lu, T.T. Anomaly detection. *Comput. Mater. Contin.* **2009**, *14*, 1–22. [CrossRef]
39. Alarab, I.; Prakoonwit, S.; Nacer, M.I. Comparative Analysis Using Supervised Learning Methods for Anti-Money Laundering in Bitcoin. In Proceedings of the 2020 5th International Conference on Machine Learning Technologies, Rajasthan, India, 13–15 February 2020. [CrossRef]
40. Alarab, I.; Prakoonwit, S.; Nacer, M.I. Competence of graph convolutional networks for anti-money laundering in bitcoin blockchain. In Proceedings of the 2020 5th International Conference on Machine Learning Technologies, Rajasthan, India, 13–15 February 2020. [CrossRef]
41. Eduardo A.; Sousa, J.; Oliveira, V.C.; Almeida Valadares, J.; Borges Vieira, A.; Bernardino, H.S.; Moraes Villela, S.; Dias Goncalves, G. Fighting Under-price DoS Attack in Ethereum with Machine Learning Techniques. ACM SIGMETRICS Perform. *Eval. Rev.* **2021**, *48*, 24–27. [CrossRef]
42. Camino, R.D.; State, R.; Montero, L.; Valtchev, P. Finding suspicious activities in financial transactions and distributed ledgers. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 18–21 November 2017. [CrossRef]
43. Balagolla, E.M.S.W.; Fernando, W.P.C.; Rathnayake, R.M.N.S.; Wijesekera, M.J.M.R.P.; Senarathne, A.N.; Abeywardhana, K.Y. Credit Card Fraud Prevention Using Blockchain. In Proceedings of the 2021 6th International Conference for Convergence in Technology (I2CT), Pune, India, 2–4 April 2021. [CrossRef]
44. Mirtaheri, M.; Abu-El-Haija, S.; Morstatter, F.; Ver Steeg, G.; Galstyan, A. Identifying and Analyzing Cryptocurrency Manipulations in Social Media. *IEEE Trans. Comput. Soc. Syst.* **2021**, *8*, 607–617. doi: 10.1109/TCSS.2021.3059286. [CrossRef]
45. Toyoda, K.; Ohtsuki, T.; Mathiopoulos, P.T. Identification of High Yielding Investment Programs in Bitcoin via Transactions Pattern Analysis. In Proceedings of the GLOBECOM 2017—2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–6. [CrossRef]
46. Victor, F.; Hagemann, T. Cryptocurrency pump and dump schemes: Quantification and detection. In Proceedings of the 2019 International Conference on Data Mining Workshops (ICDMW), Beijing, China, 8–11 November 2019; pp. 244–251. [CrossRef]
47. Podgorelec, B.; Turkanović, M.; Šestak, M. A brief review of database solutions used within blockchain platforms. In Proceedings of the Advances in Intelligent Systems and Computing, Salamanca, Spain, 6–8 October 2020; pp. 121–130. [CrossRef]
48. Hu, T.; Liu, X.; Chen, T.; Zhang, X.; Huang, X.; Niu, W.; Lu, J.; Zhou, K.; Liu, Y. Transaction-based classification and detection approach for Ethereum smart contract. *Inf. Process. Manag.* **2021**, *58*, 102462. [CrossRef]
49. Chen, W.; Guo, X.; Chen, Z.; Zheng, Z.; Lu, Y.; Li, Y. Honeypot contract risk warning on ethereum smart contracts. In Proceedings of the 2020 IEEE International Conference on Joint Cloud Computing (JCC 2020), Oxford, UK, 3–6 August 2020; pp. 1–8. [CrossRef]
50. Sayadi, S.; Ben Rejeb, S.; Choukair, Z. Anomaly detection model over blockchain electronic transactions. In Proceedings of the 2019 15th International Wireless Communications and Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; pp. 895–900. [CrossRef]
51. Demertzis, K.; Iliadis, L.; Tziritas, N.; Kikiras, P. Anomaly detection via blockchained deep learning smart contracts in industry 4.0. *Neural Comput. Appl.* **2020**, *32*, 17361–17378. [CrossRef]
