

Deep Learning for Blockchain Security: A Concise Survey, Taxonomy, and Open Problems

Omar Mohammed Ahmed ^{1,*} and Shavan Askar ²

1 Computer Information System Department, Technical College of Zakho, Duhok Polytechnic University, 42002, Duhok, Iraq

2 Department of Information Systems Engineering, Erbil Technical Engineering College, Erbil Polytechnic University, 44001, Erbil, Iraq; shavan.askar@epu.edu.iq

* Correspondence author: omar.alzakholi@dpu.edu.krd

Abstract: Blockchain is increasingly used in finance, healthcare, supply chains, and IoT, but it also faces growing security threats such as fraud, smart contract vulnerabilities, intrusions, and malware. Deep learning (DL) has emerged as a powerful tool to detect and mitigate these risks. This paper presents a structured survey of DL applications in blockchain security and proposes a taxonomy linking threat types, integration approaches (on-chain, off-chain, hybrid), DL models, data modalities, and deployment settings. We review recent advances across major security tasks and discuss key limitations and challenges. Finally, we highlight future research directions to improve robustness, efficiency, and practical deployment.

Keywords: blockchain security, deep learning, intrusion detection, smart contracts, DeFi, IoT, GNN, on-chain/off-chain, hybrid architectures

1. Introduction

Blockchain technology was first introduced with Bitcoin in 2008. It promises a radical transformation of how transactions and data are recorded and verified. With its decentralized network of computers, blockchain renders the role of intermediaries redundant, guaranteeing that its records are transparent, immutable and tamper-proof [1]. Each transaction is encrypted and added to a new block of data that is chained through time, making it very difficult to tamper with past records without getting caught. It is these properties that have made blockchain an appealing technology to industries due to the higher degree of trustworthiness and data integrity it can support [2].

Blockchain is one of the most promising technologies and is being adopted across many domains. In finance, it provides the infrastructure for cryptocurrencies and facilitates secure, peer-to-peer financial transactions without any intermediaries. Traceability of the source and path of products in supply chain can be achieved due to the traceability feature of blockchain. In the same way, within the medical field, blockchain enables the secure management of patients' data, increase the transparency in clinical trials, and guarantees the authenticity of drugs [3].

Even with all its benefits, blockchain still has security weak spots. In recent years, several major breaches have revealed serious issues. Common problems include 51% attacks, bugs in smart contracts, and weak points at user devices or services, especially in cryptocurrencies and DeFi platforms [4]. Research also shows that flaws in consensus algorithms, wallet apps, and smart contracts give attackers ways to steal money or cause disruption [5].

As blockchain grows up, these risks matter even more and raise new questions. Surveys indicate that both users and developers see security as a top priority—about 70% want stronger protections for blockchain

systems [5]. Studies point to better consensus designs, secure coding practices, and improved monitoring as practical steps to prevent future attacks. If the community tackles these gaps with fresh ideas, blockchain can become a reliable foundation for safe digital transactions across industries [6].

The remainder of this paper is organized as follows. Subsection 1.1. reviews blockchain fundamentals. Subsection 1.2. introduces deep learning and its applications to security. Subsection 1.3. describe positioning against existing Surveys. Subsection 1.4. describes our survey methodology. Subsection 1.5. presents the proposed taxonomy. Section 2 frames related work and case studies. Subsections 2.1 to 2.6 review the literature by threat class — fraud detection, smart-contract security, intrusion detection, consensus enhancement, data privacy, and malware detection. Section 3 discussion, and finally the conclusion.

1.1. Blockchain

There are few important components that constitute a blockchain system which together, provide a secure and transparent decentralized system of storing data. Combined, these elements form a decentralized, secure, and tamper-proof network that has the potential to be the foundation for multiple blockchain use cases like digital currencies, supply chain tracking, and digital identity verification [7].

