algorithms

MDPI

Review

Anomaly Detection in Blockchain Networks Using Unsupervised Learning: A Survey

Christos Cholevas, Eftychia Angeli, Zacharoula Sereti, Emmanouil Mavrikos and George E. Tsekouras *

Department of Cultural Technology and Communication, University of the Aegean, 81100 Mytilene, Greece; cti22005@ct.aegean.gr (C.C.); cti22006@ct.aegean.gr (E.A.); zsereti@sklavenitis.co.gr (Z.S.); emmmavrikos@aegean.gr (E.M.)

* Correspondence: gtsek@aegean.gr; Tel.: +30-22-51036631

Abstract: In decentralized systems, the quest for heightened security and integrity within blockchain networks becomes an issue. This survey investigates anomaly detection techniques in blockchain ecosystems through the lens of unsupervised learning, delving into the intricacies and going through the complex tapestry of abnormal behaviors by examining avant-garde algorithms to discern deviations from normal patterns. By seamlessly blending technological acumen with a discerning gaze, this survey offers a perspective on the symbiotic relationship between unsupervised learning and anomaly detection by reviewing this problem with a categorization of algorithms that are applied to a variety of problems in this field. We propose that the use of unsupervised algorithms in blockchain anomaly detection should be viewed not only as an implementation procedure but also as an integration procedure, where the merits of these algorithms can effectively be combined in ways determined by the problem at hand. In that sense, the main contribution of this paper is a thorough study of the interplay between various unsupervised learning algorithms and how this can be used in facing malicious activities and behaviors within public and private blockchain networks. The result is the definition of three categories, the characteristics of which are recognized in terms of the way the respective integration takes place. When implementing unsupervised learning, the structure of the data plays a pivotal role. Therefore, this paper also provides an in-depth presentation of the data structures commonly used in unsupervised learning-based blockchain anomaly detection. The above analysis is encircled by a presentation of the typical anomalies that have occurred so far along with a description of the general machine learning frameworks developed to deal with them. Finally, the paper spotlights challenges and directions that can serve as a comprehensive compendium for future research efforts.

Keywords: blockchain; anomaly detection; unsupervised learning; public blockchain networks; private blockchain networks

check for updates

Citation: Cholevas, C.; Angeli, E.; Sereti, Z.; Mavrikos, E.; Tsekouras, G.E. Anomaly Detection in Blockchain Networks Using Unsupervised Learning: A Survey. *Algorithms* **2024**, *17*, 201. https://doi.org/10.3390/a17050201

Academic Editors: Alessio Martino and Indro Spinelli

Received: 7 April 2024
Revised: 26 April 2024
Accepted: 7 May 2024
Published: 9 May 2024

1. Introduction

A blockchain network constitutes a decentralized software application dedicated to the perpetual maintenance of an expanding ledger of blocks of transaction information that enables the development of public or private distributed networks without the presence of a central controlling organization, supporting secure transaction strategies based on cryptographic mechanisms [1–4]. Each block includes, among others, data related to several transactions that were gathered within a certain period, a timestamp, and a unique hash value [1,4,5]. A cryptographic hash value prudently identifies each block, which also dutifully references the hash of the block that precedes it. This interplay culminates in the establishment of an unbroken chain, with each block forging a connection. The participants in the network are viewed as nodes and the network itself establishes communication channels between these nodes to perform the above-mentioned transactions.

In the inaugural phase of blockchain, the concept of decentralization took root, albeit primarily confined to financial transactions. However, to usher in further advancements, the second era of blockchain has been based on smart contracts and finds the researchers diligently working on integrating the innovative feature of programmability into the network, all thanks to the development of smart contracts [6,7]. The decentralized features of a blockchain network manifest its transparent nature, resulting in secure and effective data storage and data analysis applications, which span a wide range of research areas such as finance [5,8,9], health [10–12], IoT [13,14], industry [15], enterprise [16,17], etc.

The transaction verification process is called the consensus mainly due to the requirement that the users' majority must agree upon its validity [8,18,19]. In terms of consensus, three general types of blockchain systems have arisen, namely, public, consortium, and private networks [1,4,11,18,20–22]. A public blockchain allows everyone to join the network as a user either to perform transactions or to participate in the consensus process, whereas in a consortium network, the consensus mechanism is decided by a preselected set of users. Finally, a private blockchain system is usually governed by the organization/institution that developed it, and it decides the users that join the network as well as controls the consensus process. Typical consensus mechanisms are the proof of work (PoW), which is related to Bitcoin, and the proof of stake (PoS), which is related to Ethereum. PoW allows the miners to create new transactions by providing strategies to solve highly complex mathematical problems [22,23]. PoS is based on validating a transaction in terms of randomly choosing a maximum coin owner [24].

Cryptocurrencies are based on using blockchain technologies, each one from a different point of view but with the same goal, i.e., to make transactions safer for users. In that sense, blockchain harbors a multitude of attributes encompassing fault tolerance, resistance to tampering, and the cloak of anonymity [25,26]. However, although blockchain has been acknowledged as the spearhead in developing secure decentralized applications, it has shown certain failures regarding security flaws and transparency in cryptocurrencies and smart contracts [1,27,28]. So far, several blockchain security issues have been identified, which are related to various types of attacks. Attacks over a blockchain system are triggered by financial profits, and/or they target it to negatively affect its popularity [26].

Attack attempts are generally projected onto the recorded blockchain datasets as anomalies in the form of uncommon or unpredicted items or behavioral patterns. Various types of anomalies concern wallet attacks, Ponzi schemes, PoW vulnerabilities, cryptojacking, phishing scams, spam transactions, malicious accounts, etc. [26,27,29,30]. Anomaly detection focuses on the implementation of specialized algorithmic-based methodologies able to determine and quantify the above-mentioned anomalies that may have the form of suspicious transactions or user behaviors [31,32]. For example, identifying a suspicious/illegal transaction and preventing it from approval would eliminate the potential damage that could be expanded within the network [33]. In that sense, an illegal transaction is viewed as anomalous behavior, which appears to be very dissimilar to the rest of the transaction data.

Roughly speaking, the very core of an anomaly detection algorithm is to build a model that accurately describes and quantifies the normal user behaviors [34,35]. As such, the usage of an anomaly detection method provides the potential to implement timely actions against the anomaly and the respective malicious effects that might be imposed into the network.

Although anomaly detection technologies have been effectively used in handling security and privacy issues over several fields [30,31], the implementation of such types of methods in blockchain systems has appeared to be a tough problem due to several reasons [25,34]. The first reason is related to the complexity of a blockchain system, which implies the presence of many diverse threats and abnormal behaviors. The second is related to the constant increase in the number of anomaly detection methods rendering the appropriate selection of an effective method for the problem at hand a difficult task. The third reason concerns the need to optimize data formats for each blockchain network by considering its peculiarities and characteristics. Because a blockchain platform is decentralized, inappropriate handling of data formats could lead to the platform's collapse. From the viewpoint of anomaly detection methods, this imposes difficulties related to their adaptation in capturing the differences in data formats. Finally, anomaly detection tools are not efficient in applying detection rules to complex structures such as encrypted data used for protecting anonymity and sensitive user parameters or performing transactions with transaction rate requirements [36].

In that direction, unsupervised learning has gained increasing popularity in handling the problem of anomaly detection in blockchain. Such kinds of techniques provide a powerful means of segmenting the available data into distinct groups, enabling the identification of abnormal instances in the form of outliers. The incorporation of unsupervised learning algorithms in the anomaly detection process has shown effective results in enhancing the overall security of blockchain. So far, several unsupervised learning methods have been applied. Typical tools concern standard cluster analysis such as the k-means and its variants [37], and more sophisticated clustering methods such as the BIRCH [38], the Grey [39] and the Chameleon [40] algorithms, the one-class support vector machines (SVM) [41,42], the isolation-forest [43], and more.

This paper provides a systematic review of methods that utilize unsupervised learning in resolving the problem of anomaly detection in blockchain networks. The main contributions of the paper are enumerated as follows:

(a) Summarization of typical blockchain anomalies.
(b) Analysis of the data structures employed in the implementation of the unsupervised learning methodologies.
(c) Categorization of a large number of research methods for blockchain anomaly detection into three categories based on the implementation strategies of the corresponding algorithms.
(d) Presentation of the basic functional properties of the above-mentioned categories in terms of certain key characteristics.
(e) Highlight several challenges and future directions.

To conduct our analysis, we searched the following databases: Web of Science, Scopus, Google Scholar, IEEE Xplore Digital Library, ACM Digital Library, Springer, and Science Direct. In our investigation, we used the following keywords: "Blockchain", "Anomaly Detection", "Unsupervised Learning", "Cluster Analysis", "Bitcoin", and "Ethereum" as well as their combinations. The period used in our search was from 2014 to 2023. We have observed a significant increase in papers that deal with anomaly detection in blockchain from 2018 onward.

The rest of the paper is synthesized as follows. Section 2 presents the related work. Section 3 discusses some basic features of blockchain technologies. Section 4 summarizes the typical blockchain anomalies. Section 5 presents a categorization of the typical unsupervised learning algorithms employed in developing anomaly detection methods along with some notation on the use of supervised algorithms. Section 6 discusses the typical data representations used. Section 7 categorizes and analyzes unsupervised learning-based methodologies. Section 8 includes a detailed description of the identified challenges and future directions. Finally, the paper concludes in Section 9.

2. Related Work

So far, several survey papers have been published to encircle the applications of blockchain technologies in various research areas such as cloud computing [4,44], database systems [45], digital twins [46,47], educational technologies [48], e-voting development [49], interoperability [50], smart contracts [51–53], internet of things (IoT) [54–60], and more. In addition, a large number of surveys address issues related to system security, cybersecurity, and privacy of blockchain frameworks [26,61–69].

Although the above-referred works address to some extent issues related to anomaly detection and deanonymization, in this section, we turn our attention to the following studies that appear to be more related to the current endeavor.

Musa et al. [30] investigated the application-based domains of anomaly detection, providing a categorization of the relative methods and types in terms of learning modes and techniques. The application domains involved intrusion and fraud detection systems, industrial damaging, image processing, and medical studies. After identifying three general types of threats, the techniques were categorized in terms of supervised, semi-supervised, and unsupervised mechanisms that spanned over a wide range of algorithmic strategies such as neural and Bayesian networks, support vector machines, rule-based systems, nearest neighbor schemes, clustering-based approaches, and statistical tools. Chandola et al. [31] discussed a fundamental framework for each investigated anomaly detection category, encompassing information-theoretic and spectral techniques while incorporating existing grouping methods. As such, they refined a succinct categorization of the selected methodologies by meticulously outlining their pros and cons based on unique assumptions and criteria for defining the anomalies in each category, while reporting computational complexity analysis for each one of the studied methodologies. Pourhabibi et al. [32] analyzed the usage of graph-based anomaly detection methods in fraud detection by creating a hierarchical classification framework to group the methods into certain categories depending on several criteria such as types of networks and anomalies employed. Moreover, they offered a list of major difficulties faced by graph-based structures in fraud detection, underling the difficulties that exist in that domain. Hisham et al. [27] postulated that ensembles of classifiers can effectively cope with certain vulnerabilities of blockchain frameworks such as security, abuse and cyber-attacks, criminal activity, money laundering, and so on. To carry out a systematic and solid presentation, they also focused on the strengths and weaknesses of the above-mentioned models, spotlighting their importance during various stages of the data analysis such as the data preparation and preprocessing stages.

Even though blockchain has been acknowledged as the spearhead in the development of decentralized applications, it has shown certain failures regarding security issues in cryptocurrencies and smart contracts. An effective way to eliminate the impact of the above issues is the employment of data-mining models with specific metrics, criteria, and requirements that appear to have critical importance as far as the model's robustness is concerned. In this direction, some survey papers further delve into exclusively investigating the use of data mining and machine learning theory in detecting anomalies in blockchain networks. For example, from a detailed point of view, Li et al. [25] identified two major groups of approaches. The first group encompassed methods that have a general purpose without focusing on specific anomalies, and the second one included methods developed for specific types of anomalies. Based on well-defined criteria, each one of the above groups was further divided into several subgroups, where the description of their structure was also presented while analyzing the pros and cons of each subgroup. In [70], cryptocurrency security failures were studied in terms of an inductive methodology, where the very core of the analysis was to identify the properties of the data mining algorithms that ensure their feasible implementation given the above-mentioned failures. In [28], the implementation of data mining strategies for detecting anomalies in blockchain environments was considered from the perspective of certain characteristics involved in environments such as decentralization and transparency. Applications investigated included cryptocurrencies, supply chain management, finance, and healthcare. A different point of view was provided in [26], where the existing literature of data mining models used for blockchain anomaly detection was studied based on their implementation in different layers appearing in blockchain structures such as the data layer, network layer, incentive layer, and contract layer. To come up with a general methodology in model generation, the authors employed certain requirements to assess the corresponding performance and robustness. To this end, the authors undertook a discussion related to the open problems, challenges, and future research endeavors.

3. Blockchain Overview

This section presents the preliminary concepts related to blockchain technologies. As expected, the amount of relative information that exists in the literature is vast. For that reason, the analysis is kept brief, highlighting the very core of blockchain properties.

3.1. Basic Characteristics of Blockchain

Several characteristics and properties are attributed to a fully operational blockchain network. Figure 1 depicts a subset of those characteristics that are well documented in the existing literature and are briefly analyzed within the subsequent paragraphs.

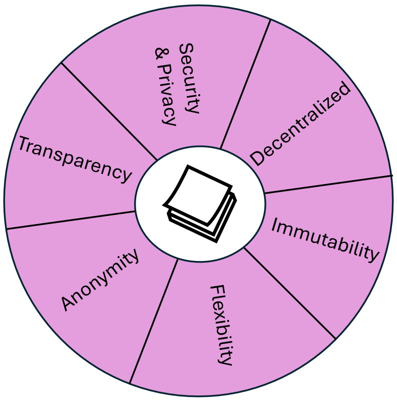

Figure 1. Basic characteristics possessed by a blockchain network.

Blockchain operates within a decentralized paradigm, enabling a network of individuals or organizations to securely record transactions [19,69]. An innovative facet introduced by blockchain is its capacity to facilitate secure agreements between multiple entities over public networks, without the need for third-party intermediaries [71]. This process, known as “mining”, assures the validity and consistency of appended agreements [59,72].

The immutability feature ensures that no tampering with the data can take place [73]. Given that each transaction is verified and recorded in distributed blocks, breaching the system becomes practically insurmountable. This intrinsic security feature guarantees the integrity of the blockchain [74], allowing for the creation of exact copies of transactions by the users, resulting in a situation where the data cannot be changed without the consent of the users [75,76].

The feature of flexibility refers to the fact that blockchain technologies are based on open-source code, meaning that any user or institution can develop applications or even new blockchain networks to fit their needs [77]. This is supported by the existing availability of several flexible blockchain platforms.

The concept of anonymity is related to the users’ engagement with the blockchain using a generated address, shrouding their identity [78]. Moreover, users can create multiple addresses to evade identity exposure [79]. Unlike centralized systems, no central entity retains users’ private information, preserving a degree of privacy in blockchain-recorded transactions [75,80].

Transparency facilitates the users to perform clear reviews of historical transactions, while no one wields the authority to alter or expunge them, preserving an immutable record of the group’s activities. Thus, the transparency feature, coupled with the blockchain’s distributed nature, ensures heightened traceability and thwarts unauthorized interventions [75,76].

The security and privacy features are related to the use of public key encryption to protect data security. Blockchain enables a user to create private and public keys. The private key is used to sign data, whereas the public key to confirm the originality and authenticity of the signed data. Keeping the private key safe from leaking is of the utmost importance. Users are anonymous, and each one of them can be assigned multiple addresses. Then, user privacy is protected by using only one address for identification purposes, while the anonymous address cannot be mapped to a user [74–77].

A task strongly related to the security feature is digital signing. A digital signature process encompasses the signing and verification, while it serves as a cryptographic proof system, affirming the blockchain's validity and cultivating trust among users [25]. Grounded in asymmetric cryptography algorithms, each user applies the private key for signing transactions accessible via public keys, permeating the network [58,80]. In essence, digital signing enhances the trust within decentralized blockchain networks.

3.2. Transactions and Smart Contracts

The exchange of assets defines the transaction. A transaction is managed under the entity service's rules, which are designed and implemented in terms of specialized script languages and forms. For example, in Bitcoin, such a language is the Bitcoin's Forth. As such, an operational set of rules allows for performing advanced transactions (e.g., escrow and multi-party signatures) [72].

Transactions in Bitcoin may feature multiple inputs and outputs, enabling complex structures and the allocation of bitcoins to various recipients in a single transaction, while supporting transaction fees and voluntary payments by the sender to incentivize miners to prioritize their transactions for block inclusion. Bitcoin transactions present a secure, transparent, and decentralized approach to peer-to-peer transactions, free from intermediary intervention [81]. Addresses in Bitcoin emerge through asymmetric cryptography, specifically using elliptic curve cryptography (ECC) [58]. However, address reuse jeopardizes privacy by exposing all associated transactions on the blockchain. Analysis of an address's transaction history enables the tracking of fund flow, which may potentially link disparate events and activities to the same address owner.

In the Ethereum network, smart contracts represent a facet of blockchain technology that reflects a synergistic blend between distributed record-keeping and executable computer code. Beyond mere documentation of past events, integration of smart contracts engenders the creation of precise code governing processes and responses to specific events [51,82]. Compared to Bitcoin network, smart contracts are also governed by similar rules. In particular, a smart contract is defined by an aggregation of script-encoded rules, which are inserted in the network to guide and control the resulting transactions, through the autonomous execution of the contract [74,82]. As a result, smart contracts act as autonomous agents with the property of being permanently tamper proof after their verification [6,72]. The very core of a smart contract is related not only to a coding-restricted process but also to the encoding of all relative terms and conditions that regulate an agreement into the transaction workflow. In addition, smart contracts leverage blockchain technology's inherent features (e.g., recording, validation, and security). Integration with digital identity support enables credible contract execution over public networks. This not only fortifies contract security and immutability but also opens avenues for automation in agreement processes between companies and their partners or customers.

3.3. Consensus Mechanisms

Consensus algorithms serve as the foundation in ensuring the decentralized and secure nature of distributed ledgers. They foster agreement among nodes in a blockchain network, validating transactions, preventing double-spending, and upholding overall ledger integrity [24].