52. Desai, H.B.; Ozdayi, M.S.; Kantarcioglu, M. BlockFLA: Accountable Federated Learning via Hybrid Blockchain Architecture. In Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy, Virtual, 26–28 April 2021; pp. 101–112. [CrossRef]

53. Wang, Y.; Gou, Y.; Guo, Y.; Wang, H.H. Construction of Audit Internal Control Intelligent System Based on Blockchain and Cloud Storage. In Proceedings of the 4th International Conference on Trends in Electronics and Informatics, Tirunelveli, India, 16–18 April 2020; pp. 292–295. [CrossRef]
54. 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. [CrossRef]
55. Maskey, S.R.; Badsha, S.; Sengupta, S.; Khalil, I. ALICIA: Applied Intelligence in blockchain based VANET: Accident Validation as a Case Study. *Inf. Process. Manag.* **2021**, *58*, 102508. [CrossRef]
56. Momeni, P.; Wang, Y.; Samavi, R. Machine Learning Model for Smart Contracts Security Analysis. In Proceedings of the 2019 17th International Conference on Privacy, Security and Trust, PST 2019, Fredericton, NB, Canada, 26–28 August 2019. [CrossRef]
57. Ashizawa, N.; Yanai, N.; Cruz, J.P.; Okamura, S. Eth2Vec: Learning Contract-Wide Code Representations for Vulnerability Detection on Ethereum Smart Contracts. In Proceedings of the ACM Asia Conference on Computer and Communications Security Virtual Event, Hong Kong, China, 7 June 2021. [CrossRef]
58. Rathore, S.; Park, J.H.; Chang, H. Deep Learning and Blockchain-empowered Security Framework for Intelligent 5G-enabled IoT. *IEEE Access* **2021**, *9*, 90075–90083. [CrossRef]
59. Munoz, J.Z.I.; Suarez-Varela, J.; Barlet-Ros, P. Detecting cryptocurrency miners with NetFlow/IPFIX network measurements. In Proceedings of the 2019 IEEE International Symposium on Measurements and Networking, Catania, Italy, 8–10 July 2019. [CrossRef]
60. Liu, J.; Zhao, Z.; Cui, X.; Wang, Z.; Liu, Q. A novel approach for detecting browser-based silent miner. In Proceedings of the 2018 IEEE 3rd International Conference on Data Science in Cyberspace, DSC 2018, Guangzhou, China, 18–21 June 2018; pp. 490–497. [CrossRef]
61. Yilmaz, I.; Kapoor, K.; Siraj, A.; Abouyoussef, M. Privacy Protection of Grid Users Data with Blockchain and Adversarial Machine Learning. In Proceedings of the 2021 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems, Virtual, 28 April 2021; ACM: New York, NY, USA, 2021.
62. Huang, D.; Chen, B.; Li, L.; Ding, Y. Anomaly Detection for Consortium Blockchains Based on Machine Learning Classification Algorithm. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Dallas, TX, USA, 11–13 December 2020; pp. 307–318. [CrossRef]
63. Song, J.; He, H.; Lv, Z.; Su, C.; Xu, G.; Wang, W. An Efficient Vulnerability Detection Model for Ethereum Smart Contracts. In Proceedings of the International Conference on Network and System Security, Sapporo, Japan, 15–18 December 2019; pp. 433–442.
64. Dashevskyi, S.; Zhauniarovich, Y.; Gadyatskaya, O.; Pilgun, A.; Ouhssain, H. Dissecting Android Cryptocurrency Miners. In Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy, Orleans, LA, USA, 16–18 March 2020; pp. 191–202. [CrossRef]
65. Agarwal, R.; Barve, S.; Shukla, S.K. Detecting malicious accounts in permissionless blockchains using temporal graph properties. *Appl. Netw. Sci.* **2021**, *6*, 1–30. [CrossRef]
66. Zarpelão, B.B.; Miani, R.S.; Rajarajan, M. Detection of bitcoin-based botnets using a one-class classifier. In Proceedings of the IFIP International Conference on Information Security Theory and Practice, Paris, France, 11–12 December 2019; pp. 174–189.
