Machine Learning for Blockchain Attack Detection: Methods, Challenges and Future Directions

Zhexi Zhang

Electrical and Electronic Engineering, University of Liverpool, Liverpool, United Kingdom

sgzzh117@liverpool.ac.uk

Abstract. Blockchain systems, despite their decentralised and secure design, remain vulnerable to a range of attacks, including Sybil attacks, 51% attacks, and smart contract exploits. In response, researchers have increasingly adopted Machine Learning (ML) to detect and mitigate these threats. This review provides a comprehensive overview of ML-based approaches for blockchain attack detection. This paper first categories existing works into traditional ML and Deep Learning (DL) methods. Traditional models such as Decision Trees, SVMS, and Logistic Regression offer efficiency and interpretability but are limited in handling complex, sequential data. In contrast, DL approaches—including Convolutional Neural Networks (CNNs), Long Short-term Memory Networks (LSTMs), and Graph Neural Networks (GNNs)—demonstrate powerful pattern recognition capabilities and adaptability to diverse attack types. However, they often suffer from poor interpretability, limited generalisation across blockchain platforms, and heavy reliance on labelled data. This paper also identifies key challenges, including data imbalance, high computational cost, and reproducibility issues. To address these, this paper discusses future directions such as integrating expert systems, employing domain adaptation, and using generative models for data augmentation and pre-training. Overall, this review highlights the opportunities and limitations of ML in blockchain security and provides a roadmap for developing more robust and scalable detection frameworks.

INTRODUCTION

Blockchain technology has transformed the design of digital systems, particularly in establishing trust and ensuring data integrity without reliance on centralised authorities. Originally developed for cryptocurrencies like Bitcoin, blockchain has since been adopted in numerous sectors, such as healthcare, finance, supply chain, and the Internet of Things (IoT), owing to its decentralisation, immutability, and transparency. However, despite its reputation for security, blockchain systems are not immune to attacks. Malicious actors have exploited vulnerabilities through various methods, including 51% attacks, double-spending, Sybil attacks, and smart contract exploits [1]. These attacks not only compromise the integrity of blockchain networks but also pose significant challenges to their broader adoption in critical applications.

Recent studies have proposed using Machine Learning (ML) to enhance blockchain security. ML algorithms, recognised for their ability to detect patterns, identify anomalies, and learn from data, are widely employed in traditional cybersecurity applications. When applied to blockchain, ML can assist in detecting malicious behaviours, classifying attack types, and identifying anomalies in transaction or consensus patterns [2]. The key benefit of implementing ML in blockchain security is its potential to detect both known and unknown threats by learning from evolving data, rendering it more adaptive than rule-based systems.

A growing body of literature has explored the intersection between blockchain and ML. One notable example is Block Hunter, a federated learning framework designed for threat hunting in blockchain-based Industrial IoT (IIoT) environments [3]. This approach enables decentralised nodes to collaboratively train a global ML model while maintaining data privacy. It enhances attack detection accuracy and aligns with the decentralised philosophy of blockchain. Similarly, Sayeed and Marco proposed a model that integrates machine learning with algorithmic game theory to mitigate majority (51%) attacks by analysing consensus behaviour [4]. Their work demonstrates how predictive models can detect abnormal actions that may indicate the onset of a coordinated attack.

On the other hand, researchers have also shown that ML can be employed offensively to exploit blockchain systems. Wu et al. introduced a cascading ML framework that de-anonymises Bitcoin transactions, raising concerns about the privacy guarantees of blockchain [2]. Meanwhile, Khan et al. developed a blockchain-based architecture that integrates ML models for IoT-based e-health applications, concentrating on securing the integrity of ML training data and protecting the model from adversarial manipulation [5]. These works collectively underscore both the defensive and offensive roles that ML can play in the blockchain ecosystem.