A blockchain is made up of a few key parts that work together to keep it running, secure, and decentralized. Blocks are the basic pieces: each one holds a list of transactions, a time stamp, and the cryptographic hash of the block before it. This creates a tamper-evident chain. Because blockchains are decentralized, there's no single company or server in charge, computers (nodes) in a peer-to-peer network handle transaction directly with each other. To keep everyone's copy of the ledger in sync, the network uses a consensus method, an algorithm that checks and agrees on which transactions are valid. Blockchains can also run smart contracts, which are small programs that automatically execute agreements written in code, enabling decentralized apps (dApps). Security relies on cryptography hash functions (like SHA-256 or Keccak-256) and digital signatures to protect data and verify identities [8]. Figure 1 shows a simple overview of how a blockchain works.

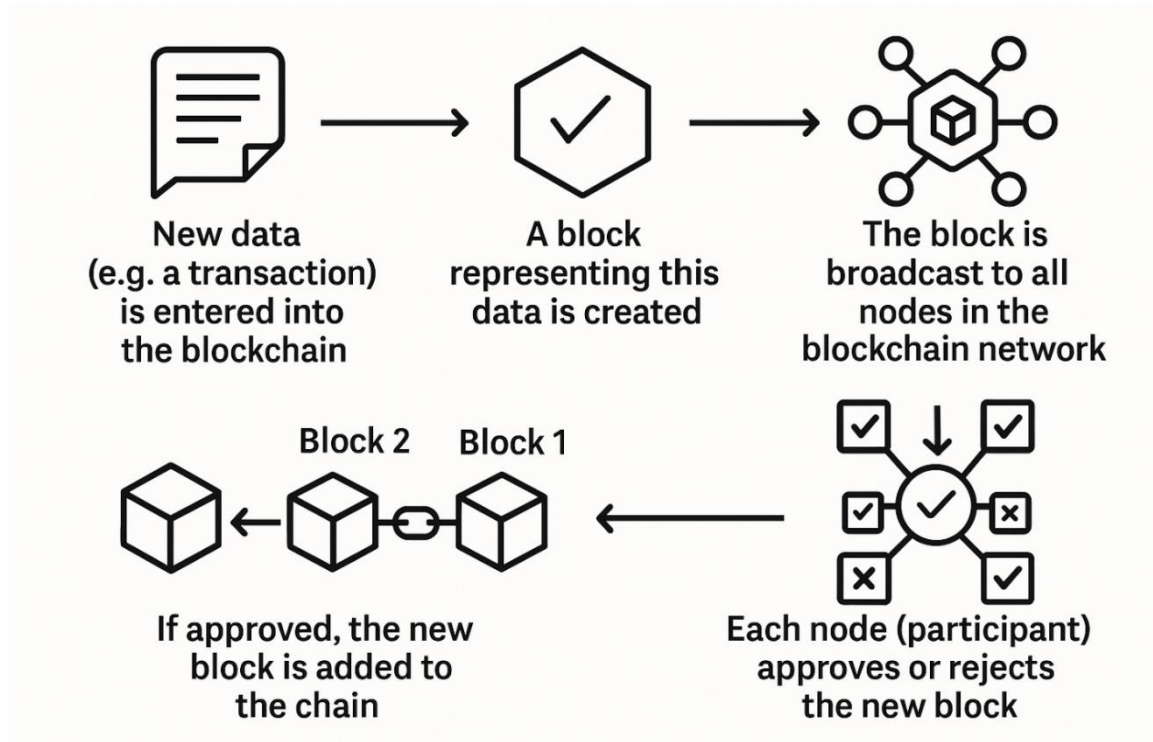


Figure 1. *Blockchain process.*

Blockchain security relies on a few core ideas. Immutability means that once information is written to a block, changing it later is extremely hard, which protects the ledger's integrity. Transparency and auditability let everyone see the transactions, so it's easy to check and track what happened [9]. Decentralization removes any single point of failure, making the network harder to attack and more reliable. Finally, consensus

algorithms let many independent nodes agree on which transactions are valid, keeping the blockchain consistent and trustworthy [10].

Consensus sits at the heart of blockchains: it's how distributed nodes agree on the ledger without a central authority. Different methods come with different trade-offs. Proof of Work (PoW) asks miners to solve hard puzzles very secure, but energy-hungry. Proof of Stake (PoS) picks validators based on how much they "stake," cutting energy use but risking centralization and "nothing-at-stake" issues. Delegated Proof of Stake (DPoS) relies on a small group of elected validators, which boosts speed and scalability but increases centralization. Practical Byzantine Fault Tolerance (PBFT) uses multiple voting rounds to handle malicious nodes, but it doesn't scale well to very large networks. All of these approaches balance security, efficiency, decentralization, and scalability in different ways, and each has its own weaknesses (for example, 51% attacks, Sybil attacks, and collusion). These risks must be weighed carefully when designing a secure, trustworthy blockchain system [11]. A quick summary of the main consensus mechanisms appears in Table 1.