67. Graf, R.; King, R. Neural network and blockchain based technique for cyber threat intelligence and situational awareness. In Proceedings of the 2018 10th International Conference on Cyber Conflict (CyCon), Tallinn, Estonia, 30 May–1 June 2018; pp. 409–425. [CrossRef]
68. Scicchitano, F.; Liguori, A.; Guarascio, M.; Ritacco, E.; Manco, G. Deep Autoencoder Ensembles for Anomaly Detection on Blockchain. In Proceedings of the International Symposium on Methodologies for Intelligent Systems, Graz, Austria, 20–22 May 2020; pp. 448–456. [CrossRef]
69. Suleman, M.; Soomro, T.R.; Ghazal, T.M.; Alshurideh, M. Combating Against Potentially Harmful Mobile Apps. In Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2021), Settat, Morocco, 28–30 June 2021. [CrossRef]
70. Soviany, S.; Scheianu, A.; Suciu, G.; Vulpe, A.; Fratu, O.; Istrate, C. Android Malware Detection and Crypto-Mining Recognition Methodology with Machine Learning. In Proceedings of the 2018 IEEE 16th International conference on embedded and ubiquitous computing (EUC), Bucharest, Romania, 29–31 October 2018; pp. 14–21. [CrossRef]
71. Liu, X.; Jiang, F.; Zhang, R. A New Social User Anomaly Behavior Detection System Based on Blockchain and Smart Contract. In Proceedings of the 2020 IEEE International Conference on Networking, Sensing and Control (ICNSC 2020), Nanjing, China, 30 October–2 November 2020. [CrossRef]
72. Lin, Y.J.; Wu, P.W.; Hsu, C.H.; Tu, I.P.; Liao, S.W. An Evaluation of Bitcoin Address Classification based on Transaction History Summarization. In Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Seoul, Korea, 14–17 May 2019. [CrossRef]
73. Kanemura, K.; Toyoda, K.; Ohtsuki, T. Identification of Darknet Markets' Bitcoin Addresses by Voting Per-Address Classification Results. In Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Seoul, Korea, 14–17 May 2019; pp. 154–158. CrossRef]

74. Li, Z.; Hou, J.; Wang, H.; Wang, C.; Kang, C.; Fu, P. Ethereum Behavior Analysis with NetFlow Data. In Proceedings of the 2019 20th Asia-Pacific Network Operations and Management Symposium: Management in a Cyber-Physical World, APNOMS 2019, Matsue, Japan, 18–20 September 2019. [CrossRef]
75. Wang, J.; Xie, X.; Fang, Y.; Lu, Y.; Li, T.; Wang, G. Attribute Propagation Enhanced Community Detection Model for Bitcoin De-anonymizing. In Proceedings of the International Conference on Machine Learning for Cyber Security, Xi'an, China, 19–22 September 2020. [CrossRef]
76. Fan, S.; Fu, S.; Xu, H.; Cheng, X. Al-SPSD: Anti-leakage smart Ponzi schemes detection in blockchain. *Inf. Process. Manag.* **2021**, *58*, 102587. [CrossRef]
77. Brinckman, E.; Kuehlkamp, A.; Nabrzyski, J.; Taylor, I.J. Techniques and Applications for Crawling, Ingesting and Analyzing Blockchain Data. In Proceedings of the 2019 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, 16–18 October 2019; pp. 717–722. [CrossRef]
78. Patel, V.; Pan, L.; Rajasegarar, S. Graph Deep Learning Based Anomaly Detection in Ethereum Blockchain Network. In Proceedings of the International Conference on Network and System Security, Tianjin,China, 22–24 October 2020; pp. 132–148. [CrossRef]
79. Zhao, L.; Sen Gupta, S.; Khan, A.; Luo, R. Temporal Analysis of the Entire Ethereum Blockchain Network. In Proceedings of the Web Conference, Ljubljana, Slovenia 19–23 April 2021; pp. 2258–2269. [CrossRef]
80. Zola, F.; Bruse, J.L.; Eguimendia, M.; Galar, M.; Urrutia, R.O. Bitcoin and cybersecurity: Temporal dissection of blockchain data to unveil changes in entity behavioral patterns. *Appl. Sci.* **2019**, *9*, 5003. [CrossRef]
81. Shah, R.S.; Bhatia, A.; Gandhi, A.; Mathur, S. Bitcoin Data Analytics: Scalable techniques for transaction clustering and embedding generation. In Proceedings of the 2021 International Conference on COMmunication Systems and NETworkS (COMSNETS 2021), Bengaluru, India, 5–9 January 2021. [CrossRef]
82. Gouda, D.K.; Jolly, S.; Kapoor, K. Design and Validation of BlockEval, A Blockchain Simulator. In Proceedings of the 2021 International Conference on COMmunication Systems and NETworkS (COMSNETS 2021), Bangalore, India, 5–9 January 2021; pp. 281–289. [CrossRef]
83. Iyer, S.; Thakur, S.; DIxit, M.; Katkam, R.; Agrawal, A.; Kazi, F. Blockchain and Anomaly Detection based Monitoring System for Enforcing Wastewater Reuse. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT 2019), Kanpur, India, 6–8 July 2019. [CrossRef]
84. Belhadi, A.; Djenouri, Y.; Srivastava, G.; Jolfaei, A.; Chun-Wei Lin, J. Privacy Reinforcement Learning for Faults Detection in the Smart Grid. *Ad Hoc Netw.* **2021**, *119*, 102541. [CrossRef]
85. Belhadi, A.; Djenouri, Y.; Srivastava, G.; Lin, J.C.W. SS-ITS: Secure scalable intelligent transportation systems. *J. Supercomput.* **2021**, *77*, 7253–7269. [CrossRef]
86. Li, M.; Zhang, K.; Liu, J.; Gong, H.; Zhang, Z. Blockchain-based anomaly detection of electricity consumption in smart grids. *Pattern Recognit. Lett.* **2020**, *138*, 476–482. [CrossRef]