One prominent consensus algorithm, the Proof of Work (PoW), has gained prominence through its association with Bitcoin. PoW relies on participating nodes' computational power to solve complex mathematical puzzles, with the first successful solver earning the right to validate and add a new block to the blockchain. While effective, PoW faces criticism for substantial energy consumption, prompting the exploration of eco-friendly alternatives [71].

Proof of Stake (PoS) has emerged as one such alternative, addressing environmental concerns linked to PoW. In a PoS system, validators create new blocks based on the cryptocurrency amount they hold and are willing to "stake" as collateral [83]. This energy-efficient approach contrasts sharply with PoW, showcasing the diverse range of consensus mechanisms within the blockchain space.

The Delegated Proof of Stake (DPoS) combines elements of both PoW and PoS. DPoS involves a group of delegates, chosen through community voting, to validate transactions and produce new blocks. This approach aims to enhance efficiency by reducing the number of participants involved in the consensus process, rendering it a more scalable solution [24,71].

The Proof-of-Authority (PoA) engages a small number of selected users to perform transaction validations and update the network's distributed registry [84,85]. The selected validators create and embed into the network the new transactions' blocks, which are accepted without any further verification. On many occasions, the above-mentioned acceptance can be achieved by the unanimous vote of the block generators, or by considering the users' majority. Rendering a user as a validator depends on several criteria such as high moral standards, no criminal record, wide acceptance by the network users, validator's willingness to stake her/his reputation, etc. One of the major advantages provided by PoA is the fact that it does not require a lot of computing power.

Finally, we report two environments related to continuous authentication and verification, namely, the zero-trust and zero-knowledge proof architectures.

The functional principle of zero-trust architecture (ZTA) is to maintain tight access control over every user, action, or request entering the network without any trust, even when the user is part of the network. ZTA aligns with the decentralized and distributed nature of blockchain networks, emphasizing the importance of continuous authentication and verification in ensuring the integrity of transactions and data. Additionally, it could be efficiently used to enhance protection against various types of attacks and anomalies [86].

On the other hand, zero-knowledge proof is a cryptographic method where one party (the prover) can prove to another party (the verifier) that they know a specific piece of information without revealing the actual information itself. This concept is crucial for maintaining privacy and security in decentralized systems, where participants may want to verify transactions without exposing sensitive details [87].

When a particular event is detected as an anomaly, the whole network must reach a consensus to validate that result to enable appropriate actions against it. Therefore, the complexity involved in the consensus algorithm directly impacts the anomaly detection process. This impact is significantly enhanced when some of the network's nodes act as malicious ones [26]. The above issue becomes crucial and must be considered when designing anomaly detection models [88]. In addition, a quantity that is proportional to the complexity of consensus algorithm is the required energy consumption. As such, the implementation of unsupervised learning is directly affected by the consensus mechanism. In general, the lesser the complexity involved, the easier the above implementation becomes. Decentralization remains a fundamental principle in the above consensus algorithms ensuring that no single entity or authority holds control. Decision-making power distributed among participating nodes guarantees network resilience, transparency, and resistance to tampering or unauthorized control.

4. Anomalies and Anomaly Detection in Blockchain

This section elaborates on the concept of an anomaly in a blockchain network and the general framework of anomaly detection. In addition, to provide a convenient connection with the sections that follow, we review some of the most employed unsupervised learning algorithms in blockchain anomaly detection.

4.1. Anomalies in Blockchain Networks

Despite its advantages, blockchain technologies are not completely secure, remaining susceptible to specific attacks and issues [83,89]. Attacks on the blockchain network are launched to impact the capital or popularity of the network, leading to a decrease in its market value.

A typical case is the double-spending attack that engages malevolent actors to expend the same cryptocurrency or digital asset on multiple occasions, thereby eroding the trustworthiness and dependability of transactions. The challenge of double-spending arises from delays in disseminating pending payments across the network, thereby enabling a Bitcoin client to engage in multiple transactions involving the same Bitcoin. Another perilous predicament emerges in the form of the 51% attack, where an individual or collective entity seizes control over more than half of the mining power within the network, with the ultimate purpose of manipulating transactions. On the other hand, the prevalence of Sybil attacks should not be overlooked, where assailants fabricate multiple counterfeit identities or nodes to acquire dominion over a significant portion of the network, thereby impeding consensus and influencing the validation of transactions [90]. Similarly, Eclipse attacks [91] constitute another grave hazard, where malefactors encircle a victim's node with malevolent nodes, granting them the ability to manipulate or censor the victim's transactions at will. Selfish mining poses a formidable threat to the equity of the blockchain network because a miner or coalition of miners intentionally withholds valid blocks from the network, gaining an unjust advantage over honest miners.

Vulnerabilities inherent in smart contracts within Ethereum can be exploited through tactics such as reentrancy or arithmetic overflows/underflows, leading to unintended repercussions and financial detriment [90–92]. Also, the blockchain ecosystem remains susceptible to distributed denial-of-service (DDoS) onslaughts [93], strategically designed to inundate the network with an overwhelming surge of requests or transactions, culminating in network congestion and the potential for disruptive consequences. Ransomware attacks have likewise emerged as a consequential menace within the blockchain sphere, wherein malicious actors encrypt users' data and demand a ransom, effectively holding their information hostage. Moreover, the exploitation of users' trust through Ponzi schemes that leverage blockchain technology poses a grave threat, luring unsuspecting participants with deceptive promises of exorbitant returns, ultimately resulting in financial losses [90,91].

Table 1 illustrates some typical cases of blockchain anomalies along with their brief descriptions and their occurrences.

Table 1. Brief description of standard blockchain anomalies and their occurrences.

Anomaly	Description	Occurrence
Sybil Attacks	Creation of multiple fake identities or nodes to gain control over a significant portion of the network, often disrupting consensus and influencing transaction validation.	They have been observed in public or consortium blockchains, e.g., in consortium blockchain used by a group of financial institutions [94].
Phishing Attacks	Malicious attempts to deceive individuals into revealing sensitive information, such as passwords or financial details, by impersonating trustworthy entities through emails, websites, or messages.	They are prevalent across the cryptocurrency space, e.g., according to a report by CipherTrace, phishing attacks accounted for millions of dollars in losses in 2023 alone [95].
Ponzi Schemes	Fraudulent investment operations where early investors are paid with funds from later investors, creating an illusion of profitability until the scheme collapsed, and causing financial losses for participants.	They plagued the cryptocurrency industry (e.g., the BitConnect), causing billions of dollars in damage before collapsing in 2018 (their frequency has decreased, but they still are a threat to decentralized finance).
Double-Spending Attacks	Attempts to spend the same digital asset more than once, exploiting the delay in transaction validation to deceive the network.	Although less common in established cryptocurrencies like Bitcoin, they can occur in smaller networks or lesser-known altcoins.

Table 1. *Cont.*

Anomaly	Description	Occurrence
Ransomwares	Encryption of victim's data, rendering it inaccessible until a ransom is paid (it poses a significant threat to individuals and organizations, causing data loss or financial harm).	Their occurrence has increased, with cryptocurrencies often serving as the preferred method of payment due to their pseudonymous nature (it is expected to cost the global economy billions of dollars annually by 2025 [96]).
DDoS Attacks	Attacks that overwhelm a network or website by flooding it with a massive volume of requests or traffic, causing service disruptions or rendering it inaccessible to legitimate users.	They are a constant threat to cryptocurrency exchanges and blockchain networks as they can disrupt services, causing financial losses, e.g., in 2023, several exchanges experienced DDoS attacks, leading to temporary outages.
Eclipse Attacks	Isolation of a victim's node by surrounding it with malicious nodes, controlling the victim's network connections, and potentially manipulating or censoring their transactions.	They have occurred in various blockchain networks, including Ethereum. While not as common as other attacks, they constitute a concern for network security [90].
51% Attacks	A single entity or group controls over 50% of a blockchain network's mining power, enabling it to manipulate transactions, potentially double-spend and disrupt the network's integrity.	They have been witnessed in several smaller cryptocurrencies. The most notable example is the 51% attack on Ethereum Classic in 2019, resulting in millions of dollars in double-spending [97].
Selfish Mining Attacks	Secret mining on top of withholding blocks, gaining an unfair advantage over honest miners in the race to add blocks to the blockchain.	They are rarely observed in practice due to their complexity (however, they remain a topic of academic research and discussion in the cryptocurrency community [98]).
Brute Force Attacks	Systematic combinations of all possible passwords or encryption keys until the correct one is discovered, typically through an exhaustive trial-and-error approach.	While successful, they are relatively rare due to the strength of modern encryption algorithms, but they can still occur, especially if users employ weak passwords.
Finney Attacks	Special type of double-spending attacks, where an attacker pre-mines a valid transaction but keeps it private while mining a new block to confirm the pre-mined transaction, excluding it from the network	Rarely observed due to their intricate nature, but when occurred, they underscore the importance of robust network security measures [99].
Fork After Withholding Attacks	Successful mining of a new block without broadcasting it to the network (instead, the miner continues mining on top of the withheld block privately, aiming to gain an advantage over other miners by producing a longer chain).	They have occurred in smaller blockchain networks, where miners attempt to gain a competitive advantage by secretly mining blocks (while not as common as other attacks, they highlight the vulnerabilities inherent in proof-of-work consensus mechanisms).
Deanonymization Attacks	Involve linking IP addresses with cryptocurrency wallets compromising user privacy and security	Although they have not been reported often, they have become increasingly sophisticated [100].

Over time, several attacks have been reported on the Bitcoin and Ethereum networks. For example, a worthwhile attack occurred in 2017 on the Bitcoin network, where a large number of spam transactions flooded the network, causing delays and stalling transaction verifications [1,93]. This, in turn, increased the mining process fee for Bitcoin, resulting in delayed payments of approximately USD 700 million [101]. Figure 2 visualizes the attack, where each bar corresponds to a specific attack's feature. The negative impact of spam transactions is evident, as illustrated by the leftmost bar. The subsequent bars depict the market value volatility, mining process fee increase, and delayed payments, all contributing to an overall negative effect on the blockchain system. The values atop each bar indicate the severity of these impacts, with negative values reflecting adverse consequences. The legend clarifies the color coding, associating green with positive impacts and red with negative ones. The concise labels to the left of each bar provide a clear description of the corresponding impact.

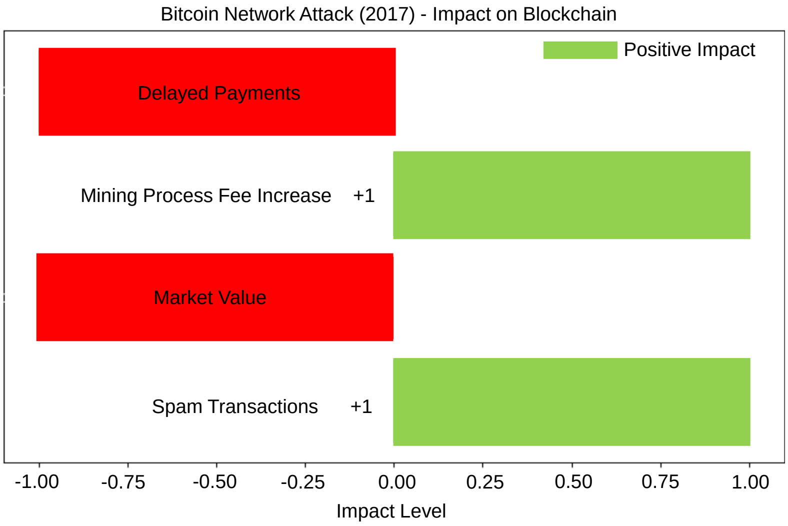

Figure 2. Impact of the 2017 attack on the Bitcoin blockchain, where delayed payments and market value are the negative impacts to the network as the spam transactions and mining process fees increase.

In 2014, an attack executed in the Ethereum blockchain called Man-in-the-Middle hijacked BGP routes within a Canadian autonomous system, resulting in the theft of USD 83,000 [102]. This incident highlighted the vulnerability of blockchain networks to attacks that aim at stealing digital assets. This attack exploited communication delays to manipulate transactions and blocks within the blockchain network, potentially leading to double-spending-based financial losses for legitimate users. The projection of blockchain anomalies on the user level is related to the concept of malicious users, i.e., users engaged in nefarious activities that trigger the above-mentioned anomaly events. Thus, it becomes crucial to identify any semblance of suspicious behavior among users, given the alarming rise in theft incidents.

4.2. Anomaly Detection in Blockchain Networks

The role of anomaly detection in blockchain security is multifaceted. Researchers are developing models for various blockchain layers, predicting anomalous commands in smart contracts, or detecting malicious block deployments. Continuous monitoring of network behavior, facilitated through both active and passive monitoring, is indispensable for timely anomaly detection [81]. Active monitoring involves the focused observation of specific network segments, while passive monitoring encompasses the comprehensive analysis of the entire network's performance. Both approaches share the common goal of identifying anomalous properties, determining whether they pose a threat, and responding accordingly. The complexity of anomaly detection aligns with the complexity of the solution, necessitating a combination of clustering, classification, and analytical tools tailored to the type of anomaly and desired outcomes [81,103]. It is important to note that the anomaly detection or prevention problem's complexity corresponds to the solution's complexity. The solution is not a singular algorithm but rather a model based on a combination of various algorithms. Analytical tools are also integrated to visualize the output, providing researchers with a clearer perspective. Similarly, when attempting to identify abrupt or systematic changes in data, models based on classification and clustering prove to be the optimal choice [26]. In contrast, if the goal is to analyze slow and long-term modifications and changes in the network, specific statistical and analytical tools become more suitable for conducting the analysis [74].

The methods employed in anomaly detection are diverse, with statistical analysis delving into transaction and block distributions, machine learning algorithms discerning from normal behavior patterns, and network analysis scrutinizing transaction flow for abnormal activity. In particular, machine learning works towards establishing baseline patterns from historical data and identifying any deviations that fall outside the expected range [104]. For example, regarding the use of cluster analysis, such kinds of deviations correspond to outliers, which refer to observations that are dissimilar to the rest of the data points within a given sample.

Table 2 depicts the basic characteristics of several anomaly detection frameworks, while the subsequent paragraphs delineate their basic operational properties.

Table 2. Anomaly detection frameworks and their characteristics.

Techniques	Strengths	Weaknesses
Statistical Analysis	Simple and interpretable approach; utilizes statistical measures to establish normal behavior patterns	Limited in detecting anomalies that deviate significantly from statistical measures
Machine Learning	Ability to learn from historical data and adapt to evolving anomalies; detection of complex and subtle anomalies	Complexity and computational overhead in training and deploying models; may generate false positives or false negatives if the anomaly patterns change over time
Network Analysis	Can capture systemic anomalies and identify network-level attacks	Limited visibility into encrypted transactions and activities; complexity in analyzing large-scale networks.
Heuristic-based	Utilizes expert knowledge and predefined indicators of suspicious activities	Limited to detecting known patterns and predefined indicators; may struggle to adapt to new and evolving types of anomalies
Deep Learning	Ability to apprehend non-linear data association; effective detection of anomalies that manifest convoluted patterns	Need for significant corpus of meticulously annotated training data, particularly when it comes to detecting anomalies, which are frequently infrequent occurrences; time-intensive training

- Statistical analysis: By harnessing the power of statistical measures, it strives to establish normal behavior patterns within a given dataset [35,105]. Through the meticulous analysis of data distributions, correlations, and probabilistic models, statistical inference methods provide insights into the expected behavior and help identify deviations that may indicate anomalous activities [104–106].
- Data mining: In the ever-evolving landscape of anomaly detection, data mining techniques emerge as powerful allies. Armed with the ability to learn from historical data, these methods adapt to changing environments and evolving anomalies [25]. Through the exploration of vast datasets, they attempt to reveal hidden patterns, correlations, and trends, enabling analysts to uncover deviations from expected behavior [28,104].
- Network analysis: When it comes to anomaly detection in interconnected systems, network analysis takes center stage. By delving into the intricate web of relationships and interactions, network analysis can capture systemic anomalies that span across multiple nodes or connections [105]. These methods leverage graph theory and network metrics to identify network-level attacks, such as coordinated efforts to disrupt communication or exploit vulnerabilities [107].
- Heuristic-based approaches: Drawing upon the wisdom of domain experts, heuristic-based approaches provide a valuable tool in the arsenal of anomaly detection [31]. These methods utilize expert knowledge and predefined indicators of suspicious activities to flag potential anomalies [23,26]. By leveraging human expertise and intuition, heuristic-based approaches can rapidly identify behaviors that deviate from established norms or violate predefined rules [79].

- Deep Learning: Deep learning models stand out for their remarkable ability to grasp intricate and non-linear associations within data [3]. These models excel at capturing complex patterns and fluctuations that may manifest convoluted relationships. By leveraging their non-linearity, deep learning models adeptly identify anomalies that may exhibit unusual patterns, previously unseen correlations, or subtle deviations from expected behavior [108–110].

5. Data Mining Techniques Employed in Blockchain Anomaly Detection

This section categorizes the unsupervised learning tools employed in the literature to develop systematic anomaly detection methodologies. The objective is to clarify their key characteristics, strengths, weaknesses, and applicability. In addition, the perspectives on using supervised and self-supervised learning for some of the anomalies reported in Table 1 are also discussed. Finally, the evaluation strategies typically used are reported and analyzed.

5.1. Categories of Unsupervised Learning Algorithms

The categorization of unsupervised learning algorithms for anomaly detection within blockchain networks underscores the diverse methodologies available to researchers and practitioners. By comprehensively understanding their attributes, strengths, and limitations, stakeholders can harness their capabilities to enhance the security, reliability, and scalability of blockchain systems.

The categorization consists of seven categories commonly employed by the anomaly detection methods presented in Section 7. Table 3 depicts the algorithms along with their categorization and computational complexity, while the categories along with their basic characteristics are described in the subsequent paragraphs.

- Partitional Methods: They mainly refer to partitional clustering algorithms such as the standard k-Means and its variants, and agglomerative hierarchical clustering [37]. Some more recent algorithms that fall into this category are the Birch [38] and affinity propagation [111].
- Graph-based methods: They are based on representing the blockchain transaction data in graph structures, where nodes represent entities (e.g., addresses, accounts), and edges represent transactions between these entities. Each node may have associated attributes such as transaction volume, frequency, etc. Then, they calculate similarities between nodes based on their behaviors and identify groups of similar nodes. Typical approaches falling in this category are the deepwalk [112], spectral clustering [113], and Louvain method [114].
- Density-based approaches: They attempt to quantify the density measure of data points in the feature space. Regarding blockchain anomaly detection, this is translated into determining the density of addresses, transactions, transaction volumes, frequency, and relations between users and addresses. Points with low densities are likely to be labeled as malicious and anomalous. Representative algorithms commonly used are the local outlier factor (LOF) [115], the DBSCAN [116], and HDBSCAN [117] algorithms.
- Probabilistic unsupervised learning algorithms: They are based on evaluating the underlying probability distributions of the data. They involve inherent modeling of latent variables, and they have been proven to be very effective in discovering hidden patterns in the data. Algorithms that are based on probabilistic modeling are the expectation maximization algorithm [37], the variational autoenconder [118], and the generative adversarial networks (GANs) [119].