Table 1. Overview of Major Consensus Algorithms in Blockchain Systems

Consensus Mechanism	Mechanism	Security Strengths	Weaknesses	Energy Efficiency	Scalability
Proof of Work (PoW) [12]	Miners solve cryptographic puzzles	High security, prevents Sybil attacks	Energy-intensive, slow transactions	Very Low	Poor
Proof of Stake (PoS) [13]	Validators are chosen based on the number of coins staked	Resistant to Sybil attacks through staking requirements	Susceptible to "nothing at stake" attacks	High	Good
Delegated Proof of Stake (DPoS) [14]	Selected nodes validate transactions on behalf of others	Fast finality and reduced attack surface with fewer nodes	Potential centralization risks	High	Excellent
Practical Byzantine Fault Tolerance (PBFT) [15]	Nodes reach consensus through a series of voting rounds	Tolerant to malicious nodes	Does not scale well for large networks	High	Poor to Moderate

1.2. Deep Learning and Its Application in Security

Deep Learning (DL), it is a subfield of machine learning that uses artificial neural networks to separate high level features from data sources. DL has significantly changed the cybersecurity landscape by increasing the accuracy of fraud detection, anomaly detection, intrusion detection and malware classification. The DL model contains several hidden layers between input and output layer for learning the hierarchical features of the data. The main elements of DL models can be summarized as follows [16]:

- **Neurons:** each of these units, which transfer information in a way that is similar to biological neurons, calculates the data and then pass it to the next layer.
- **Activation Functions:** These are required to introduce non-linearity in the neural network such as ReLU, Sigmoid, and Tanh help neural networks capture complex patterns.
- **Backpropagation:** An algorithm used to modify the network weights in order to reduce the errors made during predictions.
- **Optimization Algorithms:** Techniques like Stochastic Gradient Descent (SGD), Adam, and RMSprop optimize model training.

Several deep learning architectures have been adopted to enhance blockchain security as shown in table 2.

Table 2. Deep Learning Architectures to Enhance Blockchain Security

Deep Learning Model	Application in Blockchain Security	Strengths	Limitations
Convolutional Neural Networks (CNNs) [17]	Malware detection, smart contract security analysis	Extracts spatial patterns, high accuracy	Requires large labeled datasets
Recurrent Neural Networks (RNNs) [18]	Fraud detection, anomaly detection in transactions	Captures sequential dependencies	Training complexity, vanishing gradient problem

Long Short-Term Memory (LSTMs) [19]	Intrusion detection, network traffic analysis	Overcomes short-term memory issues in RNNs	Requires high computational power
Autoencoders [20]	Detecting anomalies in blockchain transactions	Learns compressed representations, detects outliers	May not work well for real-time detection
Graph Neural Networks (GNNs) [21]	Analyzing blockchain transaction networks	Captures complex entity relationships	High training cost, requires large-scale blockchain data

Deep Learning is well-suited for blockchain security due to its ability to [19]:

1. **Detect Fraud & Anomalies:** DL models have the ability to identify fraudulent patterns in blockchain transactions, reducing risk.
2. **Analyze Smart Contracts for Vulnerabilities:** NLP deep learning models search Solidity code for issues.
3. **Prevent Malware Attacks:** CNN-based smart contract malware classifier to detect out the dangerous smart contracts.
4. **Enhance Intrusion Detection Systems:** Autoencoders and LSTMs identify suspicious network activity in blockchain nodes.

1.3. Positioning Against Existing Surveys

Several recent surveys touch on the intersection of deep learning and blockchain, but each leaves clear gaps. Earlier reviews on blockchain security either treat machine learning as a single homogeneous block or focus narrowly on one threat class such as smart-contract vulnerabilities or DDoS in IoT. Reviews of deep learning in cybersecurity, in turn, rarely engage with the unique constraints of decentralized ledgers — gas costs, on-chain inference budgets, mempool latency, and the off-chain/on-chain integration boundary. As a result, readers must consult several disjoint sources to obtain a unified view.