87. Keshk, M.; Turnbull, B.; Moustafa, N.; Vatsalan, D.; Choo, K.K.R. A Privacy-Preserving-Framework-Based Blockchain and Deep Learning for Protecting Smart Power Networks. *IEEE Trans. Ind. Inform.* **2020**, *16*, 5110–5118. [CrossRef]
88. Xie, X.; Fang, Y.; Jian, Z.; Lu, Y.; Li, T.; Wang, G. Blockchain-driven anomaly detection framework on edge intelligence. *CCF Trans. Netw.* **2020**, *3*, 171–192. [CrossRef]
89. Hari Pranav, Senthilmurugan, M.; Pradyumna Rahul, K.; Chinnaiyan, R. IoT and Machine Learning based Peer to Peer Platform for Crop Growth and Disease Monitoring System using Blockchain. In Proceedings of the 2021 International Conference on Computer Communication and Informatics (ICCCI 2021), Coimbatore, India, 27–29 Jauary 2021. https://doi.org/10.1109/ICCCI50826.2021.9402435 (accessed on 24 June 2021)
90. Liang, C.; Shanmugam, B.; Azam, S.; Jonkman, M.; De Boer, F.; Narayansamy, G. Intrusion Detection System for Internet of Things based on a Machine Learning approach. In Proceedings of the 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), Vellore, India, 30–31 March 2019. [CrossRef]
91. Liang, W.; Xiao, L.; Zhang, K.; Tang, M.; He, D.; Li, K.C. Data Fusion Approach for Collaborative Anomaly Intrusion Detection in Blockchain-based Systems. *IEEE Internet Things J.* **2021**. [CrossRef]
92. Jadidi, Z.; Dorri, A.; Jurdak, R.; Fidge, C. Securing manufacturing using blockchain. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications, Guangzhou, China, 29 December–1 January 2020; pp. 1920–1925. [CrossRef]
93. Cheema, M.A.; Qureshi, H.K.; Chrysostomou, C.; Lestas, M. Utilizing Blockchain for Distributed Machine Learning based Intrusion Detection in Internet of Things. In Proceedings of the 16th Annual International Conference on Distributed Computing in Sensor Systems, Los Angeles, CA, USA, 25–27 May 2020; pp. 429–435. [CrossRef]
94. Ngo, Q.D.; Nguyen, H.T.; Tran, H.A.; Nguyen, D.H. IoT Botnet detection based on the integration of static and dynamic vector features. In Proceedings of the ICCE 2020—2020 IEEE 8th International Conference on Communications and Electronics, Phu Quoc Island, Vietnam, 13–15 January 2020; pp. 540–545. [CrossRef]
95. Alkadi, O.; Moustafa, N.; Turnbull, B.; Choo, K.-K.R. A Deep Blockchain Framework-Enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks. *IEEE Internet Things J.* **2020**, *8*, 9463–9472. [CrossRef]
96. Ali, J.; Ali, T.; Alsaawy, Y.; Shahrafidz Khalid, A.; Musa, S. Blockchain-based Smart-IoT Trust Zone Measurement Architecture. In Proceedings of the International Conference on Omni-Layer Intelligent Systems, Crete, Greece, 5–7 May 2019; pp. 152–157.

97. Preuveneers, D.; Rimmer, V.; Tsingenopoulos, I.; Spooren, J.; Joosen, W.; Ilie-Zudor, E. Chained anomaly detection models for federated learning: An intrusion detection case study. *Appl. Sci.* **2018**, *8*, 2663. [CrossRef]
98. Sharma, M.; Pant, S.; Kumar Sharma, D.; Datta Gupta, K.; Vashishth, V.; Chhabra, A. Enabling security for the Industrial Internet of Things using deep learning, blockchain, and coalitions. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4137. [CrossRef]
99. Kim, T.; Jung, I.Y.; Hu, Y.C. Automatic, location-privacy preserving dashcam video sharing using blockchain and deep learning. *Human-Centric Comput. Inf. Sci.* **2020**, *10*, 1–23. [CrossRef]
100. Ferrag, M.A.; Maglaras, L. DeepCoin: A Novel Deep Learning and Blockchain-Based Energy Exchange Framework for Smart Grids. *IEEE Trans. Eng. Manag.* **2020**, *67*, 1285–1297. [CrossRef]
101. Drungilas, V.; Vaičiukynas, E.; Jurgelaitis, M.; Butkienė, R.; Čeponienė, L. Towards blockchain-based federated machine learning: Smart contract for model inference. *Appl. Sci.* **2021**, *11*, 1–21. [CrossRef]
102. Ide, T. Collaborative anomaly detection on blockchain from noisy sensor data. In Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018; pp. 120–127. [CrossRef]
103. Kumar, R.; Kumar, P.; Tripathi, R.; Gupta, P.; Reddy Gadekallu, T.; Srivastava, G. SP2F: A Secured Privacy-Preserving Framework for Smart Agricultural Unmanned Aerial Vehicles. *Comput. Netw.* **2021**, *187*, 107819. [CrossRef]
104. Chen, L.; Kuang, X.; Xu, A.; Yang, Y.; Suo, S. Anomaly Detection on Time-series Logs for Industrial Network. In Proceedings of the 2020 3rd International Conference on Smart BlockChain (SmartBlock), Zhengzhou, China, 23–25 October 2020. [CrossRef]
105. Jin, R.; Wei, B.; Luo, Y.; Ren, T.; Wu, R. Blockchain-based data collection with efficient anomaly detection for estimating battery state-of-health. *IEEE Sens. J.* **2021**, 21, 13455–13465. [CrossRef]
106. Wang, Z.; Luo, N.; Zhou, P. GuardHealth: Blockchain empowered secure data management and Graph Convolutional Network enabled anomaly detection in smart healthcare. *J. Parallel Distrib. Comput.* **2020**, *142*, 1–12. [CrossRef]
107. Awoyemi, J.O.; Adetunmbi, A.O.; Oluwadare, S.A. Credit card fraud detection using machine learning techniques: A comparative analysis. In Proceedings of the 2017 International Conference on Computing Networking and Informatics (ICCNI), Lagos, Nigeria, 29–31 October 2017; pp. 1–9.