Table 3. Common unsupervised learning algorithms employed in blockchain anomaly detection along with the respective computational complexity and categorization (where n, k, m, and d stand for the number of data instances, clusters, nodes, and dimensions, respectively).

Algorithm	Computational Complexity	Category	Algorithm	Computational Complexity	Category
k-Means [37]	$O(nk)$	Partitional	Agglomerative hierarchical clustering [37]	$O(kn^2)$	Partitional
Isolation forest [120]	$O(n \log n)$	Tree-based	Local outlier factor [115]	$O(n^2)$	Density-based
DBSCAN [116]	$O(n \log n)$	Density-based	HDBSCAN [117]	$O(n^2)$	Density-based
Spectral clustering [113]	$O(n^3)$	Graph-based	Louvain algorithm [114]	$O(n \log n)$	Graph-based
t-SNE [121]	$O(n^2)$	Dimensionality reduction	Birch [38]	$O(n \log n)$	Partitional
Deepwalk [112]	$O(m \log m)$	Graph-based	Expectation maximization [37]	$O(nd)$	Probabilistic
Affinity propagation [111]	$O(n^2)$	Partitional	Variational autoencoder [118]	$O(nmd)$	Probabilistic
GANs [119]	$O(nmd)$	Probabilistic	One-class SVM [41]	$O(n^2d)$	One-class classification

- One-Class Classification: It performs anomaly detection by creating boundaries around normal data points in a high-dimensional space, which contains them in a defined region. Any data points that fall outside this boundary are identified as anomalies. The main representative of this category is the one-class support vectors machine (SVM) [41].
- Tree-Based methods: They represent the blockchain data in decision tree structures and perform a labeling process according to which nodes that are isolated from the majority of nodes are defined as malicious. The most used algorithm credited to this category is the isolation forest [120].
- Dimensionality reduction methods: They focus on transforming the available high-dimensional data points into low-dimensional points, preserving the relative distances between them. Low-dimensional representation provides several advantages such as convenient visualization and easy outlier detection. In general, they are applied as assistive tools to the above categories. Such kinds of algorithms are the well-known principal component analysis (PCA) (which is linear transformation) and the t-SNE (which is non-linear transformation) [121].

5.2. Perspectives on Supervised and Self-Supervised Approaches for Anomaly Detection

In the context of blockchain, supervised approaches can be applied to detect specific anomaly patterns such as phishing attacks, double-spending, and Ponzi schemes. These methods typically involve training machine learning models on labeled datasets containing examples of normal behavior as well as known instances of anomalies [122].

For example, in phishing attack detection, a supervised approach might involve training a classification model using features extracted from email headers, website URLs, or message content. The model aims to distinguish between legitimate communications and phishing attempts based on labeled training data. Similarly, in double-spending detection, supervised learning algorithms can be trained to recognize patterns indicative of fraudulent transactions. By providing labeled examples of confirmed double-spending incidents, these models can learn to identify similar patterns in real-time transaction data [122].

On the other hand, self-supervised learning techniques leverage the inherent structure of the data to learn representations without explicit labeling. They are particularly well suited for anomaly detection tasks where labeled data may be scarce or expensive to obtain. For instance, they can be used to reconstruct transaction sequences and identify deviations from expected patterns. For anomaly patterns like Sybil attack or 51% attack, self-supervised approaches enable the thorough analysis of the network's topology and transactions' history to identify unusual node behavioral patterns or mining activities [123].

Previous research has explored various supervised and self-supervised approaches for anomaly detection in blockchain networks. Musa et al. [30] categorized them based on learning modes and techniques, including supervised, semi-supervised, and unsupervised mechanisms. Chandola et al. [31] provided several categories and discussed their pros and cons. Pourhabibi et al. [32] analyzed the usage of unsupervised learning methods in fraud detection, while Hisham et al. [27] emphasized on the effectiveness of ensemble classifier models for addressing vulnerabilities in blockchain frameworks.

To this end, both supervised and self-supervised approaches offer valuable insights for detecting blockchain anomalies with each approach having its advantages and suitability. In addition, it will be shown later in this paper that they can be effectively combined with unsupervised learning, since the latter can provide labeling assignments to unlabeled data, enabling the usage of the former.

5.3. Evaluation Approaches

The evaluation of the results obtained by unsupervised learning or by combining unsupervised and supervised learning is a very crucial step towards developing effective and robust blockchain anomaly detection methods. Within the subsequent paragraphs, we report the most used measures identified by the current investigation:

- Within cluster mean value of the sum of squares: It is defined as the average of the square distances between points belonging to a cluster and the respective cluster center. It reveals the compactness degree of the resulting clusters. Thus, it is a measure of the distortion of a cluster. Small values correspond to highly compact clusters.
- Silhouette score: It measures the similarity of a data point belonging to a specific cluster in relation to the rest of the clusters. It employs the criteria of compactness and separation. The compactness is based on estimating the average distance of the point to all other points belonging to the same cluster. On the other hand, the separation is defined as the smallest distance between the point and all points belonging to the rest of the clusters.
- Confusion matrix-based measures: They are the well-known measures coming from the resulting confusion matrices such as true positive rates (TPRs), false positive rates (FPRs), true negative rates (TNRs), precision, recall, and Fowlkes–Mallows index. They can be used when unsupervised learning is combined with supervised or self-supervised learning or there exists a portion of labeled data in the available dataset.
- Rand Index: The Rand index is a measure of similarity between two data clustering partitions of the same dataset. It considers the TPRs and TNRs and compares the agreement between the clustering results and the true class labels, making it suitable for evaluating clustering in the presence of ground truth labels. This measure can also be applied when unsupervised learning is combined with supervised or self-supervised learning or there exists a portion of labeled data in the available dataset.
- Outlier Detection Rate: It is defined as the number of detected anomalies divided by the number of total anomalies that exist in a dataset. In general, high values of this measure imply better performance of the algorithm.
- Optimal clustering: Usually, the clustering algorithms admit a predefined value for the number of clusters. Optimal clustering refers to the process of determining the optimal number of clusters in terms of compactness and separation criteria. This can be performed by iteratively applying the clustering algorithm, where in each iteration, the number of clusters increases by one. For each iteration (i.e., for each number of clusters), a function that includes the compactness and separation criteria is evaluated.

When the iteration stops, the optimal number of clusters corresponds to the minimum value of the above-mentioned function.

6. Data Structures Used in Blockchain Anomaly Detection

Depending on the problem at hand, the implementation of unsupervised learning methods in blockchain anomaly detection is strongly related to the way the data are formatted and structured and the features that are used. These topics are discussed in the subsequent analysis.

Herein, three basic types of data structures are analyzed, namely, tabular-based, sequence-based, and graph-based structures. Each of them appears to have certain characteristics while providing various convenient ways to apply unsupervised learning in detecting blockchain anomalies.

6.1. Tabular-Based Data Structures

A common representation of the blockchain data is the tabular format. The tabular format is very convenient, in particular when cluster analysis is applied, because the interrelation between features and between instances can be easily explored. In the case of Bitcoin, due to the pseudo-anonymity status of the network, the raw data extracted from the ledger are not in the position to unravel the relationship between entities (i.e., users) on the network and addresses belonging to those entities [110,124]. Thus, data transformation to tabular format can be an easy way to capture the corresponding relationships, which on many occasions appear to be hidden. In addition, the data can be analyzed and processed at various aggregation stages, and thus, different frameworks can be used to represent and store them before they are to be used in anomaly detection tasks [125]. For example, Kinkeldey et al. [124] used the Mongo Database [126] where the data were stored in a column-based structure providing fast access and retrieval as well as effective data aggregation strategies. Finally, a tabular data format leads to an easy feature extraction process [127].

6.2. Sequence-Based Data Structures

In a blockchain network, each user node broadcasts the whole block, and therefore, the complexity involved prevents the effective implementation of the process. On the other hand, once a malicious transaction is erroneously admitted by a user it cannot be undone. Thus, an in-time detection of the anomalous transaction would prevent the depletion of valuable resources for the user [128]. In this regard, the identification of anomalous transactions would be effectively supported by representing the blockchain data as time series (i.e., data sequences) and using specialized techniques (e.g., rolling window aggregation method [129]) to extract useful features that describe the properties of the time series data and define the space where the unsupervised learning is to be applied [130].

For example, times series representation of the Bitcoin data has been involved in address clustering to identify accounts belonging to the same user. A feasible way to do this relies on encoding sequences of transaction data in patterns each of which is described by features related to the information that flows in the transactions (tx_in) and the transactions' output information (tx_out) [131]. For example, considering cybersecurity applications related to blockchain, the tracing and audit of transaction events appear to be most evident, where blockchain transactions can be treated as a sequence of events, the end of which might correspond to a malicious incident [132].

The utilization of time series representation of blockchain data has been very convenient in defining behavior patterns to identify malicious activities. This is translated in studying the data under the framework of behavior patterns, originating from the very nature of the blockchain structure, because each block contains a ledger (i.e., record) of all transactions that took place since the previous block. Therefore, there exist multiple backup ledgers in a blockchain network. Pattern sequences are generated based on features selected in terms of various ways. For example, the changes in the transaction amount over time [133] or the selection of the time taken to perform a transaction together with the transaction amount from one user node to another one [134]. Notably, in both cases, the transaction amount is the most predominant feature.

The time series representation of the Ethereum can be accomplished by representing the smart contract codes as a sequence of opcodes, which are hashed and used as input to the unsupervised learning [135], such as affinity propagation [111] and k-Medoids [136].

6.3. Graph-Based Data Structures

Graph-based analysis and modeling has manifested itself as an effective tool in dealing with financial fraud [137]. The impetus behind this fact is the inherent capability of graph-based data structures in representing the interactions between the entities encoded in the graph yielding effective classification of these entities based on predefined features [138,139].

So far, the graph-based representation of blockchain data has been proven to be very effective in dealing with the limitations involved in the implementation of more traditional considerations because it supports a decentralized, scalable, and flexible way of indexing blockchain data [140]. In addition, a graph is compatible with the blockchain network structure, a fact that renders it a scalable procedure for representing the data. In a nutshell, the main benefits of using graphs in blockchain technologies are enumerated as follows [78,139,140]: (a) it is a scalable structure and can effectively encompass large amounts of future data, (b) it provides flexibility in terms of certain query languages, rendering querying for data retrieval a straightforward process, (c) the indexing process can be easily applied for a large amount of data, and (d) it is based on decentralized protocols favoring high-security levels.

Figure 3 depicts two typical graph structures, namely, user graph and transaction graph, proposed by Reid and Harrigan in [78]. The user graph represents accounts as nodes with loops indicating transactions between accounts, allowing for the embedding of fund sources and destinations. On the other hand, in the transaction graph, the node represents a transaction while the edges represent the flow of funds between transactions, where the output of a transaction is taken as the input by the next one, and the weight of the edge corresponds to the transferred amount.

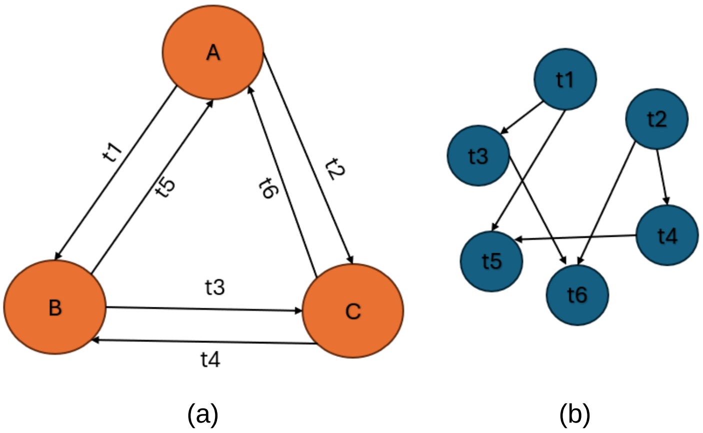

Figure 3. (**a**) User graph that depicts nodes representing accounts (i.e., A, B, C) with edges indicating transactions between accounts forming a graph with loops to capture fund flow directions and (**b**) transaction graph that shows nodes as individual transactions (i.e., t1, t2, t3, t4, t5, t6) with edges representing the flow of funds between transactions, lacking closed loops due to the sequential nature of transactions in blockchain data.

The flexibility and scalability of graph-based models make them well suited for handling the complex and interconnected nature of blockchain networks. By incorporating measures of graph regularity based on information theory, researchers can quantify the predictability of blockchain structures and enhance anomaly detection performance. These measures can help in identifying deviations from normal network behavior and distinguishing legitimate transactions from potentially malicious ones.

To elaborate blockchain graph-based data for anomaly detection tasks, specialized unsupervised learning techniques are used such as the deepwalk algorithm [112], spectral clustering [113], Louvain graph-based method [114], isolation forest [120], etc. The main target of these methods is to maintain the graph's properties (e.g., the graph's structure, the features related to nodes, the features related to edges, etc.) by incorporating graph embedding mechanisms [139]. Graph-based anomaly detection offers several advantages, including intuitive pattern recognition and flexibility in analysis. However, challenges such as scalability and privacy concerns persist. Despite these challenges, the amalgamation of various graph structures presents a visually interpretable framework for securing blockchain networks. Achieving a balance between insightful analysis and adaptability is imperative for effectively safeguarding blockchain ecosystems [110].

7. Unsupervised Learning-Based Blockchain Anomaly Detection

This section emphasizes technological achievements from the perspective of the algorithms used and their implementation schemes. From this point of view, three main categories have been identified based on the use of algorithms and their efficiency. In this direction, many research papers are reported, each of which follows one of these categories. In the analysis, prominent learning algorithms are dissected such as k-Means [37], hierarchical clustering [37], DBSCAN [116], gaussian mixture model (GMM), agglomerative clustering [37], self-organizing maps (SOM), etc.

The first category delineates methodologies that solitarily implement unsupervised learning, in the sense of the standalone application of one learning algorithm. Each methodology is scrutinized regarding its strengths, limitations, and practical applicability across diverse scenarios. The aim is to provide a comprehensive understanding of the intrinsic characteristics of these algorithms and their roles in blockchain anomaly detection.

Moving beyond the first category, the second category focuses on methods that attempt to succeed in detecting anomalies by combining two or more learning algorithms using various types of combinations such as cascade or parallel. Cascaded combinations take center stage, while the parallel ones highlight their efficacy in capturing complex structures and patterns within the data. This exploration deepens our understanding of unsupervised methods and their applicability in scenarios requiring an appropriate representation of the information.

Finally, the third category delves into the dynamic interplay between supervised and unsupervised learning techniques. The investigation explores methodologies that leverage both paradigms to address complex problems within the same domain. By fusing the strengths of supervised and unsupervised approaches, these ensemble strategies aim to enhance overall model performance.

7.1. Category 1: Solitary Implementation of Unsupervised Learning

Each method belonging to this category employs a single unsupervised learning algorithm to deal with certain problems related to anomaly detection. Their diversification relies on the type of studied anomalies, the way they process the data and the data structures used, the evaluation metrics employed to verify the results, and the programming framework. Tables 4 and 5 report the methods and the corresponding characteristics.

A public blockchain ledger is used in many different ways for legal purchases, gambling, illegal activities, and so on. Thus, it is important to attain a clear classification of those activities and assign them to specific users, shedding light on the exact usage of the network, which will further provide helpful insights in determining illegal activities [141].

In a typical public blockchain network, there exists a very large number of pseudo-anonymous addresses. Thus, to effectively identify malicious user activities, it is important to associate addresses with users. Although such a process can be very helpful, it constitutes a challenging problem due to the huge amount of data existing in public blockchain networks [124]. Any establishment of such kinds of associations would forge solid means to come up with effective anomaly detection tools. Unsupervised learning in the form of data clustering has been exemplified as a very trustworthy strategy to determine and aggregate the addresses linked to a user [124].

As such, the quest to elaborate on the relationship between addresses and users has led to various research directions with many forms and shapes, employing information coming mainly from the transaction record. For example, the extraction of an appropriate set of features that describe properties of transaction addresses can be involved in strategies that attempt to timely detect harmful activities. This kind of strategy is strongly related to the real-time identification of potentially harmful transactions and provides certain advantages such as taking in-time actions by the user to prevent them [130]. However, the main problem involved has to do with the complexity of the procedure expressed in power and bandwidth resources needed for such an attempt. Deepa and Akila [128] proposed an approach to tackle that problem by using the transaction history of a private blockchain network. They employed the erasure coding technique [142] to handle the missing data and represented the resulting dataset as time series, which enabled the usage of the rolling window aggregation method [129] to obtain a set of features described by information related to the user addresses. Having extracted the features, the k-Means algorithm was put in place to identify several anomalous activities such as fraudulent transactions, double-spending attacks, DDoS attacks, data falsifying, and node capture.

The detection of multiple addresses controlled by the same user to keep the change accounts has also been investigated to understand the differences between malicious and benign activities. In general, change addresses are employed to thwart replay attacks, and they define the very core of the temporal nature of public blockchains (especially for Bitcoin), providing private protection by increasing the anonymity of the user. Therefore, their usage appears to be effective in giving certain insights regarding illegal attacks. In this regard, temporal features related to the transactions' properties can be included. Such an approach was developed by Chaudhari et al. [143], who considered all facets involved in public blockchain transactions by using equally important non-temporal and temporal features, which were extracted by representing the blockchain data in the format of user and transaction graphs as presented in [78]. To identify the accounts belonging to the same user, they developed multi-input heuristics that revealed corresponding change addresses. Then, they applied the k-Means to obtain clusters of addresses and detected malicious accounts in terms of the cosine similarity with other benign addresses belonging to the same cluster.