This survey differs from prior work along four dimensions. Scope: we cover six threat classes (fraud, smart-contract bugs, intrusion, consensus/MEV, privacy, malware) within a single framework. Taxonomy: we organize the literature along five orthogonal axes — threat class, integration locus, deep-learning family, data modality, and deployment setting. Deployment realism: we record the on-chain/off-chain integration locus for every surveyed work and discuss gas and latency consequences of on-chain inference. Critical lens: we discuss reproducibility, dataset shift under time-ordered splits, robustness gaps, and the absence of cost-aware metrics rather than catalogue accuracies.

Contributions. This survey makes four contributions that, to our knowledge, no prior review in this intersection has combined:

1. A unified five-axis taxonomy covering six threat classes — fraud, smart-contract bugs, intrusion, consensus/MEV, privacy, and malware — rather than one in isolation.
2. Explicit on-chain / off-chain / hybrid integration labelling for every surveyed work, with a quantitative discussion of gas and latency consequences.
3. A code/data availability audit across the surveyed papers, turning the reproducibility critique from rhetorical into evidential.
4. A critical analysis that contrasts random-split benchmark performance with time-ordered deployment performance, and frames the open problems accordingly.

1.4. Survey Methodology

We ran structured searches (2020–2025) in Scopus, IEEE Xplore, ACM DL, Web of Science, and Google Scholar using Boolean combinations of (blockchain OR smart contract OR DeFi OR Ethereum OR IoT) AND (deep learning OR neural OR GNN OR transformer OR autoencoder) AND (security OR intrusion OR anomaly OR fraud). Inclusion: peer-reviewed English articles with empirical evaluation on blockchain-relevant data; Exclusion: opinion pieces, purely theoretical crypto without ML, unreproducible gray literature (unless used only as case context). Two rounds of screening removed duplicates and off-scope items. We extracted: dataset/modality, task, DL model family, integration locus (on/off/hybrid), metrics (AUC/F1/precision-recall, latency), compute and on-chain cost (gas, TPS). Disagreements were resolved by re-reading or exclusion. Figure 2 shows their distribution by publication year and dominant deep-learning