108. Sureshbhai, P.N.; Bhattacharya, P.; Tanwar, S. KaRuNa: A blockchain-based sentiment analysis framework for fraud cryptocurrency schemes. In Proceedings of the 2020 IEEE International Conference on Communications Workshops (ICC Workshops), Dublin, Ireland, 7–11 June 2020. [CrossRef]
109. 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. [CrossRef]
110. Badawi, E.; Jourdan, G.V.; Bochmann, G.; Onut, I.V. An Automatic Detection and Analysis of the Bitcoin Generator Scam. In Proceedings of the 5th IEEE European Symposium on Security and Privacy Workshops, Genoa, Italy, 7–11 September 2020; pp. 407–416. [CrossRef]
111. 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. [CrossRef]
112. 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]
113. Wen, H.; Fang, J.; Wu, J.; Zheng, Z. Transaction-based Hidden Strategies Against General Phishing Detection Framework on Ethereum. In Proceedings of the 2021 IEEE International Symposium on Circuits and Systems (ISCAS), East Lansing, MI, USA, 9–11 August 2021. [CrossRef]
114. Zhou, H.; Sun, G.; Fu, S.; Fan, X.; Jiang, W.; Hu, S.; Li, L. A distributed approach of big data mining for financial fraud detection in a supply chain. *Comput. Mater. Contin.* **2020**, *64*, 1091–1105.[CrossRef]
115. Lou, Y.; Zhang, Y.; Chen, S. Ponzi contracts detection based on improved convolutional neural network. In Proceedings of the 2020 IEEE 13th International Conference on Services Computing, San Francisco, CA, USA, 7–11 July 2020; pp. 353–360. [CrossRef]
116. Zhou, H.; Sun, G.; Fu, S.; Wang, L.; Hu, J.; Gao, Y. Internet Financial Fraud Detection Based on a Distributed Big Data Approach with Node2vec. *IEEE Access* **2021**, *9*, 43378–43386. [CrossRef]
117. Fan, S.; Fu, S.; Xu, H.; Zhu, C. Expose Your Mask: Smart Ponzi Schemes Detection on Blockchain. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020. [CrossRef]
118. 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. [CrossRef]
119. Bartoletti, M.; Pes, B.; Serusi, S. Data mining for detecting bitcoin ponzi schemes. In Proceedings of the 2018 Crypto Valley Conference on Blockchain Technology, Zug, Switzerland, 20–22 June 2018; pp. 75–84. [CrossRef]
120. 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]
121. Boughaci, D.; Alkhawaldeh, A.A.K. Enhancing the security of financial transactions in Blockchain by using machine learning techniques: Towards a sophisticated security tool for banking and finance. In Proceedings of the 2020 1st International Conference of Smart Systems and Emerging Technologies, Riyadh, Saudi Arabia, 3–5 November 2020; pp. 110–115. [CrossRef]
122. Lee, C.; Maharjan, S.; Ko, K.; Hong, J.W.K. Toward Detecting Illegal Transactions on Bitcoin Using Machine-Learning Methods. In Proceedings of the International Conference on Blockchain and Trustworthy Systems, Dali, China, 6–7 August 2020. [CrossRef]

123. Toyoda, K.; Mathiopoulos, P.T.; Ohtsuki, T. A Novel Methodology for HYIP Operators' Bitcoin Addresses Identification. *IEEE Access* **2019**, *7*, 74835–74848. [CrossRef]
124. Xu, G.; Guo, B.; Su, C.; Zheng, X.; Liang, K.; Wong, D.S.; Wang, H. Am I eclipsed? A smart detector of eclipse attacks for Ethereum. *Comput. Secur.* **2020**, *88*, 101604. [CrossRef]
125. Baek, H.; Oh, J.; Kim, C.Y.; Lee, K. A Model for Detecting Cryptocurrency Transactions with Discernible Purpose. In Proceedings of the 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN), Zagreb, Croatia, 2–5 July 2019. [CrossRef]