Benefiting from the rapid development of this field, numerous studies have made significant breakthroughs in recent years. Therefore, this review aims to systematically summarize, investigate, and analyze these advancements, offering a comprehensive understanding of the current research landscape. This research aims to examine these challenges by evaluating current ML-based solutions for blockchain attack detection, identifying their limitations, and summarising potential solutions. By bridging the divide between ML techniques and blockchain-specific threats, this review contributes to the development of intelligent and resilient blockchain infrastructures capable of defending against evolving cyber threats.

METHOD

Preliminaries of Blockchain and Machine Learning

Blockchain is a decentralized and tamper-resistant digital ledger that records transactions across a network of nodes, eliminating the need for a central authority. Each block includes a list of transactions, a timestamp, and a cryptographic hash of the previous block, ensuring immutability and traceability. Common consensus mechanisms, such as Proof of Work (PoW) and Proof of Stake (PoS), allow network participants to agree on the validity of transactions and the state of the ledger [1]. While blockchain is often viewed as secure by design, it remains susceptible to various attack vectors, including 51% attacks, double-spending, Sybil attacks, and smart contract exploits [6].

ML, a subfield of artificial intelligence, involves developing algorithms that learn patterns from data and make predictions or decisions without explicit programming. ML techniques have increasingly been applied to enhance blockchain security, particularly in detecting anomalies, malicious behaviors, and fraudulent transactions. Supervised, unsupervised, and reinforcement learning algorithms are commonly used depending on the nature of the security problem and the availability of labeled data [7].

The general workflow for integrating ML into blockchain security, shown in Figure 1, comprises several stages: data acquisition, feature engineering, model training, validation, and deployment. Blockchain data, including transaction histories, node activities, and smart contract logs, serves as input. This data is preprocessed and transformed into feature sets that are suitable for learning algorithms. Once trained and validated, ML models can monitor blockchain activity in real-time, identify suspicious patterns, and flag potential security threats [8].

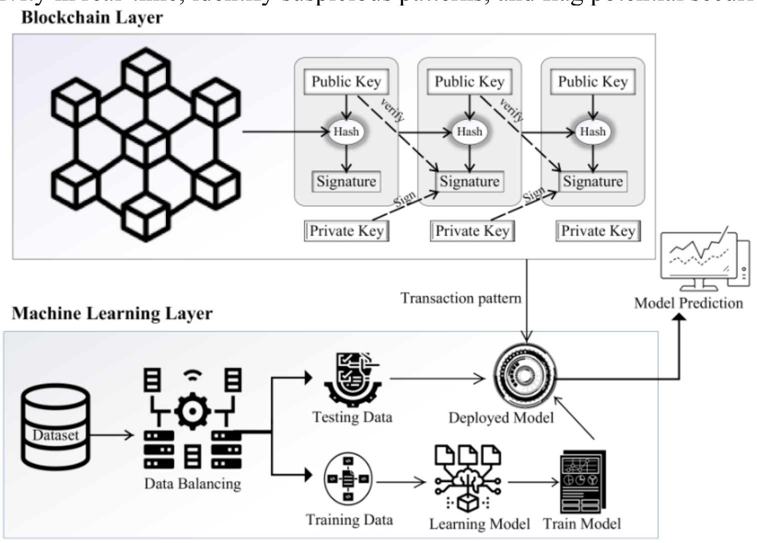

FIGURE 1. Workflow of ML-based Blockchain Attack Detection [9].

Blockchain-related attacks can be categorized into three main groups, shown in Figure 2: network-level attacks (e.g., Sybil and eclipse attacks), consensus-level attacks (e.g., 51% attacks and selfish mining), and application-level attacks (e.g., smart contract manipulation and reentrancy attacks) [10]. ML methods are tailored accordingly: supervised models (e.g., decision trees, SVMs) can classify node behaviors, while unsupervised models (e.g., clustering, autoencoders) can detect transaction anomalies without prior labels.