In the case of the Bitcoin network, a key point is to acquire incidence relations between addresses. Typically, incidence relations enable the creation of a graph-based representation of them, which further can assist the implementation of cluster analysis over the graph data. In this direction, Zheng et al. [144] proposed to use a Gephi graph to represent the address data and applied the Louvain community clustering method to obtain incidence relation between users to anonymize the corresponding transactions with the advantage of improving the traceability of the Bitcoin movement and achieving a better future utilization. The Gephi graph is generally used in two forms: (a) users and transactions are represented as vertices and edges, respectively, and (b) transactions and users are represented as vertices and edges, respectively.

The use of graph-based data has been frequently studied in supporting the identification of distinct theft attacks, mainly for the Bitcoin transaction data, predominantly in instances where reported thefts have transpired [138,144]. The related investigations concentrate on transforming the Bitcoin data into graph-based structures using several features related to the graph vertices (i.e., user nodes) and edges (i.e., transactions). In [138], the k-Means was applied to detect abnormalities reported on the graph such as the all in vain theft, stone man loss, and mass Bitcoin theft. The overall evaluation was conducted considering the relationship between the k-Means cost function and the number of clusters, thus resulting in a type of optimal clustering approach. In [145], the authors discussed the detection of double-spending attacks by transforming the Bitcoin network into a directed acyclic graph, where vertices corresponded to blocks that were created by the miners. The advantage provided by this transformation was that the blocks created by an attacker are not well connected in the graph, and they can be easily detected using specialized clustering approaches, such as spectral clustering, to categorize the graph's vertices into malicious and not malicious.

Transaction graphs have also been used in identifying information leaks. It has been shown [146] that synchronization and timing of messages of transactions in the Bitcoin network may leak information about their origin, thus enabling their exploitation by connected adversarial nodes. As a result, the timing of transaction messages can expose details about their origin, making them susceptible to manipulation by well-connected adversarial nodes. To investigate this issue (i.e., the information leaking in transaction messages by adversarial nodes), Biryukov and Tikhomirov [146] analyzed the network traffic, using the k-Means algorithm to cluster transactions based on the node that first introduced them into the network. The resulting methodology encompassed a procedure to assign weights to nodes' IP addresses, considering propagation timestamps. The Bitcoin data were collected by the bcclient [147], and the clustering implementation scheme was evaluated by the Rand index [148].

In the Ethereum network, the use of cluster analysis over smart contracts has been viewed as a tool to allow for easier contract analysis and detection of security issues, such as malicious contracts. Due to the inherent tabular format of the Ethereum data, the feature space upon which the clustering is to be applied is defined straightforwardly. In [135], Norvill et al. used the Ethereum data [149] and focused on a specific eco-efficient smart contract approach, where the affinity propagation method [111] encircled the k-Medoids [137] to quantify the similarities between pairs of contracts using the several distance functions. To encapsulate the essence of the resulting partition succinctly, the clusters were assigned labels based on a tokenization procedure, while the evaluation process was conducted using the frequency distribution values, which are in the position to assess the homogeneity within each cluster.

A particular category of anomaly detection methods relies on viewing users and types of users as entities and then partitioning those entities into groups with similar characteristics for the identification of the corresponding behavior patterns based on the transactions performed by those entities [124]. Hence, it appears that the users' behavior patterns become decisive tools in determining malicious behaviors. A promising procedure to detect and analyze behavior patterns relies on using time series (i.e., sequences) representation of the data related to user nodes. Works focused on that issue were separately developed by Huang et al. [133] and Kumari and Catherine [134]. The former defined the behavior patterns as changes in the transaction amount over time for a specific user node, while the latter as the time needed to execute each transaction as well as the amount transferred by that transaction. Both works used the Dynamic Time Warping (DTW) measure to quantify dissimilarities between sequences and variants of the k-Means algorithm to group the sequence patterns into a predefined number of clusters, where the anomalous patterns were considered, i.e., those that did not conform to any cluster representative (i.e., cluster center). Similar strategies consist of developing effective mechanisms to quantify behavior patterns in a blockchain ledger, such as the detailed auditing of log-in chain event incidents. This mechanism provides the ability to view and treat the transactions belonging to a certain block as sequences of events, each of which is assigned to a certain time interval. As a result, it can be very helpful in tracing a chain (i.e., sequence) of events preceding a particular event incident with the ultimate purpose of finding out whether that incident is malicious or not. In [132], such kind of chain events were defined as patterns, and the T-patterns method [150] was adopted to trace the event chains, which are then clustered in terms of an agglomerative hierarchical clustering procedure [151] to identify suspicious chains of events as outliers. In [106], a blockchain network specially designed for device management in IoT applications was developed. The primary assumption was that an effectively trained anomaly detection model is in the position to distinguish behaviors that deviate from normal ones, thus possessing the capability of recognizing new threats entering the network without any further learning procedures. The framework consisted of two stages. The first encompassed an anomaly detection procedure by adopting the extended Markov model. The second implements and evaluates the anomaly detection model in a well-designed blockchain-based distributed IoT environment.

Cluster analysis has also been used in supporting consensus protocols [152–160] For example, Khenfouci et al. [125], to avoid data tampering and fraudulent activities, developed a customized clustering-based consensus protocol to carry out a decentralized consensus mechanism, according to which the k-Means was applied locally by multiple competitive miners. The methodology comprised four layers (i.e., data layer, network layer, blockchain layer, and machine learning layer) and had two main actors: management and miner. Upon the convergence of the k-Means, each miner embedded the resulting block into the network.

Table 4. The characteristics of the methods belonging to Category 1, which are given in terms of types of anomalies they used for, learning algorithms, and evaluation techniques.

Method	Types of Anomalies	Unsupervised Learning Methods	Evaluation Method
Kumari and Catherine [134]	Double-spending attack	k-Means	Within cluster distortion
Norvill et al. [135]	Malicious smart contracts, DAO attack	k-Medoids	Frequency distribution score
Huang et al. [133]	Malicious node behavior	Behavior Pattern Clustering (custom modification of k-Means)	Precision
Kinkeldey et al. [124]	Malicious address behavior	k-Means	Cluster visualization with the BitConduite interface
Khenfouci et al. [125]	Fraud detection	k-Means	Precision, silhouette score, accuracy, F1-score
Zambre and Shah [138]	All in vain theft, stone man loss, mass bitcoin thefts, malicious user identification	k-Means	Within cluster standard deviation
Epishkina et al. [132]	Malicious behavior patterns	Agglomerative hierarchical clustering	Ratio statistical distance
Mirsky et al. [106]	Intrusion-based adversarial attacks in IoT environment	Extended Markov model	Probability scores, false positive rates
Deepa and Akila [128]	Advanced attacks centered on the heresies of safety strategies, DDoS attacks	k-Means	% detection accuracy
Swaroopa and Sharma [145]	Double-spending attack	Spectral clustering	Several spectral properties
Shi et al. [152]	Malicious network activities	k-Means++	Fowlkes–Mallows Index [135]
Zheng et al. [144]	Malicious Bitcoin transactions	Louvain algorithm	Louvain runtime efficiency
Monamo et al. [127]	Fraud detection	Trimmed k-Means	Within cluster sum of squares
Shayegan et al. [160]	Theft attacks (stone mass loss, Stefan Thomas loss, all in vain theft, mass MyBitcoin theft, Linode Hacks, Bitfloor theft, and Cdecker theft)	Trimmed k-Means	Cluster dispersion rate
Biryukov and Tikhomirov [146]	Information leaking in transaction messages by adversarial nodes	k-Means	Rand score
Chaudhari et al. [143]	Malicious addresses	k-Means	F-measure, precision

Table 5. The characteristics of the methods belonging to Category 1, which are given in terms of types of network type, data source and structure, and programming framework.

Method	Network Type	Data Source	Data Structure	Programming Framework
Kumari and Catherine [134]	Private	Artificially generated	Transaction sequences	Python
Norvill et al. [135]	Public (Ethereum)	etherscan.io	Smart contract codes as sequences of opcodes	Not reported
Huang et al. [133]	Private (stock trading dataset)	Real blockchain application data on stock trading	Sequences of transaction data	Not reported
Kinkeldey et al. [124]	Public (Bitcoin)	Bitcoin core client	Tabular	Python, JavaScript/D3
Khenfouci et al. [125]	Private	UCI repository	Tabular	Go language, Go-LibP2P, Ubuntu System
Zambre and Shah [138]	Public (Bitcoin)	Publicly available data	Graph-based	Not reported
Epishkina et al. [132]	Public (Bitcoin)	Bitcoin Core client	Sequences of transaction data	Not reported
Mirsky et al. [106]	Private (IoT environment)	Specially designed IoT database	Tabular	C++
Deepa and Akila [128]	Private	Transaction data (private blockchain network)	Time sequences of transaction data	Python on Anaconda Framework
Swaroopa and Sharma [145]	Private	Custom data	Graph-based	Python
Shi et al. [152]	Private	Custom data	Binary protocol messages	Python
Zheng et al. [144]	Public (Bitcoin)	Bitcoin historical transactions	Graph-based	Python
Monamo et al. [127]	Public (Bitcoin)	University of Illinois	Tabular	R programming language
Shayegan et al. [160]	Public (Bitcoin)	ELTE Bitcoin Project	Tabular	Matlab
Biryukov and Tikhomirov [146]	Public (Bitcoin)	Bitcoin Testnet	List Structure	Python–Scikit Learn
Chaudhari et al. [143]	Public (Bitcoin)	Bitcoin Core client	Graph-based	Python–Scikit Learn

A crucial factor in identifying illegal user activities is the discrimination between different types of unknown network protocols regarding the respective security issues, as far as the effectiveness of clustering difficulties caused by different protocol message lengths is concerned. Shi et al. [152] employed the k-Means++ method [153] to address this issue, where the method extracted the maximum frequent sequences from the binary protocol messages using a minimum support threshold, while they used an algorithmic methodology based on the Bide algorithm to mine the maximum frequent sequences [154]. The evaluation of the whole approach was based on the Fowlkes–Mallows Index [155].

It has been stated that k-Means does not perform well in blockchain anomaly detection problems [156,157]. One reason behind this is that the algorithm appears to be very sensitive to initialization, and therefore, the resulting clusters are unbalanced in the sense that the utilizations of the corresponding clusters appear to be very different from each other, without obtaining a certain level of equalization [158]. However, traditionally, it has been treated as a basic tool in blockchain anomaly detection, especially some of its variants.

Monamo et al. [127] embedded the trimmed k-Means [159] into a framework, which performed object clustering within a multivariate setup. The final number of clusters was determined through optimal clustering, where the sum of square distances within clusters played the role of the performance index. The trimmed k-Means was also used in [160] to develop a collective anomaly detection approach, diverging from the conventional methods in that instead of implementing anomaly detection considering individual addresses and wallets, the study focused on scrutinizing anomalies at the user level, where the available dataset was taken from Kondor et al. [161]. An interesting result of the study indicated that anomalies were more conspicuous among users with multiple wallets.

7.2. Category 2: Combining Unsupervised Learning Algorithms

The combinations of several unsupervised algorithms in blockchain anomaly detection have been carried out in different ways depending on the problem at hand. The methods studied in this section fall into two combination types, namely, cascade and parallel, the basic structures of which are illustrated in Figure 4. Tables 6 and 7 present the characteristics of methods belonging to this category.

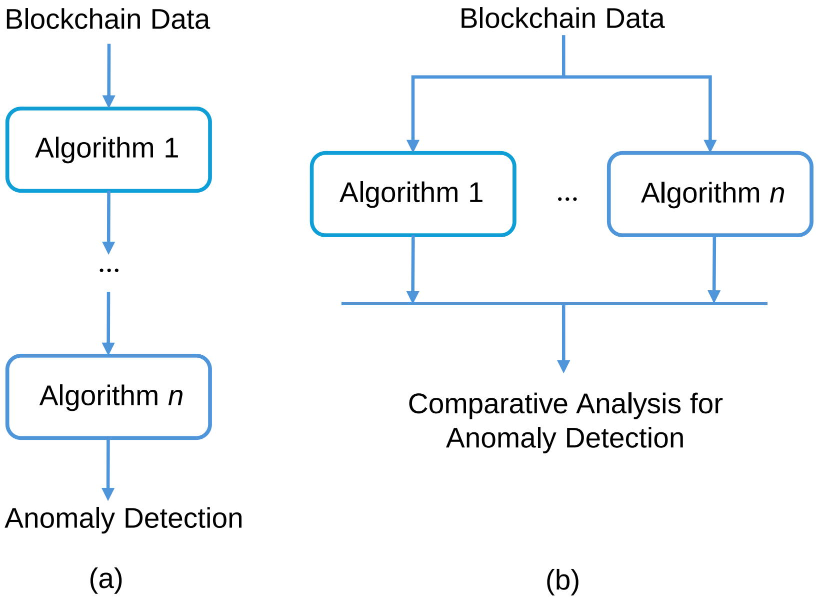

Figure 4. Combination types between unsupervised learning algorithms for blockchain anomaly detection: (**a**) cascade type and (**b**) parallel type.

The cascade type describes methodologies that apply in sequence at least two unsupervised learning algorithms. In most cases, only two algorithms are used, where the results of the first are further processed by the subsequent one. At a practical level, the first focuses on identifying blockchain anomalies, while the second attempts to cluster those anomalies into groups, each of which corresponds to a general type of anomaly, thus obtaining the types of anomalies involved in the problem at hand.

The parallel type usually refers to the implementation of several unsupervised learning techniques for the same problem to determine the one with the best performance.

These two combination types are analytically described within the next subsections in terms of the problems they attempt to address.

7.2.1. Cascade Combination Type

In general, it is widely assumed that any suspicious user-based and/or transaction-based behavior is likely to act as a proxy for anomaly behavior. It is also well accepted that any solution to that problem can be extended to other network settings that may not concern financial transactions. To address that issue, Pham and Lee [156] represented the Bitcoin data in the form of two graph structures to encode the users (i.e., users' graph structure) and the transactions (i.e., transactions' graph structure), according to the methodology of Reid and Harrigan [78]. Considering both graphs they extracted a set of 12 features that described the user and transaction properties of the respective nodes, and applied three unsupervised learning mechanisms, namely, the k-means, Mahalanobis distance, and the one-class SVM. However, they found that the k-Means clustering algorithm was not effective enough for anomaly detection. Therefore, in their subsequent study [157], they improved the performance of the k-Means algorithm using the local outlier factor (LOF) method [115]. In particular, they used the k-Means in sequence with the LOF algorithm, where the LOF indices were determined by the k-Means. The number of clusters used by the k-Means algorithm was evaluated by optimal clustering in terms of the cross-entropy measure.

The study of the cryptocurrency data for identifying market manipulation can provide all stakeholders (i.e., traders, investors, etc.) with insightful points that concern the assets they are interested to sell or buy. Such trade-based market analysis was carried out in [162]. To effectively detect contextual-based anomalies in the form of outliers, the authors used tree-based data representation and developed an unsupervised learning algorithm that combined in sequence the KDE-Track algorithm [163] and the isolated forest method [120]. The former assumed that values in sparse regions are likely to be identified as outliers, while the latter was based on the observation that the mean path length from the tree root to an anomaly leaf is shorter than the paths to normal tree nodes.

The in-sequence implementation of k-Means and RolX [164] was investigated in [165]. The RolX constitutes an unsupervised learning mechanism that attempts to classify the nodes of a graph in several classes (called roles), each of which contains nodes with similar structural features. The authors represented the data as user graphs in a similar way to the approach in [78]. The key issue was to use the k-Means to categorize nodes with multiple transactions (called hubs), and to use the RolX algorithm to assign to each hub a role able to identify anomalous user behaviors related to money laundering/mixing services.

Turner et al. [139] discussed the problem of identifying ransomware attacks in the Bitcoin network. They employed a graph-based representation of the Bitcoin data extracted from the wallet explorer API [166] with random seed addresses. To accomplish this task, they used in sequence three unsupervised learning algorithms, namely, the Louvain algorithm to preprocess the graph data, the Deepwalk embeddings method [112] to carry out the feature extraction process, and the k-Means algorithm to evaluate the risky nodes in terms of the cosine similarity measure. Graph embeddings have also been used in coping with the pseudonymity established between users and transactions in the Bitcoin network. For example, to reveal relationships hidden in the data, Shah et al. [167] determined patterns that are linked to those relationships by setting up a cluster analysis based on embedding feature generation. Embeddings were created by the utilization of the variational graph autoencoder [118] and explainable k-Means [168]. The obtained clusters' visualization was underpinned by the Kohonen self-organizing map, which ultimately provided informative insights into the way the design parameters defined the clusters' structure.

An interesting problem is the simultaneous detection of multiple malicious activities. An effective way to accomplish this task is to use in sequence two unsupervised learning methods, where the first obtains a partition of the blockchain data, and the second elaborates on the outliers obtained previously and generates a partition of them into several clusters each of which corresponds to one malicious activity. Such a methodology was reported by Sayadi et al. in [105], where the bitcoin data [169] were processed in sequence by the one-class support vector machine algorithm that identified the outliers and then the use of k-Means obtained four clusters corresponding to the DDoS attack, double-spending attack, 51% vulnerability, and selfish mining attack.

Table 6. The characteristics of the methods belonging to Category 2, which are given in terms of combination type, type of anomalies they used for, learning algorithms, and evaluation techniques.

Method	Combination Type	Types of Anomalies	Unsupervised Learning Methods	Evaluation Method
Pham and Lee [156]	Parallel	Anomalous behavior as a proxy for suspicious users and transactions	One-class SVM, Mahalanobis distance, k-Means	Dual evaluation (custom metric)
Pham and Lee [157]	Cascade	Fraud detection	One-class SVM, local outlier factor, k-Means	Dual evaluation (custom metric)
Sayadi et al. [105]	Cascade	DDoS attack, double-spending attack, 51% vulnerability, selfish mining attack	One-class SVM, k-Means	Silhouette score
Saravanan et al. [170]	Parallel	Hacked transactions, fraudulent activities, money laundering	Isolated forest, k-Means, autoencoder, clustering based local outlier factor	Accuracy, precision, recall, F1-score
Sun et al. [171]	Cascade	Malicious user accounts	t-SNE algorithm, Birch algorithm	Customized methodology
Zhang et al. [172]	Parallel	Abnormal transactions	k-Means, generative adversarial network	Precision, recall, F1-measure
Kampers et al. [162]	Cascade	Cryptocurrency market manipulation	KDE-Track algorithm, isolated forest	Domain expert reviews, F1-score
Hirshman et al. [165]	Cascade	Money laundering Mixing Services	k-Means, Role eXtraction (RolX) algorithm	Factorization error
Turner et al. [139]	Cascade	Ransomware attacks	Deepwalk, PCA, k-Means	Cosine similarity measure of risk
Shah et al. [167]	Cascade	Outlier pattern detection (wallet authority detection)	Explainable k-Means, Variational autoencoder, Self-organizing maps	True positive rate, Cluster distortion measure
Agarwal et al. [173]	Parallel	Phishing, gambling, Ponzi scheme	k-Means, HDBSCAN, spectral clustering, agglomerative clustering, one-class SVM	Silhouette score

Table 7. The characteristics of the methods belonging to Category 2, which are given in terms of types of network type, data source and structure, and programming framework.