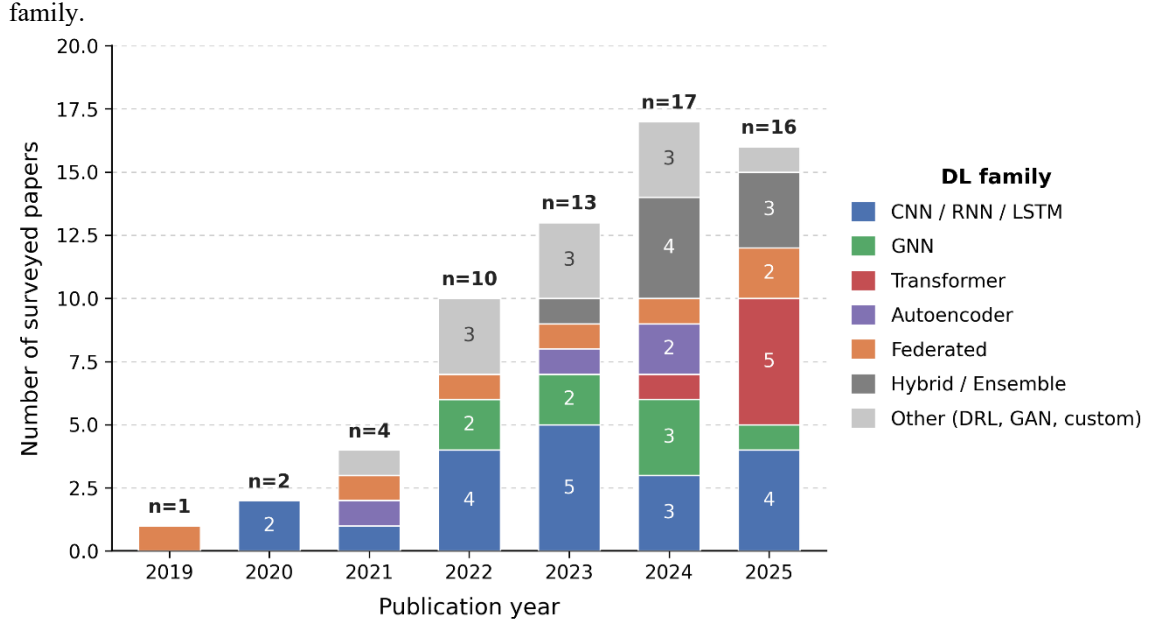


Figure 2. Distribution of the surveyed papers by publication year and dominant deep-learning family.

1.5. Taxonomy of Deep Learning for Blockchain Security

We categorize works along five orthogonal axes as summarized in Figure 3:

- Threat class — what the attacker tries to achieve: fraud/manipulation, intrusion/malware, smart-contract bugs & exploits, Sybil/phishing, consensus/MEV, privacy deanonymization.
- Integration locus — where the DL component runs relative to the chain: on-chain (contracts, zk/TEE-assisted inference), off-chain (indexers, SIEM, oracles), hybrid (on-chain verification with off-chain DL).
- DL family — the model architecture: GNNs (transaction/contract graphs), Transformers (time-series/logs/code), CNNs/RNNs (netflow/bytecode images/sequences), Autoencoders (unsupervised anomaly), GNN+AE/Contrastive (representation learning).
- Data modality — what is fed into the model: transaction graphs, bytecode/AST, mempool/events, network traffic, log traces, prices/DEX flows, IoT sensor streams.
- Deployment setting — the operational environment: DeFi/CeFi, smart-contract auditing, L1/L2 core infrastructure, IoT/IIoT, cross-chain bridges.

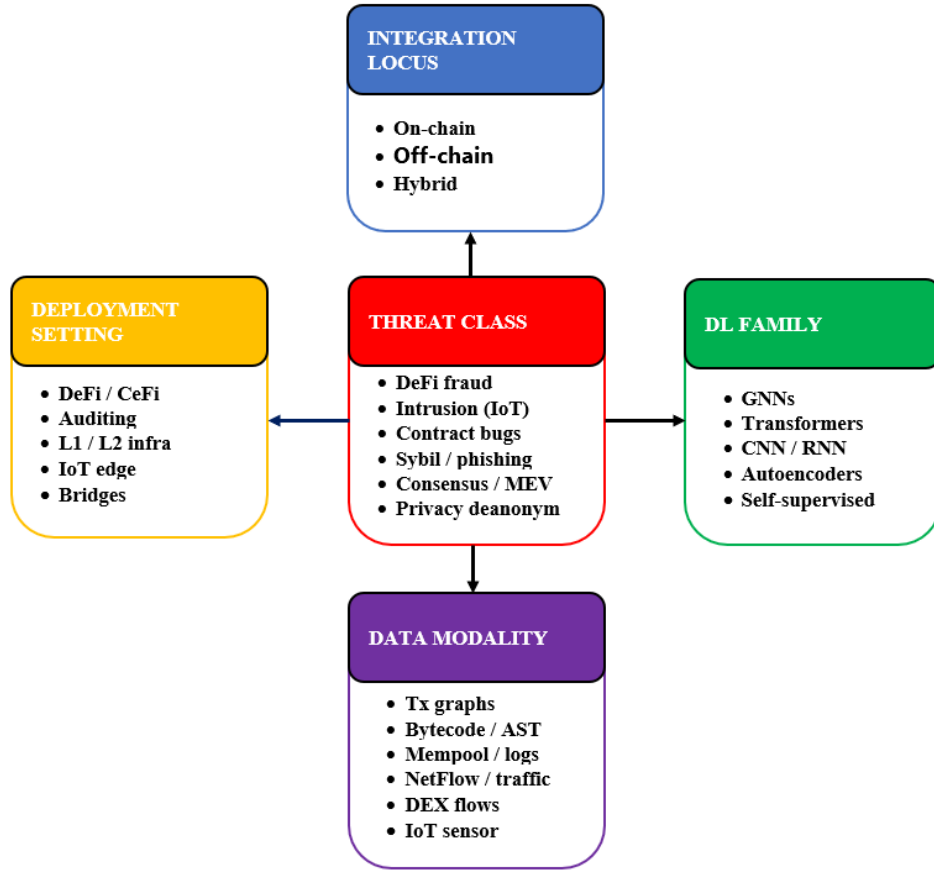


Figure 3. Master Taxonomy: Threats, Integration Locus, DL Families, Data Modalities, and Deployment Settings