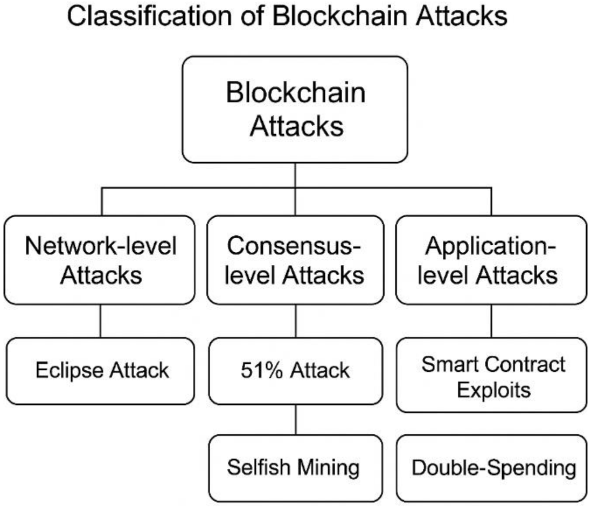

FIGURE 2. Classification of Blockchain Attack Type (Picture credit: Original).

ML-based Methods for Blockchain Attack Detection: Traditional Approaches

Traditional ML techniques have been extensively researched in the context of blockchain security because of their interpretability, efficiency, and ease of deployment. These models are especially suitable for structured data and situations that require low-latency decision-making. This section presents representative studies that utilize classical ML methods to detect blockchain-related attacks, emphasizing the methodological processes and model architectures involved.

Chen et al. proposed a Sybil node detection system for permissionless blockchain networks [11], leveraging Random Forest (RF) and k-Nearest Neighbors (KNN) classifiers. Their approach involved collecting topological and behavioral features from the network, including node connectivity, propagation delay, and message frequency. The selected features were used to train classifiers capable of accurately distinguishing honest nodes from Sybil attackers. The system was designed to operate in real-time with minimal overhead, making it suitable for integration into lightweight blockchain clients.

Li et al. investigated the issue of double-spending detection by applying Support Vector Machines (SVM) to transactional data [12]. Their framework extracted timing intervals, balance patterns, and distributions of transaction values from blockchain records. SVM was chosen for its ability to manage high-dimensional feature spaces and deliver robust binary classification. Their experiments demonstrated that SVM models generalized well across various blockchain environments and were particularly effective at identifying subtle transactional inconsistencies.

Ahmed et al. focused on detecting anomalies in smart contracts on the Ethereum blockchain [13]. They designed a logistic regression model that utilized opcode frequency vectors and function call graphs to represent contract behavior. The training process involved labeling known vulnerable contracts and using regression coefficients to evaluate new contracts. Their method demonstrated that even basic linear classifiers could achieve high precision when enhanced with domain-specific feature engineering.

In a different approach, Singh and Sood addressed the issue of selfish mining by utilizing Decision Tree and Naïve Bayes classifiers [14]. Their system extracted mining-related metadata, including block generation time, miner identity frequency, and block acceptance rates. Decision Trees provided an interpretable, rule-based structure for categorizing mining behavior, while Naïve Bayes acted as a probabilistic baseline. The study concluded that traditional classifiers, although less complex than deep models, still deliver strong performance in controlled settings.

Deep Learning-based Approaches

Deep learning techniques have garnered increasing attention in blockchain security research due to their strong ability to model complex, nonlinear relationships and learn hierarchical feature representations directly from raw data. Unlike traditional machine learning algorithms, which often require manual feature engineering, deep learning models can automatically capture temporal, structural, and contextual patterns, making them well-suited for sophisticated detection tasks involving sophisticated attacks. This section highlights notable studies that apply deep learning approaches to identify and mitigate blockchain-based attacks.

Wu et al. proposed a cascading deep learning framework to de-anonymize Bitcoin transactions [15]. Their architecture comprised multiple stacked neural networks that processed transaction metadata, user behavior patterns, and timing sequences. The system learned to associate transactions with user profiles, exposing vulnerabilities in the blockchain's pseudonymity. The model demonstrated that deep learning can effectively detect hidden patterns in large-scale blockchain data and infer identity linkage with high accuracy.