Method	Network Type	Data Source	Data Representation	Programming Framework
Pham and Lee [156]	Public (Bitcoin)	University of Illinois Urbana	Graph-based	Python, NetworkX library
Pham and Lee [157]	Public (Bitcoin)	Stanford Network Analysis Project	Graph-based	Python, NetworkX library
Sayadi et al. [105]	Public (Bitcoin)	Bitcoin blockchain using Blockchain.info API	Tabular	Python on Spyder/Anaconda, Orange3 API
Saravanan et al. [170]	Public (Bitcoin)	IEEE Data Port, Kaggle	Tabular	Not reported
Sun et al. [171]	Public (Ethereum)	Etherscan blockchain explorer APIs	Eigenvector-based	Not reported
Zhang et al. [172]	Public (Bitcoin)	Reid and Harrigan [77]	Graph-based	Python–Tensorflow
Kampers et al. [162]	Public	Amazon Web Services cloud	Tree-based	Python
Hirshman et al. [165]	Public (Bitcoin)	Bitcoin transaction network dataset	Graph-based	Not reported
Turner et al. [139]	Public (Bitcoin)	Walletexplorer API	Graph-based	Python
Shah et al. [167]	Public (Bitcoin)	Bitcoin full historical data	Graph-based	Python, Apache Spark
Agarwal et al. [173]	Public (Ethereum)	Etherscan blockchain explorer APIs	Tabular	Python

Considering the Ethereum network, there are two types of accounts, namely, externally owned accounts (EOAs), which represent the users in the form of a hash value, and smart contract accounts. A typical approach is to study certain types of transaction relationships between EOAs and smart contract accounts, which along with the above two types of accounts form a heterogeneous Ethereum network. The analysis of Ethereum data at the present stage is mostly based on the statistical characteristics of Ethereum nodes and lacks analysis of the transaction behavior between them. However, the presence of several features renders such kinds of approaches very complex to implement. Due to the inherent tabular format of Ethereum data, dimensionality reduction approaches can be conveniently applied to carry out a transformation of the original data into points in a low dimension, which can also enable their visualization. Effective dimensionality approaches used in blockchain anomaly detection are the PCA and the t-SNE [121,139]. Relative methodologies obtain clusters of users and smart contracts by incorporating transaction information coming from Ethereum blocks to perform identity detection of malicious users [170,171]. This can be accomplished by first using a node embedding method to obtain the eigenvectors for the EOA and smart contract nodes. Then, reduction dimensionality approaches [121] can be used to assist various clustering algorithms (e.g., the Birch algorithm) to calculate the user and smart contract clusters and detect malicious user accounts.

7.2.2. Parallel Combination Type

The usage of generative adversarial networks [119] has been seen as an alternative tool to surpass the need for labeling approaches, usually employed in supervised learning implementation, and perform an effective anomalous detection strategy. The benefit of using GANs comes from the fact that it does not require any time-consuming labeling procedure for the dataset. In this direction, Zhang et al. [172] applied a GAN-based mechanism to the Bitcoin transaction data. The data were represented as a user-based graph [78]. After standardizing the dataset, they trained a GAN network and for comparative reasons a k-Means model, where the former appeared to outperform the latter.

On many occasions, comparative analysis between several unsupervised learning algorithms may give valuable results concerning the relative behavior of the above methods for the problem at hand. Such an analysis is reported in [170], where the performances of four well-known unsupervised methods (i.e., isolated forest, k-Means, autoencoder, and cluster-based local outlier factor (CBLOF)) were evaluated using bitcoin data collected from diverse sources such as IEEE Data Port and Kaggle.

As mentioned above, in the Ethereum network, two transaction types are carried out in terms of smart contracts, namely, internal and external. Smart contracts that are not labeled as malicious cannot be considered as such (even in the case it is malicious), but rather only to hypothesize (i.e., suspect) that it is malicious. From a clear point of view, an account is considered malicious if there is proof that it is involved in (i.e., carries out, facilitates, and/or supports) illegal activities such as Ponzi schemes, Lendf Hack, Akropolis Hack, Phishing, Gambling, etc. Some of these activities are motivated by social behavior, while others take place due to the exploitation of bugs and vulnerabilities. Therefore, since there are many different types of activities originating from different motivations, any associations between vulnerabilities and particular malicious activities appear to be a difficult task. Unsupervised learning can be effectively used to identify smart contracts showing malicious behavior (even when the contract is not labeled as malicious), while at the same time, it has vulnerabilities. In this regard, it has been shown that there is a strong correlation effect between vulnerabilities in smart contracts (SCs) and illegal activities on the cryptocurrency platform. In [173], the authors studied the vulnerabilities of SCs using the concept of severity score [174]. As it was expected that the severity score would assist in identifying the correlation between vulnerabilities and malicious behaviors, they added it to the set of features originally extracted from the dataset taken from Etherscan [149]. To carry out the anomaly detection they applied, over the obtained feature space, several unsupervised learning algorithms such as the k-means, HDBSCAN, spectral clustering, agglomerative clustering, and one-class SVM, while the silhouette score was used to quantify and compare the resulting performances.

7.3. Category 3: Combining Unsupervised and Supervised Learning Algorithms

In this section, the synergy between unsupervised and supervised learning is studied. In the existing literature, several papers elaborate on such kinds of learning schemes to enhance the performance of either the former or the latter (or both of them). The impetus to choose the synergy between these two learning paradigms lies in the way they are implemented. While unsupervised learning can detect anomalies, supervised learning can predict and classify anomalies. These two functions can be thought of as having equal contributions to the final result. As such, herein, two pipelines are studied, namely, combination type 1, where unsupervised learning is used to assist the prediction capabilities of supervised algorithms, and combination type 2, where the results obtained by the implementation of supervised methods are further processed by unsupervised ones. It is worth noting that the vast majority of algorithmic schemes existing in the literature belong to the first type (i.e., combination type 1). Figure 5 depicts the general pipelines of the two combination types. Tables 8 and 9 present the characteristics of methods belonging to this category.

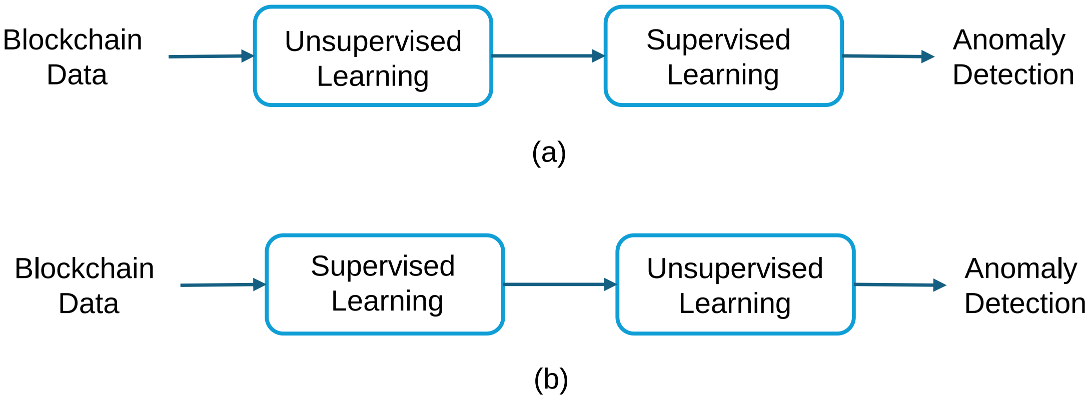

Figure 5. Combination types between supervised and unsupervised learning for blockchain anomaly detection: (**a**) type 1 and (**b**) type 2.

7.3.1. Combination Type 1

As blockchain networks evolve, their dynamic properties provide a foothold to better distinguish between malicious and benign activities. In that sense, as the illegal activities (e.g., money laundering, Ponzi schemes, phishing, scamming, etc.) increase, the employment of predictive classification of the blockchain entities (e.g., accounts, transactions, etc.) is of major importance in establishing security [175]. Supervised learning can effectively address that issue since it is very effective in providing accurate predictions.

The usage of supervised learning is based on the existence of labeled data. However, in blockchain technologies, there is a lack of labels mainly due to two reasons. First, data labeling relies on clearly denoting and assigning ground truth values, and it can be accomplished by adopting data annotation techniques. A major problem related to data annotation is that it requires a huge workload, and therefore, it is difficult to obtain a sufficient and accurate dataset suitable for supervised algorithms. In addition, labels resulting from law enforcement investigations are not immediate and, as indicated above, annotation is costly [29]. Second, the evolving complexity of blockchain ecosystems has also led to an increase in the complexity of illegal activities making the process of identifying all (or even most) of the entities and parameters a difficult problem. As such, the use of unsupervised learning becomes more appealing having better application prospects because it does not need to mark the available dataset [172].

It turns out that without designing an effective strategy to handle the above issues, the usage of supervised algorithms in blockchain anomaly detection might be limited. An effective solution is to integrate unsupervised learning to preprocess the data and generate the appropriate data labels (i.e., ground truth assignments).

In that direction, Baek et al. [176] made a valuable contribution to the labeling process of wallets by providing a two-step model that applied in sequence the Expectation Maximization and the k-means algorithms to preprocess the data. The data were taken from Binance [177] and ethescan.io [149] with a Python API. The authors extracted several features related to the wallets' properties and applied the Expectation Maximization to detect the outliers in the feature space. Then, the implementation of k-means was set up to further cluster the outliers and provide the appropriate labels to each one of the resulting clusters (i.e., malicious or benign wallets). The labeled data were represented in terms of a decision tree structure, and the random forest algorithm was used to obtain effective predictions as far as the malicious wallets are concerned.

A different point of view concerns the collection of the addresses involved in the detection of malicious activities, such as scams, in public networks [178]. Any algorithmic (i.e., automatic) elaboration on the transactions linked to a scam or any other type of threat, that has the purpose of quantifying the effect imposed by that threat, can be implemented after the above-mentioned collection of the addresses. Moreover, the difficulties involved in the above process increase since many addresses related to illegal activities may not be publicly available. In this case, those addresses might privately establish contact with the registered users. A solution to this problem, for the test case of Ponzi scheme threats in the Bitcoin network, was given in [179]. The authors adopted a graph-based representation of the data and additional datasets related to advertisements of investment programs taken from Reddit [180] and bitcointalk.org [181] that on many occasions hide Ponzi schemes. Then, they employed an unsupervised mechanism, based on "multi-input" heuristics [100], with the ultimate purpose of obtaining and analyzing clusters of transactions to detect Ponzi schemes and create a new dataset that contains ground-truth data that indicate the malicious and the benign users. In the final step, the above dataset was fed into several supervised algorithms, such as Bayes network and random forest, to provide predictions for the illegal activities.

The use of partially labeled datasets was also studied by several authors. Commonly, the labeled data constitute a small portion of the whole available dataset. As an example, in the Elliptic dataset [182], 2% of the instances are labeled as illicit, and 21% of the instances are labeled as illicit. A straightforward way to support the implementation of supervised learning is to manually increase the existing labeled data. Under the usual scenario, there are two labels corresponding to malicious and non-malicious. Unsupervised learning can be applied to partition the whole dataset into two distinct clusters and use the above information to provide labels to a part or all the data. As such, the unlabeled data are aligned with their respective clusters and augmented with the already labeled data. In the subsequent step, the whole dataset undergoes a classification process in terms of some prominent classifiers [183].

Considering the Ethereum network, adversarial attacks have risen. An adversarial attack attempts to create artificial data that follow the distribution of the original data, thus appearing to be almost identical. This imposes risky situations for decentralized applications (dapps) like Ethereum, as illegal activities can be easily hidden in the generated artificial data. In [184], this issue was addressed by implementing in sequence a generative adversarial network (GAN) and a recurrent neural network (RNN). The GAN was used to generate synthetic data that mimics the behavior of normal transactions in Ethereum, while the RNN was used to classify transactions as normal or adversarial. The above implementation generated a dataset upon which an LSTM network was trained to predict adversarial such as the cyber kill chain attack.

The digital signing of the transaction appears to be vulnerable to certain malicious issues because the signing is usually executed manually by the user. The effects imposed by this fact exponentially increase when the user executes frequent transactions over long periods since in this case a superficial signing process is required. Thus, malicious counterparties can take advantage of such behavior seeking to convince the user to sign a transaction with adverse impacts on the user's digital resources, e.g., a transaction with tampered data. An approach for automatic digital signing is described in [130], where historical transaction data on Ethereum were represented as time series and elaborated by the rolling window aggregation method [129] to extract certain features (e.g., transactions' timestamps and values), which enabled the usage of the isolated forest algorithm to perform anomaly detection in terms of detecting the outliers of the resulting partition. Then, the random forest was employed to conduct the classification and the prediction of the anomaly labels as outputted by the isolation forest. The overall algorithmic structure yields a personalized malicious transaction detection model.

7.3.2. Combination Type 2

Data partitions obtained by unsupervised learning applied to unlabeled data may fail to correctly perform anomaly detection because data corresponding to illicit behaviors might be hidden within clusters of licit transactions. To provide a labeling procedure, resolving the above problem for the elliptic dataset, Lorenz et al. [185] used several classifiers trained on the existing labeled data and projected the results over the whole dataset. Then, they employed the approach developed in [186] to split and process the data in terms of several unsupervised learning methods (e.g., LOF, one-class SVM, isolation forest). To this end, they put in place an active learning approach [187] to match the performance of a fully supervised baseline by performing money laundering detection assuming minimal access to labels.

The synergy between unsupervised and supervised learning has been recognized as a very supportive tool in Ethereum data analysis. Here, the accounts are of two main types, namely, externally owned accounts or smart contracts. It is well stated that Ethereum smart contracts might include hidden malicious schemes, which may not be known to the users because those schemes come in the form of "high-yielding: advertisements" [179,188]. A major property of the Ethereum network is spotlighted by the fact that for the accounts the aggregated degree distribution pursues the power law [189]. This property implies that the incorporation of dynamic (i.e., temporal) features (e.g., inter-event time, attractiveness, busty behaviors of in- and out-degree, etc.) might impose strong capabilities in detecting/predicting malicious behaviors. Temporal features can be extracted from graph-based [175,190] or tabular-based [191] representation of the Ethereum data.

Table 8. The characteristics of the methods belonging to Category 3, which are given in terms of combination type, type of anomalies they used for, learning algorithms, and evaluation techniques.

Method	**Combination Type**	**Types of Anomalies**	**Unsupervised Learning Methods**	**Evaluation Method**
Sachan et al. [29]	Type 2	Domain names crypto jacking detection	k-Means	Silhouette score
Agarwal et al. [175]	Type 1	Phishing, spamming, scams, and Ponzi schemes	K-means, DBSCAN, HDBSCAN, and one-class SVM	Silhouette score
Baek et al. [176]	Type 1	Malicious wallets	Expectation maximization, k-Means	Precision, recall, F-measure
Bartoletti et al. [179]	Type 1	Ponzi schemes	Multi-input heuristics	Precision, F-measure
Boughaci et al. [183]	Type 1	Malicious transactions	k-Means	Precision, recall
Rabieinejad et al. [184]	Type 1	Malicious activities/cyber kill chain	GAN	Trx index

Table 8. *Cont.*

Method	Combination Type	Types of Anomalies	Unsupervised Learning Methods	Evaluation Method
Podgorelec et al. [130]	Type 1	Malicious transactions	Isolated forest	Ranks for time frames of feature extraction process
Lorenz et al. [185]	Type 2	Money laundering	Local outlier factor, isolation forest, one-class support vector machine	F1-score
Sachan et al. [190]	Type 2	Domain Names crypto jacking detection	k-Means	Silhouette score
Agarwal et al. [191]	Type 2	Malicious detection through adversarial activities	k-Means, GANs	Precision, recall, F1-score
Agarwal et al. [192]	Type 2	Malicious detection through adversarial activities	k-Means, GANs	Precision, recall, F1-score

Table 9. The characteristics of the methods belonging to Category 3, which are given in terms of types of network type, data source and structure, and programming framework.

Method	Network Type	Data Source	Data Representation	Programming Framework
Sachan et al. [29]	Public permissionless	Cisco Umbrella top 1 million Dataset, Indian Government URLs	Graph-based browser metadata	Python, NumPy
Agarwal et al. [175]	Public (Ethereum)	Ethereum transaction data (79 million accounts and Cryptoscam.db dataset)	Graph-based	Python
Baek et al. [176]	Public (Ethereum)	Binance and Ethereum wallets from etherscan.io	Tree-based	Python API
Bartoletti et al. [179]	Public (Bitcoin)	Reddit, bitcointalk.org	Graph-Based	Weka software
Boughaci et al. [183]	Public (Bitcoin)	Elliptic dataset (Kaggle)	Graph-based	Java (Netbeans environment)
Rabieinejad et al. [184]	Public (Ethereum)	Ethereum transaction data	Tabular	Python
Podgorelec et al. [130]	Public (Ethereum)	Etherscan.io	Time Series	Python, Scikit-learn
Lorenz et al. [185]	Public (Bitcoin)	Eliptic dataset	Graph-based	Python, Scikit-learn
Sachan et al. [190]	Public-permissionless	Cisco Umbrella top 1 million Dataset, Indian Government URLs	Graph-based browser metadata	Python, NumPy
Agarwal et al. [191]	Public (Ethereum)	Ethereum.org	Tabular	Python, Keras, NumPy
Agarwal et al. [192]	Public (Ethereum)	Ethereum.org	Tabular	Python, Keras, NumPy

In [175], the temporal features were transformed into vectors to provide a time series facet based on the assumption that this choice can assist the detection of malicious accounts considering past attacks analysis. To elaborate on the data, the authors used a two-step hybrid scheme. The first step applied several supervised algorithms in terms of an AutoML framework that incorporated hyperparameter optimization. In the second step, the results of the supervised learning were further processed by implementing the k-Means to detect accounts that are similar to the above-mentioned malicious accounts. The number of clusters was obtained via optimal clustering using the silhouette score as an evaluation index and treating the number of clusters as hyperparameter in the AutoML framework. In [190], temporal and non-temporal features were defined in terms of Ethereum domain name (DN) metadata and used to detect DNs crypto-jacking activities. The approach followed the above-mentioned AutoML-based synergy between supervised learning and k-Means. In [191,192], several attacks on the Ethereum (e.g., ransomware payments, phishing, scamming, upbit hack, spam token, Ponzi schemes, EtherDelta Hack, etc.) were considered, while addressing issues related to the effective application of k-Means and GANs, where their results were preprocessed by various supervised ML structures such as the Extra-Tree classifier and a neural network in the form of Multiple-Layer Perceptron (MLP). An interesting point of those approaches is to use the partitions coming from the k-Means to study the bias effect imposed by the supervised ML algorithms in different abnormal accounts associated with certain malicious attacks/activities that may be represented in large amounts regarding the corresponding clusters' sizes.