2. Related works

Security, privacy and operational challenges of blockchain will decrease as blockchain keeps developing. The neural network-based techniques have become a powerful method to solve the above problems in different areas. In this section, we examine important places where deep learning could potentially commingle with blockchain, and include real-world examples as well as recent research findings.

We begin with fraud detection, looking at attacks like the Poly Network exploit and how models such as autoencoders and graph neural networks (GNNs) can flag suspicious transactions. Next, we cover smart-contract failures (like the DAO bug) and explain how deep learning helps analyze code and find vulnerabilities.

We then move to intrusion detection, focusing on 51% attacks on Ethereum Classic and how deep learning improves real-time threat monitoring. We also examine ways AI can strengthen consensus algorithms to make them more efficient and secure.

Finally, we explore decentralized identity, data privacy, and malware detection, supported by case studies and techniques like federated learning, GANs, and opcode analysis. Together, these examples show how deep learning and blockchain are coming together to build more secure, reliable systems. Figure 4 illustrates common blockchain security vulnerabilities.

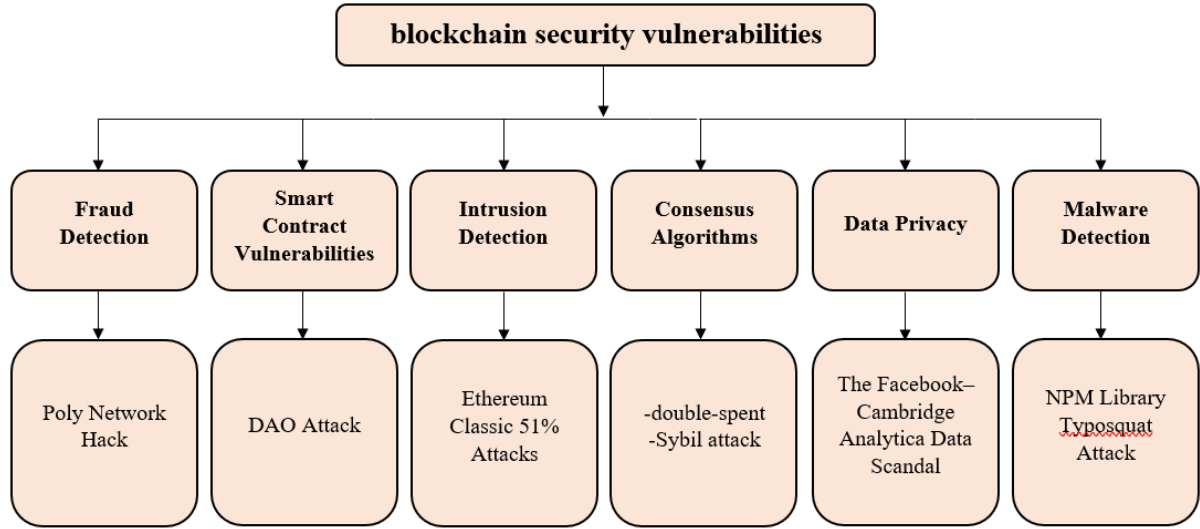


Figure 4. Blockchain Security Vulnerabilities

2.1. Fraud Detection in Blockchain Transactions

The issue of fraud detection in blockchain transactions is increasingly a topic of interest, especially in the wake of large hacks exposed through weaknesses in decentralized finance (DeFi) solutions. Deep learning models can be a great solution for detecting and preventing fraudulent transactions in blockchain system and its integration provides a high potential to improve the detection and prevention of fraud activities in blockchain based systems.

Last August, Poly Network, a system that enables users to send tokens to one another over several blockchains, was hit by a huge hack. Hackers abused Poly Network’s weaknesses to shift about \$610 million in digital assets to wallets they own. The attack has been considered one of the biggest DeFi hacks to have occurred [22].