126. Liu, Y.; Peng, J.; Kang, J.; Iliyasu, A.M.; Niyato, D.; El-Latif, A.A.A. A Secure Federated Learning Framework for 5G Networks. *IEEE Wirel. Commun.* **2020**, *27*, 24–31. [CrossRef]
127. Bhowmik, M.; Sai Siri Chandana, T.; Rudra, B. Comparative Study of Machine Learning Algorithms for Fraud Detection in Blockchain. In Proceedings of the 2021 5th International Conference on Computing Methodologies and Communication, Erode, India, 8–10 April 2021. [CrossRef]
128. Abdulqadder, I.H.; Zhou, S.; Aziz, I.T.; Zou, D.; Deng, X.; Abrar Akber, S.M. An Effective Lightweight Intrusion Detection System with Blockchain to Mitigate Attacks in SDN/NFV Enabled Cloud. In Proceedings of the 2021 6th International Conference for Convergence in Technology (I2CT), Pune, India, 2–4 April 2021. [CrossRef]
129. Liu, G.; Zhou, J.; Ma, X. Classification and Sharing Method of Malware Based on Threat Intelligence. In Proceedings of 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference, Chongqing, China, 12–14 June 2020; pp. 2203–2207. [CrossRef]
130. Raje, S.; Vaderia, S.; Wilson, N.; Panigrahi, R. Decentralised firewall for malware detection. In Proceedings of the International Conference on Advances in Computing, Communication and Control 2017, Udupi, India, 13–16 September 2017; pp. 1–5. [CrossRef]
131. Yazdinejad, A.; Haddadpajouh, H.; Dehghantanha, A.; Parizi, R.M.; Srivastava, G.; Chen, M.-Y. Cryptocurrency malware hunting: A deep Recurrent Neural Network approach. *Appl. Soft Comput. J.* **2020**, *96*, 106630. [CrossRef]
132. Yang, J.; Li, T.; Liang, G.; Wang, Y.; Gao, T.; Zhu, F. Spam transaction attack detection model based on GRU and WGAN-div. *Comput. Commun.* **2020**, *161*, 172–182. [CrossRef]
133. Gao, F.; Jiang, F.; Zhang, Y.; Doss, R. Quorum chain-based malware detection in android smart devices. In Proceedings of the International Conference on Future Network Systems and Security, Melbourne, Australia, 27–29 November 2019; pp. 212–224. [CrossRef]
134. Chen, Q.; Zhang, S.; Wei, W. Decentralized Autonomous Video Copyright Protection. In Proceedings of the Future Technologies Conference, San Francisco, CA, USA, 24–25 October 2019. [CrossRef]
135. Kumar, R.; Zhang, X.; Wang, W.; Khan, R.U.; Kumar, J.; Sharif, A. A Multimodal Malware Detection Technique for Android IoT Devices Using Various Features. *IEEE Access* **2019**, *7*, 64411–64430. [CrossRef]
136. Mittal, R.; Bhatia, M.P.S. Detection of Suspicious or UnTrusted Users in Crypto-Currency Financial Trading Applications. *Int. J. Digit. Crime Forensics.* **2021**, *13*, 79–93. [CrossRef]
137. Darabian, H.; Homayounoot, S.; Dehghantanha, A.; Hashemi, S.; Karimipour, H.; Parizi, R.M.; Choo, K.K.R. Detecting Cryptomining Malware: A Deep Learning Approach for Static and Dynamic Analysis. *J. Grid Comput.* **2020**, *18*, 293–303. [CrossRef]
138. Caprolu, M.; Raponi, S.; Oligeri, G.; Di Pietro, R. Cryptomining makes noise: Detecting cryptojacking via Machine Learning. *Comput. Commun.* **2021**, *171*, 126–139. [CrossRef]
139. Gangwal, A.; Piazzetta, S.G.; Lain, G.; Conti, M. Detecting covert cryptomining using HPC. In Proceedings of the International Conference on Cryptology and Network Security, Vienna, Austria, 14–16 December 2020; pp. 344–364.