8. Challenges and Future Directions

The subject of blockchain anomaly detection lies at the intersection of information mining strategies and the transformative capacity of blockchain technology. As we explore present methodologies and pave the way for future advancements, numerous challenges and avenues for exploration emerge.

8.1. Scalability and Complexity

One of the most significant challenges in detecting anomalies in blockchain networks is scalability. As the volume and complexity of blockchain data increase, traditional detection methods may struggle to keep pace. Moreover, the heterogeneous nature of blockchain structure consists of various layers and protocols, thus adding complexity. Addressing scalability and complexity issues necessitates a shift towards dynamic trends that can adapt to evolving network systems and accommodate diverse data sources [30].

To mitigate scalability and complexity challenges, future research should embrace the synergy of graph learning and neural network methodologies. Graph learning provides a robust framework for modeling blockchain networks by leveraging their structures. Incorporating neural network techniques such as graph neural networks or LSTM can uncover complex patterns, offering insights into anomalous behaviors while circumventing scalability constraints.

To enhance the ability to detect unusual events, a symbiotic relationship between data mining techniques and the inherent network's resilience must be established. Leveraging unsupervised learning methods, and exploring novel avenues in deep learning approaches, we can uncover latent anomalies within the blockchain ecosystem, enhancing its overall integrity and reliability for future endeavors.

8.2. Generative AI and Adversarial Attacks in Blockchain Anomaly Detection

Anomaly detection techniques in blockchain systems confront a spectrum of threats. Malicious actors may manipulate transactions or introduce spurious data to deceive detection algorithms, jeopardizing system integrity. Additionally, the presence of fraudulent or misleading data within the blockchain can lead to inaccurate anomaly detection, impeding the identification of genuine anomalies. To tackle these threats, innovative mitigation strategies in the ongoing research efforts could concentrate in two directions, namely generative AI (GenAI) and adversarial learning (AL).

8.2.1. Potential Approaches Related to GenAI

GenAI powered anomaly detection (GADE) approaches hold significant promise in enhancing anomaly detection within blockchain networks. GADE methods can complement the existing processes outlined in recent research.

GADE could deal with the challenge of associating addresses with users in public blockchain networks. Unsupervised learning, particularly data clustering, has proven to be a reliable strategy in aggregating addresses linked to a user. By employing GADE-based procedures, cluster analysis could achieve higher accuracy and scalability, rendering the identification of malicious user activities more effective. Moreover, the dynamic interplay between supervised and unsupervised techniques, as explored in recent research, could be further optimized using GenAI approaches. By harnessing the strengths of both paradigms, ensemble strategies powered by GADE could enhance the overall model's performance in detecting complex anomalies within blockchain networks. For example, integrating GADE and unsupervised algorithms could improve their accuracy in detecting anomalies, especially in scenarios where the data are complex. Thus, cyber defenders can improve their intelligence by identifying emerging threats and extracting relevant data. In a nutshell, the GenAI tools can be used in analyzing network traffic data, system output, and large volumes of log files. This could assist defenders to automate and speed up their incident response process. GenAI can also be used in generating secure code and writing secure code. However, attackers can also misuse GenAI to create malicious code, phishing attacks, and social engineering attacks [193].

While the existing approaches provide valuable insights to anomaly detection, incorporation of GenAI approaches could substantially enhance the accuracy, scalability, and efficiency of anomaly detection algorithms.

8.2.2. Potential Approaches Related to Adversarial Attacks

Potential approaches encompass developing robust anomaly detection algorithms resilient to adversarial attacks, implementing mechanisms to ensure the integrity of blockchain data, exploring advanced privacy-preserving techniques, and promoting user awareness and education regarding privacy best practices in blockchain systems. Addressing the above challenges necessitates a multifaceted approach encompassing technical innovations, robust defense mechanisms, and user empowerment.

Unsupervised learning frameworks offer remarkable adaptability as the boundary between normal and anomalous behavior evolves dynamically. In this regard, deep learning assumes a pivotal role in this paradigm, capable of identifying anomalous activity by tapping into the latent features embedded in blockchain transactions.

The future of anomaly detection in blockchain networks is likely to embrace a fusion of typical unsupervised learning and deep learning methodologies, where techniques like autoencoders and GANs can unveil hidden anomalies in blockchain data assisting the overall endeavor. Additionally, delving into temporal aspects such as time series analysis and recurrent neural networks can enhance the accuracy of anomaly detection mechanisms.

In addition, future research should generate robust datasets encompassing various facets of blockchain transactions, serving as benchmarks for evaluating the efficacy of anomaly detection algorithms and fostering interdisciplinary collaboration.

8.3. Distributed Ledger under the Framework of AI

In the current status, the emerging trends in distributed ledger technology (DLT) intersect with the advancements in artificial intelligence. While the focus remains on unsupervised learning methodologies for anomaly detection, potential synergies with DLT and AI present novel avenues for exploration. The development of reliable, interpretable, and explainable AI models capable of governing DLT protocols and smart contracts remains pivotal for ensuring fairness and transparency within blockchain networks. In this direction, scalability and performance overhead issues can be effectively addressed in terms of implementing DLT-based federated AI models [170] able to enhance data privacy and security demands.

DLT's capability to provide transparent and secure ledgers aligns with the need for explainable AI, offering opportunities to enhance the security and reliability of AI systems. Moreover, tokenization of data on DLT-based marketplaces could incentivize data sharing supporting federated learning processes and leading to more diverse datasets for AI models, with considerations for privacy and data security. These developments underscore the necessity for further research to integrate unsupervised learning approaches with DLT-driven AI applications, ensuring robustness, transparency, and privacy in anomaly detection mechanisms.

Furthermore, the convergence of AI with DLT-based consensus algorithms and decentralized coordination requires meticulous exploration to optimize efficiency and mitigate risks. As such, researchers must investigate the complexities of these intersections, balancing innovation with the imperative of maintaining privacy, security, and integrity within blockchain ecosystems.

8.4. On the Effect of the Blockchain Continuous Evolution

While unsupervised learning methodologies offer remarkable adaptability in detecting anomalies, a significant challenge arises from ensuring the ongoing fit and adaptability of the resulting models to the evolving blockchain architectures and components. As blockchain technology continues to evolve with advancements in consensus mechanisms, network protocols, and the introduction of new features, existing anomaly detection models may become outdated or less effective over time. This problem is exacerbated by the decentralized and distributed nature of blockchain networks, which can lead to heterogeneous data distributions and dynamic transaction patterns [194].

To address the challenge of model fit and adaptability, researchers must focus on developing techniques that enable continuous learning and adaptation in real time. This requires the integration of mechanisms for model retraining and updating based on the latest blockchain data and network dynamics. Additionally, the incorporation of techniques such as transfer learning and domain adaptation can facilitate the transfer of knowledge between different blockchain environments, improving the generalization and robustness of anomaly detection models [195].

Moreover, the utilization of blockchain-based governance mechanisms, such as decentralized autonomous organizations (DAOs), can enable collective decision-making processes for model updates and parameter tuning, ensuring community-driven governance. By addressing the challenge of model fit and adaptability, researchers can enhance the long-term effectiveness and reliability of anomaly detection mechanisms in blockchain networks, supporting the integrity and security of decentralized ecosystems.

8.5. Implications of Zero-Trust and Zero-Knowledge Proof Environments

The implications of zero-trust architecture (ZTA) and zero-knowledge proof (ZKP) environment on types of attacks are significant. Since these environments require continuous authentication and verification, traditional attack vectors such as unauthorized access, privilege escalation, and data breaches become more challenging to execute successfully. Attackers must overcome multiple layers of verification, making it more difficult to infiltrate systems or tamper with data [196].

However, despite the enhanced security provided by those environments, new types of attacks may emerge. Attackers may focus on exploiting vulnerabilities in the authentication and verification mechanisms themselves, attempting to bypass or compromise them. For example, they might target weaknesses in the implementation of ZKP's protocols or attempt to deceive the ZTA infrastructure into granting unauthorized access [197].

Anomaly detection approaches in such environments may need to adapt also in order to account for the unique characteristics of ZTA and ZKP. Traditional anomaly detection methods rely on identifying patterns of normal behavior and flagging deviations from these patterns as anomalies. In ZTA environments, where every interaction is treated with skepticism, normal behavior may vary significantly from user to user and over time [197]. Anomaly detection algorithms would need to be more dynamic and context-aware, continuously adjusting their understanding of normal behavior based on real-time observations.

Furthermore, anomaly detection approaches in ZTA and ZKP may need to incorporate additional layers of verification and validation. For example, anomaly detection algorithms could leverage ZKP techniques to verify the authenticity of transactions or user interactions without revealing sensitive information. By combining anomaly detection with ZKPs, organizations can enhance their ability to detect and respond to suspicious activities while preserving privacy and security. It is worth noting that zero-knowledge attacks pose a significant challenge to authentication systems, particularly in decentralized peer-to-peer (P2P) environments. Traditional public key infrastructure models rely on centralized trust servers, such as certification authorities (CAs), to validate the binding of user public keys and identities. However, in decentralized P2P networks, the CA-based models become impractical. Instead, anonymous systems in P2P environments adopt self-signed certificates to authenticate peers, providing anonymity but leaving them vulnerable to man-in-the-middle attacks. In these attacks, an attacker can intercept and impersonate transaction participants without detection due to the lack of knowledge about communication channels. However, a promising approach to mitigating such types of attacks is the pseudo trust mechanism, which embeds the knowledge of the communication path in exchanged messages [197]. By verifying the validity of combined messages, transaction participants can detect impersonation attempts, offering enhanced security.

8.6. Privacy Concerns in Unsupervised Learning for Anomaly Detection

Despite the potential benefits of unsupervised learning in anomaly detection, privacy concerns are likely to arise [62]. As these algorithms often operate without explicit labels or supervision, they may inadvertently reveal sensitive information or patterns that compromise user privacy [36]. For example, clustering algorithms that group similar transactions or behaviors may disclose patterns of user activity, potentially enabling deanonymization or profiling attacks [36]. Similarly, dimensionality reduction techniques applied to blockchain data may expose underlying relationships or correlations, raising concerns about user privacy and data confidentiality [63].

Addressing privacy concerns in unsupervised learning for anomaly detection requires careful consideration of data anonymization techniques, differential privacy mechanisms, and privacy-enhancing technologies [198]. Techniques such as data masking, noise injection, and anonymization algorithms can help mitigate the risk of privacy breaches while preserving the utility of the data [199]. Moreover, integrating privacy-preserving models with federated learning approaches can enable collaborative anomaly detection without compromising individual user privacy.

While unsupervised learning algorithms offer promising solutions, it is imperative to address privacy concerns to ensure the ethical and responsible use of these technologies [199]. By using privacy-enhancing techniques and adopting privacy-preserving models, stakeholders can harness the benefits of unsupervised learning while safeguarding user privacy and data confidentiality.

9. Conclusions

The necessity of unsupervised learning spans diverse domains within the realm of blockchain and emerging technologies, where their presence offers unparalleled adaptability in dynamically distinguishing between normal and anomalous behaviors, crucial for maintaining system integrity amidst evolving threats. As scalability and complexity challenges persist, the current investigation focused on studying the integration of unsupervised learning methods and on identifying certain advantages that render that integration an effective tool, considering both public and private networks. The whole approach encompassed three basic levels of analysis. First, it scrutinized the blockchain anomalies and provided a brief overview of the general machine-learning frameworks used. Second, the data structures employed by several approaches were meticulously reported and analyzed. Regarding this task, it was shown that the way the data are transformed and processed plays an important role in the effectiveness of the anomaly detection model. Third, the most important level that defines the whole analysis was the categorization of the methods that exist in the literature into three distinct categories depending on the way the unsupervised learning algorithms are used and/or combined to tackle the problem at hand. Finally, the survey delineates certain challenges that are related to the field and concern the development of robust, secure, and privacy-preserving systems across various domains, driving technological advancements and innovation in blockchain ecosystems.

Author Contributions: Conceptualization, G.E.T. and C.C.; methodology, C.C. and E.A.; formal analysis, Z.S. and E.M.; investigation, C.C., E.A., Z.S. and E.M.; resources, C.C. and Z.S.; writing—original draft preparation, C.C., E.A., Z.S. and E.M.; writing—review and editing, G.E.T., E.A., Z.S. and E.M.; supervision, G.E.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors thank the anonymous reviewers for their effort to provide very helpful comments that improved the paper.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Saad, M.; Spaulding, J.; Njilla, L.; Kamhoua, C.; Shetty, S.; Nyang, D.-H.; Mohaisen, D. Exploring the attack surface of blockchain: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1977–2008. [CrossRef]
2. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 16 October 2023).
3. Xie, M.; Li, H.; Zhao, Y. Blockchain financial investment based on deep learning network algorithm. *J. Comput. Appl. Math.* **2020**, *372*, 112723. [CrossRef]
4. Sarker, S.; Saha, A.K.; Ferdous, M.S. A survey on blockchain and cloud integration. In Proceedings of the 23rd International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 19–21 December 2020; pp. 1–7.
5. Gan, Q.-Q.; You, R.; Lau, K. Trust in a 'trust-free' system: Blockchain acceptance in the banking and finance sector. *Technol. Forecast. Soc. Chang.* **2024**, *199*, 123050. [CrossRef]
6. Zheng, Z.; Xie, S.; Dai, H.-N.; Chen, W.; Chen, X.; Weng, J.; Imran, M. An overview on smart contracts: Challenges, advances and platforms. *Future Gener. Comput. Syst.* **2020**, *105*, 475–491. [CrossRef]
7. Kose, J.; Leonid, K.; Fahad, S. Smart contracts and decentralized finance. *Annu. Rev. Financ. Econ.* **2023**, *15*, 523–542.
8. Dong, C.; Huang, Q.; Fang, D. Channel selection and pricing strategy with supply chain finance and blockchain. *Int. J. Prod. Econ.* **2023**, *265*, 109006. [CrossRef]
9. Boakye, E.A.; Zhao, H.; Kwame Ahia, B.N. Emerging research on blockchain technology in finance; conveyed evidence of bibliometric-based evaluations. *J. High Technol. Manag. Res.* **2022**, *33*, 100437. [CrossRef]
10. Wang, T.; Wu, Q.; Chen, J.; Chen, F.; Xie, D.; Shen, H. Health data security sharing method based on hybrid blockchain. *Future Gener. Comput. Syst.* **2024**, *153*, 251–261. [CrossRef]
11. Xiang, X.; Zhao, X. Blockchain-assisted searchable attribute-based encryption for e-health systems. *J. Syst. Archit.* **2022**, *124*, 102417. [CrossRef]
12. Uppal, S.; Kansekar, B.; Mini, S.; Tosh, D. HealthDote: A blockchain-based model for continuous health monitoring using interplanetary file system. *Healthc. Anal.* **2023**, *3*, 100175. [CrossRef]
13. Tian, J.; Tian, J.-F.; Du, R.-Z. MSLShard: An efficient sharding-based trust management framework for blockchain-empowered IoT access control. *J. Parallel Distrib. Comput.* **2024**, *185*, 104795. [CrossRef]
14. Dhar, D.; Khare, A.; Dwivedi, A.D.; Singh, R. Securing IoT devices: A novel approach using blockchain and quantum cryptography. *Internet Things* **2024**, *25*, 101019. [CrossRef]
15. Hameed, K.; Barika, M.; Garg, S.; Amin, M.B.; Kang, B. A taxonomy study on securing blockchain-based industrial applications: An overview, application perspectives, requirements, attacks, countermeasures, and open issues. *J. Ind. Inf. Integr.* **2022**, *26*, 100312. [CrossRef]
16. Tseng, F.-M.; Liang, C.-W.; Nguyen, N.-B. Blockchain technology adoption and business performance in large enterprises: A comparison of the United States and China. *Technol. Soc.* **2023**, *73*, 102230. [CrossRef]