Zhao et al. developed a model based on Convolutional Neural Networks (CNN) for detecting vulnerabilities in smart contracts [16]. By transforming smart contract code into image-like opcode matrices, the CNN extracted spatial features that represent execution logic and control flows. This innovative encoding strategy enabled the network to classify contracts as either vulnerable or benign. Their results underscored the effectiveness of CNNs in identifying subtle structural flaws in smart contract code.

Kumar and Tripathi introduced a Recurrent Neural Network (RNN) framework to identify sequential attack behaviors in blockchain transactions [17]. Their system utilized Long Short-term Memory (LSTM) units to capture dependencies across transaction sequences. The model was trained on labeled datasets that represent normal and malicious behaviors, achieving robust performance in predicting coordinated attacks such as double-spending and transaction flooding.

Another notable contribution was made by Zhang et al. [18], who proposed a hybrid deep learning model that combines Graph Neural Networks (GNNs) and autoencoders to detect anomalous node behavior in blockchain networks. The GNN component captured relational patterns among nodes and transactions, while the autoencoder component learned latent representations to reconstruct typical behaviors. This combined framework enabled the system to flag outliers that deviate from expected patterns, providing a robust solution for network-level threat detection.

DISCUSSION

Over the past decade, ML has increasingly played a prominent role in enhancing blockchain security. As the threat landscape for blockchain systems has expanded, ranging from Sybil attacks and double-spending to smart contract exploits, researchers have turned to ML techniques to automate threat detection and provide data-driven insights. Early developments in this field predominantly employed traditional ML models, such as decision trees, Random Forests (RF), Support Vector Machines (SVM), logistic regression, and Naïve Bayes classifiers [11-14]. These models gained traction due to their ease of deployment, interpretability, and low computational requirements, making them well-suited for use in permissionless and resource-constrained blockchain environments.

Traditional ML approaches generally rely on manually engineered features extracted from blockchain data, including transaction frequencies, value distributions, network latency, miner behaviors, and opcode usage. As highlighted by Chen et al. and Li et al. [11, 12], such methods have proven effective in detecting specific types of attacks, particularly in scenarios with clear, structured patterns. Moreover, the decision logic behind these models is often transparent, enabling developers and system administrators to trace and validate detection outcomes—a feature particularly valuable in financial or regulatory contexts. However, these models also exhibit notable weaknesses. Their ability to capture complex, nonlinear, or dynamic attack behaviours is inherently limited. Attacks that unfold over time, such as transaction flooding or contract reentrancy, may evade detection due to the static nature of traditional features.

To overcome these limitations, researchers have increasingly explored the use of Deep Learning (DL) models. As demonstrated in multiple studies [15-18], DL offers the ability to learn high-level feature representations directly from raw data such as transaction graphs, opcode sequences, and temporal logs. Convolutional Neural Networks, for instance, have been applied to detect smart contract vulnerabilities by encoding bytecode into structured matrices [16]. Recurrent Neural Networks, particularly Long Short-Term Memory models, have been effective in modelling sequential behaviors within transaction flows [17]. Graph Neural Networks, which learn representations based on the topological structure of blockchain ledgers, have proven useful in detecting anomalous node behaviours or fraud patterns [18].

Despite their success, deep learning methods present their own set of challenges, starting with the issue of interpretability. Unlike traditional ML models, which offer rule-based or probabilistic outputs, deep neural networks operate as black boxes. This opacity in decision-making is a major concern in blockchain environments, where transparency, accountability, and traceability are essential. For example, while Zhao et al. [16] demonstrated the potential of CNNS to detect smart contract bugs, their model lacked interpretability mechanisms to explain which patterns in the opcode sequence led to specific predictions. This makes it difficult for system operators to trust and verify the output, especially in high-stakes applications involving financial assets or governance protocols [7, 15].