Attackers targeted Poly Network’s tools for moving assets between blockchains like Ethereum, Binance Smart Chain, and Polygon. They found a weakness in the smart contracts that manage these cross-chain transfers and used it to make unauthorized transactions.

Poly Network publicly asked the hackers to give the funds back, warning about possible legal action and the harm to the wider crypto community. In a surprising twist, the hackers began returning the money a few days later, saying their goal was to expose the flaws, not to steal assets.

In recent blockchain fraud-detection work, researchers are using deep learning and ensemble methods to boost accuracy and robustness. For example, Roy et al. (2022) built an Ethereum fraud classifier using an LSTM with a dense layer and used Information Gain to pick the most useful features, outperforming standard models [23]. Meanwhile, Ghosh et al. (2023) tackled Bitcoin fraud with an ensemble of deep models (including MLP, feed-forward networks, and attention-based LSTMs) to capture time-based and non-linear patterns, achieving strong accuracy and reliability [24]. Continuing with the interpretability, Olusegun and Yang (2024) proposed a fraud detector model for Ethereum based on Tabular Prior-Data Fitted Network (TabPFN), which is composed of several deep learning models such as MLP, LSTM, CNN and CLSTM, instructed by the features selected through SHAP the SHapley Additive exPlanations-based feature selection. In this case, their model does not only favor accuracy but emphasizes explainability, exposing insights about feature influence and model behavior [25].

Lakhan et al. (2022) proposed a serverless Blockchain-Enabled Task Scheduling (SBETS) model tailored for intelligent transportation systems (ITS), which integrates a deep graph convolutional neural network within a fog-cloud environment. Their approach effectively reduced processing overhead and improved fraud detection by 50%, utilizing a function-based price model to optimize computational cost [26]. Taher et al. (2024) built an Ethereum fraud detector by combining several classifiers with hard voting, reaching 99% accuracy, and they added XAI tools like LIME so users can see why the model made a decision [27]. Asiri

and Somasundaram (2025) used a graph convolutional network on the Elliptic dataset to spot illegal Bitcoin transactions and found it outperformed logistic regression and random forest, with 98.5% accuracy and better AUC and RMSE [28]. Baabdullah et al. (2024) paired federated learning with blockchain to create a privacy-preserving credit card fraud system, letting different institutions train models without sharing raw data; they tried LSTM, CNN, and random forest, tuned with ADAM, SGD, and MSGD, and fixed class imbalance using SMOTE [29]. Finally, Gu and Dib (2025) designed an Ethereum fraud detector that ensembles SVM, XGBoost, and random forest; after grid-search tuning, it scored above 98% on accuracy, precision, recall, and F1, and ran quickly with about 0.13-second inference time [30].

Airlangga (2024) explored deep learning-based anomaly detection in the Open Metaverse by evaluating multi-layer perceptrons (MLP), convolutional neural networks (CNN), and long short-term memory (LSTM) models. Using a dataset of 78,600 blockchain transactions from virtual environments, the study employed 5-fold cross-validation with mean squared error (MSE) as the primary evaluation metric. Results showed that MLP outperformed other architectures, offering a robust and adaptive method for detecting anomalies in complex virtual economies [31]. Sun et al. (2025) proposes a joint pipeline that (i) turns raw Ethereum transaction attributes into “transaction sentences” and learns semantic embeddings with a BERT-style Transaction Language Model, (ii) builds a Transaction Attribute Similarity Graph (NPMI/TF-IDF) and an Account Interaction Graph, and (iii) jointly trains a multi-head attention fusion network with a GCN on the account graph. Evaluated on three Ethereum datasets (MulDiGraph, B4E, SPN), reporting +10–20% F1 over SOTA. Off-chain analytics; authors also release a new dataset.[32]

Kang and Hwang (2025) Introduces a Fraud-Oriented Transformer with Random Traversal: it stochastically traverses the transaction graph to form informative sequences for a Transformer, integrating deep graph features to classify fraudulent Ethereum accounts. On large Ethereum graphs, the paper reports superior AUC vs. baselines (details in article). Off-chain detection model [33]. Ferreira (2025) works with the Elliptic Bitcoin dataset. They do exploratory data analysis, pick and compress features (using methods like PCA and UMAP), and test several classifiers. XGBoost comes out on top for both two-class and three-class tasks. Dimensionality reduction helps with understanding the results, but full feature sets still perform best. The study also points out two big issues: severe class imbalance and very high feature dimensionality. The workflow runs off-chain [34].