17. Zhu, X.; Liu, Y.; Cao, Y.; Jiao, Z. Demand response scheduling based on blockchain considering the priority of high load energy enterprises. *Energy Rep.* **2023**, *9*, 992–1000. [CrossRef]
18. Zhen, P.; Jiang, Z.; Wu, J.-J.; Zheng, Z. Blockchain-based decentralized application: A survey. *IEEE Open J. Comput. Soc.* **2024**, *4*, 121–133. [CrossRef]
19. Banoth, R.; Dave, M.B. A survey on decentralized application based on blockchain platform. In Proceedings of the International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 7–9 April 2022; pp. 1171–1174.
20. Tang, H.; Jiao, Y.; Huang, B.; Lin, C.; Goyal, S.; Wang, B. Learning to classify blockchain peers according to their behavior sequences. *IEEE Access* **2018**, *6*, 71208–71215. [CrossRef]
21. Buterin, V. On Public and Private Blockchains. Available online: https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains (accessed on 10 December 2023).
22. Xu, M.; Guo, Y.; Liu, C.; Hu, Q.; Yu, D.; Xiong, Z.; Niyato, D.; Cheng, X. Exploring blockchain technology through a modular lens: A survey. *arXiv* **2023**, arXiv:2304.08283v1. [CrossRef]
23. Oumaima, F.; Karim, Z.; Abdellatif, E.G.; Mohammed, B. A survey on blockchain and artificial intelligence technologies for enhancing security and privacy in smart environments. *IEEE Access* **2022**, *10*, 93168–93186.
24. Frankenfield, J. What Are Consensus Mechanisms in Blockchain and Cryptocurrency? Available online: https://www.investopedia.com/terms/c/consensus-mechanism-cryptocurrency.asp (accessed on 15 December 2023).
25. Li, J.; Gu, C.; Wei, F.; Chen, X. A survey on blockchain anomaly detection using data mining techniques. In Proceedings of the 1st International Conference on Blockchain and Trustworthy Systems (BlockSys 2019), Guangzhou, China, 7–8 December 2019; pp. 491–504.
26. Ul Hassan, M.; Rehmani, M.H.; Chen, J. Anomaly detection in blockchain networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 289–318. [CrossRef]
27. Hisham, S.; Makhtar, M.; Aziz, A.A. Combining Multiple Classifiers using Ensemble Method for Anomaly Detection in Blockchain Networks: A Comprehensive Review. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 404–422. [CrossRef]
28. Kamisalic, A.; Kramberger, R.; Fister, I.J. Synergy of blockchain technology and data mining techniques for anomaly detection. *Appl. Sci.* **2021**, *11*, 7987. [CrossRef]
29. Sachan, R.K.; Agarwal, R.; Shukla, S.K. Identifying malicious accounts in blockchains using domain names and associated temporal properties. *arXiv* **2021**, arXiv:2106.13420v1. [CrossRef]
30. Abu Musa, T.A.; Bouras, A. Anomaly detection: A survey. *Lect. Notes Netw. Syst.* **2022**, *217*, 391–401.
31. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *ACM Comput. Surv.* **2009**, *41*, 15. [CrossRef]
32. Pourhabibi, T.; Ong, K.-L.; Kam, B.H.; Boo, Y.L. Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decis. Support Syst.* **2020**, *133*, 113303. [CrossRef]
33. Morishima, S. Scalable anomaly detection method for blockchain transactions using GPU. In Proceedings of the 20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), Gold Coast, QLD, Australia, 5–7 December 2019; pp. 163–168.
34. Martin, K.; Rahouti, M.; Ayyash, M.; Alsmadi, I. Anomaly detection in blockchain using network representation and machine learning. *Secur. Priv.* **2022**, *5*, e192. [CrossRef]
35. Signorini, M.; Pontecorvi, M.; Kanoun, W.; Di Pietro, R. BAD: A blockchain anomaly detection solution. *IEEE Access* **2020**, *8*, 173481–173490. [CrossRef]
36. De Haro-Olmo, F.J.; Varela-Vaca, A.J.; Alvarez-Bermejo, J.A. Blockchain from the perspective of privacy and anonymization: A systematic literature review. *Sensors* **2020**, *20*, 7171. [CrossRef]
37. Jain, A.K.; Dubes, R.C. *Algorithms for Clustering Data*; Prentice-Hall Inc.: Upper Saddle River, NJ, USA, 1988.
38. Zhang, T.; Ramakrishnan, R.; Livny, M. BIRCH: A new data clustering algorithm and its applications. *Data Min. Knowl. Discov.* **1997**, *1*, 141–182. [CrossRef]
39. Qi, J.; Guo, Z.; Lu, Y.; Gao, J.; Guo, Y.; Fanyao, M. Security evaluation model of blockchain system based on combination weighting and grey clustering. In Proceedings of the 7th IEEE International Conference on Data Science in Cyberspace (DSC, 2022), Guilin, China, 11–13 July 2022; pp. 440–447.
40. Karypis, G.; Han, E.H.; Kumar, V. Chameleon: A hierarchical clustering algorithm using dynamic modeling. *IEEE Comput. Mag.* **1999**, *32*, 68–75. [CrossRef]
41. Scholkopf, B.; Williamson, R.; Smola, A.; Shawe-Taylor, J.; Platt, J. Support vector method for novelty detection. In Proceedings of the 12th International Conference on Neural Information Processing Systems, Denver, CO, USA, 29 November–4 December 1999; pp. 582–588.
42. Tax, D.M.J.; Duin, R.P.W. Support vector data description. *Mach. Learn.* **2004**, *54*, 45–66. [CrossRef]
43. Liu, F.T.; Ting, K.M.; Zhou, Z.H. Isolation-based anomaly detection. *ACM Trans. Knowl. Discov. Data* **2012**, *6*, 3. [CrossRef]
44. Pavithra, S.; Ramya, S.; Prathibha, S. A survey on cloud computing security issues and blockchain. In Proceedings of the 3rd International Conference on Computing and Communications Technologies (ICCCT), Chennai, India, 21–22 February 2019; pp. 136–140.
45. Hong, A.; Sun, C.; Chen, M. A survey of distributed database systems based on blockchain. In Proceedings of the 3rd International Conference on Smart BlockChain (SmartBlock), Zhengzhou, China, 23–25 October 2020; pp. 191–196.

46. Sadad, A.; Khan, M.A.; Ghaleb, B.; Khan, F.A.; Driss, M.; Boulila, W.; Ahmad, J. Distributed twins in edge computing: Blockchain and IOTA. *arXiv* **2023**, arXiv:2305.07453v1.
47. Sadri, H.; Yitmen, I.; Tagliabue, L.C.; Westphal, F.; Tezel, A.; Taheri, A.; Sibenik, G. Integration of blockchain and digital twins in the smart built environment adopting disruptive technologies—A systematic review. *Sustainability* **2023**, *15*, 3713. [CrossRef]
48. Malibari, N.A. A survey on blockchain-based applications in education. In Proceedings of the 7th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 12–14 March 2020; pp. 266–270.
49. Al-Maaitah, S.; Qatawneh, M.; Quzmar, A. E-voting system based on blockchain technology: A survey. In Proceedings of the International Conference on Information Technology (ICIT), Amman, Jordan, 14–15 July 2021; pp. 200–205.
50. Ren, K.; Ho, N.-M.; Loghin, D.; Nguyen, T.-T.; Ooi, B.C.; Ta, Q.T.; Zhu, F. Interoperability in blockchain: A survey. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 12750–12769. [CrossRef]
51. Qian, P.; Liu, Z.; He, Q.; Huang, B.; Tian, D.; Wang, X. Smart contract vulnerability detection technique: A survey. *arXiv* **2022**, arXiv:2209.05872v1.
52. Ivanov, N.; Li, C.; Yan, Q.; Sun, Z.; Cao, Z.; Luo, X. Security defense for smart contracts: A comprehensive survey. *arXiv* **2023**, arXiv:2302.07347v3.
53. Meisami, S.; Bodell III, W.E. A comprehensive survey of upgradeable smart contract patterns. *arXiv* **2023**, arXiv:2304.03405.
54. Cho, S.; Lee, S. Survey on the application of blockchain to IoT: Research trend for applying blockchain to IoT. In Proceedings of the International Conference on Electronics, Information, and Communication (ICEIC), Auckland, New Zealand, 22–25 January 2019; pp. 1–2.
55. Shammar, E.A.; Zahary, A.T.; Al-Shargabi, A.A. A Survey of IoT and blockchain integration: Security perspective. *IEEE Access* **2021**, *9*, 156114–156150. [CrossRef]
56. Qatawneh, M. Use of blockchain in the Internet of Things: A survey. *arXiv* **2023**, arXiv:2303.06035.
57. Xue, H.; Chen, D.; Zhang, N.; Dai, H.-N.; Yu, K. Integration of blockchain and edge computing in Internet of Things: A survey. *arXiv* **2022**, arXiv:2205.13160v1. [CrossRef]
58. Dai, H.-N.; Zheng, Z.; Zhang, Y. Blockchain for internet of Things: A survey. *arXiv* **2020**, arXiv:1906.00245v5. [CrossRef]
59. Khan, Z.A.; Namin, A.S. A survey on the applications of blockchains in security of IoT systems. *arXiv* **2021**, arXiv:2112.09296v1.
60. Jiang, Y.; Ma, B.; Wang, X.; Yu, P.; Yu, G.; Wang, Z.; Ni, W.; Liu, R.P. Blockchained federated learning for Internet of Things: A comprehensive survey. *arXiv* **2023**, arXiv:2305.04513v1. [CrossRef]
61. Conti, M.; Kumar, E.S.; Lal, C.; Ruj, S. A survey on security and privacy issues of Bitcoin. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3416–3452. [CrossRef]
62. Zhang, R.; Xue, R.; Liu, L. Security and privacy on blockchain. *arXiv* **2019**, arXiv:1903.07602v2. [CrossRef]
63. Zhang, R.; Xue, R.; Liu, L. Security and privacy for healthcare blockchains. *arXiv* **2021**, arXiv:2106.06136v1. [CrossRef]
64. Manimurgan, S.; Anitha, T.; Divya, G.; Charlyn Pushpa Latha, G.; Mathupriya, S. A survey on blockchain technology for network security applications. In Proceedings of the 2nd International Conference on Computing and Information Technology (ICCIT), Tabuk, Saudi Arabia, 25–27 January 2022; pp. 440–445.
65. Kumar, A.; Sharma, I. Enhancing cybersecurity policies with blockchain technology: A survey. In Proceedings of the 5th International Conference on Contemporary Computing and Informatics (IC3I), Uttar Pradesh, India, 14–16 December 2022; pp. 1050–1054.
66. Salman, T.; Zolanvari, M.; Erbad, A.; Jain, R.; Samaka, M. Security services using blockchains: A state of the art survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 858–880. [CrossRef]
67. Yuan, G.; Feng, L.; Ning, J.; Yang, X. Survey on the application of blockchain in digital rights protection. In Proceedings of the International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI), Sanya, China, 4–6 December 2020; pp. 183–187.
68. Zhu, L.; Zheng, B.; Shen, M.; Gao, F.; Li, H.; Shi, K. Research on the security of blockchain data: A survey. *arXiv* **2018**, arXiv:1812.02009v2.
69. Li, X.; Jiang, P.; Chen, T.; Luo, X.; Wen, Q. A survey on the security of blockchain systems. *arXiv* **2020**, arXiv:1802.06993v3. [CrossRef]
70. Rai, G.S.; Goyal, S.B.; Chatterjee, P. Anomaly detection in blockchain using machine learning. *Lect. Notes Electr. Eng.* **2023**, *984*, 487–499.
71. Lashkari, B.; Musilek, P. A comprehensive review of blockchain consensus mechanisms. *IEEE Access* **2021**, *9*, 43620–43652. [CrossRef]
72. Sultan, K.; Ruhi, U.; Lakhani, R. Conceptualizing blockchains: Characteristics and applications. In Proceedings of the 11th IADIS International Conference on Information Systems, Lisbon, Portugal, 14–16 April 2018; pp. 49–57.
73. Parizi, R.M.; Dehghantanha, A.; Raymond Choo, K.-K.; Singh, A. Empirical vulnerability analysis of automated smart contracts security testing on blockchains. In Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering (CASCON '18), Markham, ON, Canada, 29–31 October 2018; pp. 103–113.
74. Kosba, A.; Miller, A.; Shi, E.; Wen, Z.; Papamanthou, C. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 839–858.

75. Panigrahi, A.; Nayak, A.K.; Paul, R. Impact of clustering technique in enhancing the blockchain network performance. In Proceedings of the 2022 International Conference on Machine Learning, Computer Systems and Security (MLCSS), Bhubaneswar, India, 5–6 August 2022; pp. 363–367.
76. Joshi, P.; Kumar, S.; Kumar, D.; Singh, A.K. A blockchain based framework for fraud detection. In Proceedings of the 2019 Conference on Next Generation Computing Applications (NextComp), Balaclava, Mauritius, 19–21 September 2019.
77. Ma, J.; Lin, S.Y.; Chen, X.; Sun, H.-M.; Chen, Y.-C.; Wang, H. A blockchain-based application system for product anti-counterfeiting. *IEEE Access* **2020**, *8*, 77642–77652. [CrossRef]
78. Reid, F.; Harrigan, M. An analysis of anonymity in the bitcoin system. In *Security and Privacy in Social Networks*; Altshuler, Y., Elovici, Y., Cremers, A.B., Aharony, N., Pentland, A., Eds.; Springer: New York, NY, USA, 2013; pp. 197–222.
79. Zhang, Y.; Wang, J.; Luo, J. Heuristic-based address clustering in bitcoin. *IEEE Access* **2020**, *8*, 210582–210591. [CrossRef]
80. Ferrag, M.A.; Derdour, M.; Mukherjee, M.; Derhab, A.; Maglaras, L.; Janicke, H. Blockchain technologies for the Internet of Things: Research issues and challenges. *IEEE Internet Things J.* **2018**, *6*, 2188–2204. [CrossRef]
81. Crosby, M.; Nachiappan Pattanayak, P.; Verma, S.; Kalyanaraman, V. Blockchain technology: Beyond Bitcoin. *Appl. Innov.* **2016**, *2*, 6–10.
82. Zapotochnyi, A. What Are Smart Contracts? Available online: https://blockgeeks.com/guides/smart-contracts (accessed on 5 March 2024).
83. 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]
84. Manolache, M.A.; Manolache, S.; Tapus, N. Decision making using the blockchain proof of authority consensus. *Procedia Comput. Sci.* **2022**, *199*, 580–588. [CrossRef]
85. Alrubei, S.; Ball, E.; Rigelsford, J. HDPoA: Honesty-based distributed proof of authority via scalable work consensus protocol for IoT-blockchain applications. *Comput. Netw.* **2022**, *217*, 109337. [CrossRef]
86. Dash, B. Zero-trust architecture (ZTA): Designing an AI-powered cloud security framework for LLMs' black box problems. *Curr. Trends Eng. Sci. (CTES)* **2024**, *4*, 1058. [CrossRef]
87. Wu, W.; Liu, E.; Gong, X.; Wang, R. Blockchain based zero-knowledge proof of location in IoT. In Proceedings of the International IEEE Conference on Communications (ICC' 20), Dublin, Ireland, 7–11 June 2020; pp. 1–7.
88. Xu, G.; Liu, Y.; Khan, P.W. Improvement of the dpos consensus mechanism in blockchain based on vague sets. *IEEE Trans. Ind. Inform.* **2020**, *16*, 4252–4259. [CrossRef]
89. Ul Hassan, M.U.; Rehmani, M.H.; Chen, J. Deal: Differentially private auction for blockchain-based microgrids energy trading. *IEEE Trans. Serv. Comput.* **2020**, *13*, 263–275. [CrossRef]
90. Heilman, E.; Kendler, A.; Zohar, A.; Goldberg, S. Eclipse attacks on Bitcoin's peer-to-peer network. In Proceedings of the 24th USENIX Security Symposium, Washington, DC, USA, 12–14 August 2015; pp. 129–144.
91. Alangot, B.; Reijsbergen, D.; Venugopalan, S.; Szalachowski, P.; Yeo, K.S. Decentralized and lightweight approach to detect Eclipse attacks on Proof of Work blockchains. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 1659–1672. [CrossRef]
92. Rahouti, M.; Xiong, K.; Ghani, N. Bitcoin concepts, threats, and machine-learning security solutions. *IEEE Access* **2018**, *6*, 67189–67205. [CrossRef]
93. Saad, M.; Thai, M.T.; Mohaisen, A. POSTER: Deterring DDoS attacks on blockchain-based cryptocurrencies through Mempool optimization. In Proceedings of the Asia Conference on Computer and Communications Security (ASIACCS '18), Incheon, Republic of Korea, 4 June 2018; pp. 809–811.
94. Bano, S.; Sonnino, A.; Al-Bassam, M.; Azouvi, S.; McCorry, P.; Meiklejohn, S.; Danezis, G. SoK: Consensus in the age of blockchains. In Proceedings of the 1st ACM Conference on Advances in Financial Technologies, Zurich, Switzerland, 21–23 October 2019; pp. 183–198.
95. CipherTrace. Available online: https://ciphertrace.com/ (accessed on 10 April 2024).
96. Cybersecurity Ventures. Ransomware Damage Costs Predicted to Reach $265 Billion by 2031. Available online: https://cybersecurityventures.com/global-ransomware-damage-costs-predicted-to-reach-250-billion-usd-by-2031/ (accessed on 15 April 2024).
97. Zohar, A. Bitcoin: Under the hood. *Commun. ACM* **2015**, *58*, 104–113. [CrossRef]
98. Eyal, I.; Sirer, E.G. Majority is not enough: Bitcoin mining is vulnerable. *arXiv* **2013**, arXiv:1311.0243v5. [CrossRef]
99. Gomez, W. What Is a Finney Hack or Finney Attack? Available online: https://academy.bit2me.com/en/which-is-a-hack-finney-attack-finney/ (accessed on 18 April 2024).
100. Meiklejohn, S.; Pomarole, M.; Jordan, G.; Levchenko, K.; McCoy, D.; Voelker, G.M.; Savage, S. A fistful of bitcoins: Characterizing payments among men with no names. *Commun. ACM* **2016**, *59*, 86–93. [CrossRef]
101. Memoria, F. 700 Million Stuck in 115,000 Unconfirmed Bitcoin Transactions. Available online: https://www.ccn.com/700-million-stuck-115000-unconfirmed-bitcoin-transactions (accessed on 4 February 2024).
102. Ekparinya, P.; Gramoli, V.; Jourjon, G. Impact of Man-in-the-Middle Attacks on Ethereum. In Proceedings of the 37th IEEE Symposium on Reliable Distributed Systems (SRDS), Salvador, Brazil, 2–5 October 2018; pp. 11–20.
103. Kang, C.; Lee, C.; Ko, K.; Woo, J.; Hong, J.W.-K. De-anonymization of the Bitcoin network using address clustering. *Commun. Comput. Inf. Sci.* **2020**, *1267*, 489–501.