140. Pastor, A.; Mozo, A.; Vakaruk, S.; Canavese, D.; López, D.R.; Regano, L.; Gómez-Canaval, S.; Lioy, A. Detection of encrypted cryptomining malware connections with machine and deep learning. *IEEE Access* **2020**, *8*, 158036–158055. [CrossRef]
141. Neto, H.N.C.; Lopez, M.A.; Fernandes, N.C.; Mattos, D.M.F. MineCap: Super incremental learning for detecting and blocking cryptocurrency mining on software-defined networking. *Ann. Telecommun.* **2020**, *75*, 121–131. [CrossRef]
142. Ostapowicz, M.; Żbikowski, K. Detecting Fraudulent Accounts on Blockchain: A Supervised Approach. In Proceedings of the International Conference on Web Information Systems Engineering, Amsterdam, The Netherlands, 20–24 October 2019; pp. 18–31. [CrossRef]
143. Nukala, V.S.K.A. Website Cryptojacking Detection Using Machine Learning : IEEE CNS 20 Poster. In Proceedings of the 2020 IEEE Conference on Communications and Network Security, Avignon, France, 29 June–1 July 2020. [CrossRef]
144. Kharraz, A.; Lever, C.; Borisov, N.; Ma, Z.; Mason, J.; Antonakakis, M.; Murley, P.; Miller, A.; Bailey, M. Outguard: Detecting in-browser covert cryptocurrency mining in the wild. In Proceedings of the The World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019. [CrossRef]
145. Cheema, M.A.; Ashraf, N.; Aftab, A.; Qureshi, H.K.; Kazim, M.; Azar, A.T. Machine Learning with Blockchain for Secure E-voting System. In Proceedings of the 2020 1st International Conference of Smart Systems and Emerging Technologies, Riyadh, Saudi Arabia, 3–5 November 2020. pp. 177–182. [CrossRef]
146. Gangwal, A.; Conti, M. Cryptomining Cannot Change Its Spots: Detecting Covert Cryptomining Using Magnetic Side-Channel. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 1630–1639. [CrossRef]

147. Mansor, W.N.A.B.W.; Ahmad, A.; Zainudin, W.S.; Saudi, M.M.; Kama, M.N. Crytojacking Classification based on Machine Learning Algorithm. In Proceedings of the 2020 8th International Conference on Communications and Broadband Networking, Auckland, New Zealand, 15–18 April 2020; pp. 73–76. [CrossRef]
148. Sun Yin, H.; Vatrapu, R. A first estimation of the proportion of cybercriminal entities in the bitcoin ecosystem using supervised machine learning. In Proceedings of the 2017 IEEE International Conference on Big Data, Boston, MA, USA, 11–14 December 2017; pp. 3690–3699. [CrossRef]
149. Nerurkar, P.; Bhirud, S.; Patel, D.; Ludinard, R.; Busnel, Y.; Kumari, S. Supervised learning model for identifying illegal activities in Bitcoin. *Appl. Intell.* **2021**, *51*, 3824–3843. [CrossRef]
150. Veselý, V.; Adník, M.Z. How to detect cryptocurrency miners? By traffic forensics! *Digit. Investig.* **2019**, *31*, 100884. [CrossRef]
151. Deebak, B.D.; Al-Turjman, F. Privacy-preserving in smart contracts using blockchain and artificial intelligence for cyber risk measurements. *J. Inf. Secur. Appl.* **2021**, *58*, 102749. [CrossRef]
152. Farrugia, S.; Ellul, J.; Azzopardi, G. Detection of illicit accounts over the Ethereum blockchain. *Expert Syst. Appl.* **2020**, *150*, 113318. [CrossRef]
153. Nerurkar, P.; Busnel, Y.; Ludinard, R.; Shah, K.; Bhirud, S.; Patel, D. Detecting Illicit Entities in Bitcoin using Supervised Learning of Ensemble Decision Trees. In Proceedings of the 2020 10th International Conference on Information Communication and Management, Paris, France, 12–14 August 2020; pp. 25–30. [CrossRef]
154. Sun, H.; Ruan, N.; Liu, H. Ethereum Analysis via Node Clustering. In Proceedings of the International Conference on Network and System Security, Sapporo, Japan, 15–18 December 2019; pp. 114–129. [CrossRef]
155. Jourdan, M.; Blandin, S.; Wynter, L.; Deshpande, P. Characterizing entities in the bitcoin blockchain. In Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Beijing, China, 8–11 November 2019; pp. 55–62. [CrossRef]
156. Jan, S.; Musa, S.; Ali, T.; Nauman, M.; Anwar, S.; Ali Tanveer, T.; Shah, B. Integrity verification and behavioral classification of a large dataset applications pertaining smart OS via blockchain and generative models. *Expert Syst.* **2021**, *38*, e12611. [CrossRef]