Another significant limitation is generalisation ability. Deep learning models are usually trained and tested in controlled environments with specific datasets; yet, blockchain systems are highly heterogeneous. Differences in network protocols, consensus mechanisms, and transaction volumes across platforms mean that models optimized for one setting may perform poorly in another. Zhang et al. [18], for instance, proposed a GNN-autoencoder hybrid for anomaly detection, which worked well in structured settings but faced difficulties adapting to blockchains with irregular or sparse transaction graphs. This variability hinders the widespread deployment of DL-based security tools across multiple blockchain ecosystems.

The third core issue is data availability and imbalance. DL models typically require large, diverse, and well-labelled datasets to achieve competitive accuracy. However, in blockchain security research, obtaining labelled datasets is often difficult. Many attacks, especially sophisticated or zero-day attacks, are rare and poorly documented. Moreover, public blockchain data may be anonymised or obfuscated, complicating the process of generating meaningful labels. Wu et al. [15], who trained a cascading DL framework to infer user identities from transaction metadata, noted that the scarcity of high-quality labelled data limited model robustness. Even when data is available, class imbalance (i.e., few attack samples versus many benign samples) can bias the model, leading to high false negatives or overfitting to the majority class [3, 6].

Furthermore, while Deep Learning reduces the need for manual feature engineering, it significantly increases the demand for computational resources. Model training and inference often require specialised hardware (e.g., GPUS), which may not be feasible for deployment on lightweight blockchain clients or nodes with limited bandwidth and processing power. This computational burden can create scalability issues, especially when real-time attack detection is necessary across large-scale decentralised networks.

Finally, there is a broader concern regarding reproducibility and benchmarking in this emerging field. As noted across multiple studies [10, 17], the lack of standardised datasets, metrics, and experimental protocols makes it challenging to compare models fairly or reproduce results. Variations in data preprocessing, label definitions, and evaluation criteria further obscure meaningful performance comparisons. This lack of reproducibility hinders progress and complicates efforts to identify truly effective models.

In summary, the integration of ML and DL into blockchain security holds potential, although challenges remain. Traditional ML techniques offer interpretability and efficiency, yet they fall short in representational strength. In contrast, deep learning supports complex modelling but encounters obstacles such as explainability, generalisation, data requirements, and reproducibility. As this field progresses, researchers need to evaluate not just accuracy but also practicality, fairness, and robustness when developing blockchain attack detection systems.

Moving forward, future research should address current limitations in ML-based blockchain attack detection by leveraging a combination of expert systems and domain knowledge integration. For instance, enhancing the interpretability of deep models can be achieved by embedding rule-based reasoning or incorporating symbolic AI techniques. This hybrid approach can help explain model decisions in security-critical blockchain applications. To improve generalisation across heterogeneous blockchain systems, domain adaptation and domain generalisation techniques, such as transfer learning or meta-learning, should be explored. These strategies enable models trained on one blockchain environment to be effectively adapted to others with minimal retraining. In response to data scarcity and imbalance, generative models such as Generative Adversarial Networks (GANS) and Variational Autoencoders (VAES) offer promising solutions by synthetically generating attack samples. These synthetic datasets can be used for pre-training deep models, followed by fine-tuning on real blockchain data to improve robustness. Additionally, collaboration between academia and industry could facilitate the creation of benchmark datasets and evaluation protocols, addressing reproducibility concerns. Overall, future work should aim for a balanced integration of model transparency, adaptability, and data efficiency to enhance the resilience and trustworthiness of blockchain security frameworks in real-world deployments.

CONCLUSION

This review explores the growing intersection between blockchain security and Machine Learning, highlighting the application of both traditional and deep learning models for attack detection. Traditional ML approaches, while interpretable and efficient, often struggle to capture the complexity and sequential nature of blockchain data. In contrast, Deep Learning methods demonstrate strong capabilities in learning intricate patterns and representations from raw data, enabling the detection of more advanced and evasive attacks. However, these models face challenges such as poor interpretability, limited generalisation across blockchain platforms, and reliance on large, labelled datasets.