104. Goldstein, M.; Uchida, S. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLoS ONE* **2016**, *11*, e0152173. [CrossRef]
105. Sayadi, S.; Rejeb, B.; Choukair, Z. Anomaly detection model over blockchain electronic transactions. In Proceedings of the 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; pp. 895–900.
106. Mirsky, Y.; Golomb, T.; Elovici, Y. Lightweight collaborative anomaly detection for the IoT using blockchain. *J. Parallel Distrib. Comput.* **2020**, *145*, 75–97. [CrossRef]
107. Kim, J.; Nakashima, M.; Fan, W.; Wuthier, S.; Zhou, X.; Kim, I.; Chang, S.-Y. A machine learning approach to anomaly detection based on traffic monitoring for secure blockchain networking. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 3619–3632. [CrossRef]
108. Patel, V.; Pan, L.; Rajasegarar, S. Graph deep learning based anomaly detection in Ethereum blockchain network. *Lect. Notes Comput. Sci.* **2020**, *12570*, 132–148.
109. 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]
110. Guo, C.; Zhang, S.; Zhang, P.; Alkubati, M.; Song, J. LB-GLAT: Long-term bi-graph layer attention convolutional network for anti-money laundering in transactional blockchain. *Mathematics* **2023**, *11*, 3927. [CrossRef]
111. Frey, B.J.; Dueck, D. Clustering by passing messages between data points. *Science* **2007**, *315*, 972–976. [CrossRef] [PubMed]
112. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.
113. Ng, A.Y.; Jordan, M.I.; Weiss, Y. On spectral clustering: Analysis and an algorithm. In Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic (NIPS'01), Vancouver, BC, Canada, 3–8 December 2001; pp. 849–856.
114. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *10*, P10008. [CrossRef]
115. Breunig, M.M.; Kriegel, H.-P.; Ng, R.T.; Sander, J. LOF: Identifying density-based local outliers. *ACM SIGMOD Rec.* **2000**, *29*, 93–104. [CrossRef]
116. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96), Portland, OR, USA, 2–4 August 1996; pp. 226–231.
117. Campello, R.J.G.B.; Moulavi, D.; Zimek, A.; Sander, J. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Trans. Knowl. Discov. Data* **2015**, *10*, 5. [CrossRef]
118. Kipf, T.N.; Welling, M. Variational graph auto-encoders. *arXiv* **2016**, arXiv:1611.07308v1.
119. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *arXiv* **2014**, arXiv:1406.2661.
120. Liu, F.T.; Ting, K.M.; Zhou, Z.-H. Isolation forest. In Proceedings of the 8th IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 413–422.
121. Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
122. Hasan, M.; Rahman, M.S.; Janicke, H.; Sarker, I.H. Detecting anomalies in blockchain transactions using machine learning classifiers and explainability analysis. *arXiv* **2024**, arXiv:2401.03530.
123. Hojjati, H.; Ho, T.; Armanfard, N. Self-Supervised anomaly detection: A survey and outlook. *arXiv* **2022**, arXiv:2205.05173.
124. Kinkeldey, C.; Fekete, J.-D.; Isenberg, P. BitConduite: Visualizing and analyzing activity on the Bitcoin network. In Proceedings of the Eurographics Conference on Visualization (EuroVis' 17), Barcelona, Spain, 12–16 June 2017; pp. 25–27.
125. Khenfouci, Y.; Challal, Y.; Hamdad, L. ClusterChain: Decentralized and trustworthy clustering over blockchain. In Proceedings of the International Conference on Networking and Advanced Systems (ICNAS), Annaba, Algeria, 27–28 October 2016; pp. 1–6.
126. Mongo Database. Available online: https://www.mongodb.com/ (accessed on 5 February 2024).
127. Monamo, P.; Marivate, V.; Twala, B. Unsupervised learning for robust Bitcoin fraud detection. In Proceedings of the Information Security for South Africa (ISSA) Conference, Johannesburg, South Africa, 17–18 August 2016; pp. 129–134.
128. Deepa, M.; Akila, D. Cost-effective anomaly detection for blockchain transactions using unsupervised learning. *Lect. Notes Netw. Syst.* **2021**, *248*, 445–453.
129. Li, L.; Noorian, F.; Moss, D.J.; Leong, P.H. Rolling window time series prediction using MapReduce. In Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014), Redwood City, CA, USA, 13–15 August 2014; pp. 757–764.
130. Podgorelec, B.; Turkanovic, M.; Karakatic, S. A machine learning-based method for automated blockchain transaction signing including personalized anomaly detection. *Sensors* **2020**, *20*, 147. [CrossRef]
131. Chang, T.-H.; Svetinovic, D. Improving Bitcoin ownership identification using transaction patterns analysis. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 9–20. [CrossRef]
132. Epishkina, A.; Zapechnikov, S. Discovering and clustering hidden time patterns in blockchain ledger. In *Biologically Inspired Cognitive Architectures (BICA) for Young Scientists*; Samsonovich, A.V., Klimov, V.V., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 245–250.

133. Huang, B.; Liu, Z.; Chen, J.; Liu, A.; Liu, Q.; He, Q. Behavior pattern clustering in blockchain networks. *Multimed. Tools Appl.* **2017**, *76*, 20099–20110. [CrossRef]
134. Kumari, R.; Catherine, M. Anomaly detection in blockchain using clustering protocol. *Int. J. Pure Appl. Math.* **2018**, *118*, 391–396.
135. Norvill, R.; State, R.; Awan, I.; Fiz Pontiveros, B.B.; Cullen, A. Automated labeling of unknown contracts in Ethereum. In Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Barcelona, Spain, 31 July–3 August 2017; pp. 1165–1172.
136. Schubert, E.; Rousseeuw, P.J. Fast and eager *k*-medoids clustering: $O(k)$ runtime improvement of the PAM, CLARA, and CLARANS algorithms. *Inf. Syst.* **2021**, *101*, 101804. [CrossRef]
137. Tsoulias, K.; Palaiokrassas, G.; Fragkos, G.; Litke, A.; Varvarigou, T.A. A graph model based blockchain implementation for increasing performance and security in decentralized ledger systems. *IEEE Access* **2020**, *8*, 130952–130965. [CrossRef]
138. Zambre, D.; Shah, A. Analysis of Bitcoin Network Dataset for Fraud. Stanford CS 224W Project Final Report 2013. Available online: https://snap.stanford.edu/class/cs224w-2013/projects2013/cs224w-030-final.pdf (accessed on 12 December 2023).
139. Turner, A.B.; McCombie, S.; Uhlmann, A.J. Follow the money: Revealing risky nodes in a ransomware-bitcoin network. In Proceedings of the 54th Hawaii International Conference on System Sciences, Maui, HI, USA, 5–8 January 2021; pp. 1560–1572.
140. Khandelwal, N. How the Graph Is Changing the Way We Access Blockchain Data. Available online: https://medium.com/@navanshkhandelwal14/how-the-graph-is-changing-the-way-we-access-blockchain-data-c197334cd63e (accessed on 29 February 2024).
141. Mc Ginn, D.; Birch, D.; Akroyd, D.; Molina-Solana, M.; Guo, Y.; Knottenbelt, W. Visualizing dynamic Bitcoin transaction patterns. *Big Data* **2016**, *4*, 109–119. [CrossRef] [PubMed]
142. Yang, C.; Chin, K.-W.; Wang, J.; Wang, X.; Liu, Y.; Zheng, Z. Scaling blockchains with error correction codes: A survey on coded blockchains. *arXiv* **2022**, arXiv:2208.09255v1. [CrossRef]
143. Chaudhari, D.; Agarwal, R.; Shukla, S.K. Towards malicious address identification in Bitcoin. In Proceedings of the 2021 IEEE International Conference on Blockchain (Blockchain), Melbourne, VIC, Australia, 6–8 December 2021; pp. 425–432.
144. Zheng, B.; Zhu, L.; Shen, M.; Du, X.; Yang, J.; Gao, F.; Li, Y.; Zhang, C.; Liu, S.; Yin, S. Malicious Bitcoin transaction tracing using incidence relation clustering. In Proceedings of the International Conference on Mobile Networks and Management (MONAMI), Melbourne, VIC, Australia, 13–15 December 2017; pp. 313–323.
145. Swaroopa, R.B.; Sharma, G.V. UL-blockDAG: Unsupervised learning based consensus protocol for blockchain. In Proceedings of the 40th International Conference on Distributed Computing Systems (ICDCS' 20), Singapore, 29 November–1 December 2020; pp. 1243–1248.
146. Biryukov, A.; Tikhomirov, S. Transaction clustering using network traffic analysis for bitcoin and derived blockchains. In Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 9 April-May 2019; pp. 204–209.
147. Pustogarov, I. Bitcoin Network Probing Tool. Available online: https://github.com/ivanpustogarov/bcclient (accessed on 15 December 2023).
148. Diaz, C.; Seys, S.; Claessens, J.; Preneel, B. Towards measuring anonymity. *Lect. Notes Comput. Sci.* **2002**, *2482*, 54–68.
149. Etherscan. Available online: https://etherscan.io/ (accessed on 3 February 2024).
150. Magnusson, M. Discovering hidden time patterns in behavior: T-patterns and their detection. *Behav. Res. Methods Instrum. Comput.* **2000**, *32*, 93–110. [CrossRef] [PubMed]
151. Ward, J.H. Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.* **1963**, *58*, 236–244. [CrossRef]
152. Shi, J.; Ye, L.; Li, Z.; Zhan, D. Unsupervised binary protocol clustering based on maximum sequential patterns. *CMES-Comput. Model. Eng. Sci.* **2022**, *130*, 495–510. [CrossRef]
153. Arthur, D.; Vassilvitskii, S. k-Means++ The Advantages of Careful Seeding. Technical Report. Stanford. 2016. Available online: https://theory.stanford.edu/~sergei/papers/kMeansPP-soda.pdf (accessed on 16 December 2023).
154. Wang, J.; Han, J. Bide: Efficient mining of frequent closed sequences. In Proceedings of the 20th International Conference on Data Engineering, Boston, MA, USA, 2 April 2004; pp. 79–90.
155. Fowlkes, E.B.; Mallows, C.L. A method for comparing two hierarchical clusterings. *J. Am. Stat. Assoc.* **1983**, *78*, 553–569. [CrossRef]
156. Pham, T.; Lee, S. Anomaly detection in Bitcoin network using unsupervised learning methods. *arXiv* **2017**, arXiv:1611.03941v2.
157. Pham, T.; Lee, S. Anomaly detection in the Bitcoin system—A network perspective. *arXiv* **2017**, arXiv:1611.03942.
158. Tsolakis, D.; Tsekouras, G.E.; Niros, A.D.; Rigos, A. On the systematic development of fast fuzzy vector quantization for grayscale image compression. *Neural Netw.* **2012**, *36*, 83–96. [CrossRef] [PubMed]
159. Cuesta-Albertos, J.A.; Gordaliza, A.; Matran, C. Trimmed k-means: An attempt to robustify quantizers. *Ann. Stat.* **1997**, *25*, 553–576. [CrossRef]
160. Shayegan, M.J.; Sabor, H.R.; Uddin, M.; Chen, C.-L. A collective anomaly detection technique to detect crypto wallet frauds on Bitcoin network. *Symmetry* **2022**, *14*, 328. [CrossRef]
161. Kondor, D.; Posfai, M.; Csabai, I.; Vattay, G. Do the rich get richer? An empirical analysis of the Bitcoin transaction network. *PLoS ONE* **2014**, *9*, e86197. [CrossRef] [PubMed]
162. Kampers, O.; Qahtan, A.; Mathur, S.; Velegrakis, Y. Manipulation detection in cryptocurrency markets: An anomaly and change detection based approach. In Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing (SAC '22), Virtual Event, 25–29 April 2022; pp. 326–329.

163. Qahtan, A.; Zhang, X.; Wang, S. Efficient estimation of dynamic density functions with an application to outlier detection. In Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM 12), Maui, HI, USA, 29 October–2 November 2012; pp. 2159–2163.
164. Henderson, K.; Gallagher, B.; Eliassi-Rad, T.; Tong, H.; Basu, S.; Akoglu, L.; Koutra, D.; Faloutsos, C.; Li, L. RolX: Structural role extraction and mining in large graphs. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12), Beijing China, 12–16 August 2012; pp. 1231–1239.
165. Hirshman, Y.; Huang, S.; Macke, S. Unsupervised Approaches to Detecting Anomalous Behavior in the Bitcoin Transaction Network. Technical Report, Stanford University 2013, cs229.stanford.edu. Available online: https://cs229.stanford.edu/proj2013/ (accessed on 15 January 2024).
166. Wallet Explorer. Available online: https://www.walletexplorer.com (accessed on 3 March 2024).
167. Shah, R.S.; Bhatia, A.; Gandhi, A.; Mathur, S. Bitcoin data analytics: Scalable techniques for transaction clustering and embedding generation. In Proceedings of the International Conference on Communication Systems & Networks (COMSNETS '21), Bangalore, India, 5–9 January 2021; pp. 1–8.
168. Frost, N.; Moshkovitz, M.; Rashtchian, C. Exkmc: Expanding explainable k-means clustering. *arXiv* **2020**, arXiv:2006.02399v2.
169. Blockchain Charts. Available online: https://www.blockchain.com/explorer/charts (accessed on 2 March 2024).
170. Saravanan, R.; Sreeparvathy, V.S.; Santhiya, S.; Shalini, K. Comparative study analysis of machine learning algorithms for anomaly detection in blockchain. In Proceedings of the International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE '23), Ballar, India, 29–30 April 2023; pp. 1–6.
171. Sun, H.; Ruan, N.; Liu, H. Ethereum analysis via node clustering. *Lect. Notes Comput. Sci.* **2019**, *11928*, 114–129.
172. Zhang, X.; Li, G.; Wang, Y. GAN-based abnormal transaction detection in Bitcoin. In Proceedings of the 7th IEEE International Conference on Smart Cloud (SmartCloud), Shanghai, China, 8–10 October 2022; pp. 157–162.
173. Agarwal, R.; Thapliyal, T.; Shukla, S.K. Vulnerability and transaction behavior based detection of malicious smart contracts. *Lect. Notes Comput. Sci.* **2022**, *13172*, 79–96.
174. Dingman, W.; Cohen, A.; Ferrara, N.; Lynch, A.; Jasinski, P.; Black, P.E.; Deng, L. Defects and vulnerabilities in smart contracts, a classification using the NIST bugs framework. *Int. J. Networked Distrib. Comput.* **2019**, *7*, 121–132. [CrossRef]
175. Agarwal, R.; Kumar, A.; Singh, A.K. Detecting malicious accounts in permissionless blockchains using temporal graph properties. *Appl. Netw. Sci.* **2021**, *6*, 9. [CrossRef]
176. Baek, H.; Oh, J.; Kim, C.Y.; Lee, K. A model for detecting cryptocurrency transactions with discernible purpose. In Proceedings of the 11th International Conference on Ubiquitous and Future Networks (ICUFN), Zagreb, Croatia, 2–5 July 2019; pp. 713–717.
177. Binance. Available online: https://www.binance.com/ (accessed on 4 October 2023).
178. Bartoletti, M.; Carta, S.; Cimoli, T.; Saia, R. Dissecting Ponzi schemes on Ethereum: Identification, analysis, and impact. *Future Gener. Comput. Syst.* **2020**, *102*, 259–277. [CrossRef]
179. Bartoletti, M.; Pes, B.; Serusi, S. Data mining for detecting Bitcoin Ponzi schemes. In Proceedings of the Crypto Valley Conference on Blockchain Technology (CVCBT), Zug, Switzerland, 20–22 June 2018; pp. 75–84.
180. Reddit. Available online: https://www.reddit.com/ (accessed on 12 January 2024).
181. Bitcointalk. Available online: https://bitcointalk.org/ (accessed on 12 January 2024).
182. Elliptic Data Set. Available online: https://www.kaggle.com/ellipticco/elliptic-data-set (accessed on 1 November 2023).
183. 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 1st International Conference of Smart Systems and Emerging Technologies (SMARTTECH), Riyadh, Saudi Arabia, 3–5 November 2020; pp. 110–115.
184. Rabieinejad, E.; Yazdinejad, A.; Parizi, R.M.; Dehghantanha, A. Generative adversarial networks for cyber threat hunting in Ethereum blockchain. *Distrib. Ledger Technol. Res. Pract.* **2023**, *2*, 1–19. [CrossRef]
185. Lorenz, J.; Silva, M.I.; Aparicio, D.; Ascensao, J.T.; Bizarro, P. Machine learning methods to detect money laundering in the Bitcoin blockchain in the presence of label scarcity. *arxiv* **2020**, arXiv:2005.14635.
186. Weber, M.; Domeniconi, G.; Chen, J.; Weidele, D.K.I.; Bellei, B.; Robinson, T.; Leiserson, C.E. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv* **2019**, arXiv:1908.02591.
187. Settles, B. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison. 2009. Available online: https://minds.wisconsin.edu/handle/1793/60660 (accessed on 25 November 2023).
188. Farrugia, S.; Ellul, J.; Azzopardi, G. Detection of illicit accounts over the Ethereum blockchain. *Expert Syst. Appl.* **2020**, *150*, 113318. [CrossRef]
189. Chen, T.; Zhu, Y.; Li, Z.; Chen, J.; Li, X.; Luo, X.; Lin, X.; Zhange, X. Understanding Ethereum via graph analysis. In Proceedings of the IEEE Conference on Computer Communications (IEEE INFOCOM '18), Honolulu, HI, USA, 16–19 April 2018; pp. 1484–1492.
190. Sachan, R.K.; Agarwal, R.; Shukla, S.K. DNS based in-browser cryptojacking detection. *arXiv* **2022**, arXiv:2205.04685v1.
191. Agarwal, R.; Thapliyal, T.; Shukla, S. Analyzing malicious activities and detecting adversarial behavior in cryptocurrency based permissionless blockchains: An Ethereum usecase. *Distrib. Ledger Technol. Res. Pract.* **2022**, *1*, 8. [CrossRef]
192. Agarwal, R.; Thapliyal, T.; Shukla, S. Detecting malicious accounts showing adversarial behavior in permissionless blockchains. *arXiv* **2021**, arXiv:2101.11915v.

193. Kumar, K.; Bhushan, B. Augmenting cybersecurity and fraud detection using artificial intelligence advancements. In Proceedings of the 4th International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 3–4 November 2023; pp. 1207–1212.
194. Gad, A.G.; Mosa, D.T.; Abualigah, L.; Abohany, A.A. Emerging trends in blockchain technology and applications: A review and outlook. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 6719–6742. [CrossRef]
195. Jha, R.K. Challenges of effective decision making in decentralized autonomous organizations (DAOs). *World J. Res. Rev.* **2023**, *17*, 18–25.
196. Buck, C.; Olenberger, C.; Schweizer, A.; Volter, F.; Eymann, T. Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust. *Comput. Secur.* **2021**, *110*, 102436. [CrossRef]
197. Lu, L.; Han, J.; Liu, Y.; Hu, L.; Huai, J.-P.; Ni, L.; Ma, J. Pseudo Trust: Zero-knowledge authentication in anonymous P2Ps. *IEEE Trans. Parallel Distrib. Syst.* **2008**, *19*, 1325–1337. [CrossRef]
198. Arazzi, M.; Nicolazzo, S.; Nocera, A. A fully privacy-preserving solution for anomaly detection in IoT using federated learning and homomorphic encryption. *Inf. Syst. Front.* **2023**. [CrossRef]
199. Bernabe, J.B.; Canovas, J.L.; Hernandez-Ramos, J.L.; Moreno, R.T.; Skarmeta, A. Privacy-preserving solutions for blockchain: Review and challenges. *IEEE Access* **2019**, *7*, 164908. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.