157. Linoy, S.; Stakhanova, N.; Matyukhina, A. Exploring Ethereum's Blockchain Anonymity Using Smart Contract Code Attribution. In Proceedings of the 15th International Conference on Network and Service Management, Halifax, NS, Canada, 21–25 October 2019. [CrossRef]
158. Hamdi, T.; Slimi, H.; Bounhas, I.; Slimani, Y. A Hybrid Approach for Fake News Detection in Twitter Based on User Features and Graph Embedding. In Proceedings of the International Conference on Distributed Computing and Internet Technology, Bhubaneswar, India, 9–12 January 2020; pp. 266–280. [CrossRef]
159. Kaci, A.; Rachedi, A. Toward a Machine Learning and Software Defined Network Approaches to Manage Miners' Reputation in Blockchain. *J. Netw. Syst. Manag.* **2020**, *28*, 478–501. [CrossRef]
160. Ding, Z.; Liu, S.; Li, M.; Lian, Z.; Xu, H. A Blockchain-Enabled Multiple Object Tracking for Unmanned System with Deep Hash Appearance Feature. *IEEE Access* **2021**, *9*, 1116–1123. [CrossRef]
161. Zhang, C.; Li, W.; Luo, Y.; Hu, Y. AIT: An AI-Enabled Trust Management System for Vehicular Networks Using Blockchain Technology. *IEEE Internet Things J.* **2021**, *8*, 3157–3169. [CrossRef]
162. Zhang, Q.; Zhu, J.; Wang, Y. Trustworthy Dynamic Target Detection and Automatic Monitor Scheme for Mortgage Loan with Blockchain-Based Smart Contract. In Proceedings of the Communications in Computer and Information Science, Lima, Peru, 1–3 October 2020; pp. 415–427. [CrossRef]
163. Liu, H.; Zhang, S.; Zhang, P.; Zhou, X.; Shao, X.; Pu, G.; Zhang, Y. Blockchain and Federated Learning for Collaborative Intrusion Detection in Vehicular Edge Computing. *IEEE Trans. Veh. Technol.* **2021**, *70*, 6073–6084. [CrossRef]
164. Hao, X.; Ren, W.; Xiong, R.; Zhu, T.; Raymond Choo, K.-K. Asymmetric cryptographic functions based on generative adversarial neural networks for Internet of Things. *Futur. Gener. Comput. Syst.* **2021**, *124*, 243–253. [CrossRef]
165. Salimitari, M.; Joneidi, M.; Chatterjee, M. AI-enabled blockchain: An outlier-aware consensus protocol for blockchain-based iot networks. In Proceedings of the 2019 IEEE Global Communications Conference, GLOBECOM, Big Island, HI, USA, 9–13 December 2019. [CrossRef]
166. Maskey, S.R.; Badsha, S.; Sengupta, S.; Khalil, I. BITS: Blockchain based Intelligent Transportation System with Outlier Detection for Smart City. In Proceedings of the 2020 IEEE International Conference on Pervasive Computing and Communications Workshops, Austin, TX, USA, 23–27 March 2020. [CrossRef]
167. Haddaji, A.; Ayed, S.; Fourati, L.C. Blockchain-based Multi-Levels Trust Mechanism against Sybil Attacks for Vehicular Networks. In Proceedings of the 2020 IEEE 14th International Conference on Big Data Science and Engineering, Guangzhou, China, 31 December–1 January 2020; pp. 155–163. [CrossRef]
168. Dhieb, N.; Ghazzai, H.; Besbes, H.; Massoud, Y. A Secure AI-Driven Architecture for Automated Insurance Systems: Fraud Detection and Risk Measurement. *IEEE Access* **2020**, *8*, 58546–58558. [CrossRef]
169. Wu, H.; Cao, J.; Yang, Y.; Tung, C.L.; Jiang, S.; Tang, B.; Liu, Y.; Wang, X.; Deng, Y. Data management in supply chain using blockchain: challenges and a case study. In Proceedings of the 2019 28th International Conference on Computer Communication and Networks (ICCCN), Valencia, Spain, 29 July–1 August 2019. [CrossRef]
170. Le Khac, N.A.; Kechadi, M.T. Application of data mining for anti-money laundering detection: A case study. In Proceedings of the 2010 IEEE International Conference on Data Mining Workshops, Sydney, Australia, 13 December 2010; pp. 577–584.
171. Byman, D. Understanding, and misunderstanding, state sponsorship of terrorism. *Stud. Confl. Terror.* **2020**, 1–19. [CrossRef]
172. Khatoon, A. A blockchain-based smart contract system for healthcare management. *Electronics* **2020**, *9*, 94. [CrossRef]

173. Pham, H.L.; Tran, T.H.; Nakashima, Y. A secure remote healthcare system for hospital using blockchain smart contract. In Proceedings of the 2018 IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6.
174. Mackey, T.K.; Miyachi, K.; Fung, D.; Qian, S.; Short, J. Combating health care fraud and abuse: Conceptualization and prototyping study of a blockchain antifraud framework. *J. Med. Internet Res.* **2020**, *22*, e18623. [CrossRef]