By synthesising recent advancements, this review emphasises the necessity for balanced trade-offs between model complexity, transparency, and deployment feasibility. Although significant progress has been made, blockchain-based attack detection remains an evolving area that requires ongoing attention to both algorithmic innovation and practical implementation. Future studies should consider not only enhancing detection performance but also addressing the real-world constraints that affect the scalability and reliability of Machine Learning systems in decentralised environments.

REFERENCES

1. M. Conti, C. Lal, and S. Ruj, "A survey on security and privacy issues of Bitcoin," IEEE Commun. Surv. Tutor. 20, 3416–3452 (2018).
2. L. Wu, H. Zhang, and Y. Zheng, "Cascading machine learning to attack Bitcoin anonymity," arXiv preprint arXiv:1910.06560 (2019).
3. T. Zeng, Y. Zhang, and M. Li, "Block Hunter: Federated learning for cyber threat hunting in blockchain-based IIoT networks," arXiv preprint arXiv:2204.09829 (2022).
4. A. Sayeed and C. Marco, "Securing majority-attack in blockchain using machine learning and algorithmic game theory: A proof of work," arXiv preprint arXiv:1806.05477 (2018).
5. A. Khan, S. Arshad, and M. Aalsalem, "Blockchain-based attack detection on machine learning algorithms for IoT-based e-health applications," arXiv preprint arXiv:2011.01457 (2020).
6. N. Ali, R. Ali, J. Li, and M. Arif, "Blockchain meets machine learning: a survey," J. Big Data 10, 1–28 (2023).
7. M. R. Islam et al., "Blockchain and Machine Learning: A Critical Review on Security," Information 14, 295 (2023).
8. J. Huang, M. Nie, and Z. Han, "Machine Learning on Blockchain Data: A Systematic Mapping Study," arXiv preprint arXiv:2403.17081 (2023).
9. T. Ashfaq, R. Khalid, A. S. Yahaya, S. Aslam, A. T. Azar, S. Alsafari, and I. A. Hameed, "A Machine Learning and Blockchain Based Efficient Fraud Detection Mechanism," Sensors 22, 7162 (2022).
10. H. Liu, X. Zhang, and T. Yang, "A review on deep anomaly detection in blockchain," J. Inf. Secur. Appl. 80, 103–127 (2024).
11. Y. Chen, A. Shoker, and H. Hegazy, "A Sybil detection framework using ML in blockchain networks," Future Gener. Comput. Syst. 136, 1–12 (2022).
12. X. Li, J. Ma, and P. He, "Double-spending detection using SVMs in blockchain transactions," Expert Syst. Appl. 198, 116797 (2022).
13. T. Ahmed, N. Karim, and R. Rehman, "Smart contract anomaly detection using logistic regression," Inf. Syst. Front. 24, 743–758 (2022)
14. R. Singh and M. Sood, "Detection of selfish mining using traditional ML algorithms," J. Netw. Comput. Appl. 192, 103207 (2021).
15. L. Wu, H. Zhang, and Y. Zheng, "Cascading machine learning to attack Bitcoin anonymity," arXiv preprint arXiv:1910.06560 (2019).
16. X. Zhao, Y. Lin, and C. Wang, "Smart contract vulnerability detection using deep learning with opcode sequence representation," IEEE Access 9, 36610–36622 (2021).
17. R. Kumar and A. Tripathi, "Detection of malicious blockchain transactions using LSTM-based deep learning model," Comput. Commun. 181, 252–260 (2022).
18. Y. Zhang, X. Li, and M. Chen, "GNN-AE: A hybrid graph neural network and autoencoder model for blockchain anomaly detection," Future Gener. Comput. Syst. 139, 102–114 (2023).