Taken together, these papers suggest a clear shift toward deeper, more interpretable learning models that fit the complex patterns in blockchain transactions. As summarized in Table 3, multiple deep-learning methods have been tried for blockchain fraud detection, each offering specific advantages and trade-offs.

Table 3. Various Deep Learning Approaches Have Been Applied Blockchain Fraud Detection

Criteria	Blockchain Focus	Blockchain Integration	Main Technique	Dataset	Performance	Key Contributions	Weaknesses/ Challenges
Lakhan et al. (2022) [26]	Serverless Blockchain (SBETS)	Hybrid	Deep Graph Convolutional Network (GCN)	Simulated ITS dataset	10% cost reduction, 50% fraud detection gain	Serverless task scheduling and secure ITS fraud detection with GCN	Simulation-based, lacks real-world testing
Roy et al. (2022) [23]	Ethereum	Off-chain	LSTM + Dense Neural Network with Information Gain for feature selection	Ethereum transaction dataset	Significant improvement over baseline models	Introduces a reliable Ethereum fraud detection model using LSTM; highlights need for quality datasets; feature selection reduces model complexity	Lack of public dataset limits reproducibility; focuses only on binary classification; limited explanation of model decisions
Ghosh et al. (2023) [24]	Bitcoin	Off-chain	Ensemble of MLP, FNN, and Attention-based LSTM	Bitcoin transaction data	Accuracy: 99.62%; Precision & Recall >99%	Combines multiple deep learning models to achieve superior accuracy and robustness in fraud detection	Only focused on Bitcoin network; lacks model interpretability; dataset specifics are missing
Olusegun and Yang (2024) [25]	Ethereum	Off-chain	Ensemble of deep learning models (MLP, LSTM, CNN, CLSTM, TabPFN) with SHAP-based feature selection and explainable AI	Ethereum transaction dataset	Accuracy: 99.2%	Incorporates explainable AI (SHAP) and transformer-based TabPFN to provide both high accuracy and transparency in fraud detection	Complexity and potential overfitting from ensemble models; limited real-world deployment discussion
Airlangga (2024) [31]	Open Metaverse blockchain	Off-chain	MLP, CNN, LSTM	metaverse transactions	MLP achieved lowest MSE	Comparative DL approach to metaverse fraud detection	Offline only; lacks real-time capability
Taher et al. (2024) [27]	Ethereum	Off-chain	Ensemble Learning + XAI (LIME)	Ethereum transaction dataset	99% accuracy	Ensemble + XAI for interpretable fraud detection	Scalability and real-time deployment not evaluated
Baabdullah et al. (2024) [29]	Credit card blockchain-integrated	Hybrid	Federated Learning + Blockchain, LSTM, CNN, RF	Federated dataset from 3 banks	Improved accuracy and privacy	Privacy-preserving fraud detection via FL and blockchain	Trade-off between latency and privacy; dataset access

Asiri and Somasundaram (2025) [28]	Bitcoin	Off-chain	Graph Convolutional Network (GCN)	Elliptic Bitcoin Dataset	98.5% accuracy, AUC 0.9444, RMSE 0.1123	GCN applied to illicit Bitcoin detection	Dataset partially labeled; possible generalization issues
Gu and Dib (2025) [30]	Ethereum	Off-chain	Ensemble (RF, SVM, XGBoost)	Ethereum transaction dataset	>98% on all metrics; 0.13s inference time	Real-time, high-performance ensemble fraud detection model	Class imbalance in Ethereum transaction data
Sun et al. (2025) [32]	Ethereum	Off-chain	Hybrid LM+Graph	MulDiGraph, B4E, SPN	+10–20% F1 over SOTA across all three datasets	First to fuse transaction semantics (LM) with similarity and structure under joint optimization; releases dataset	Complexity & compute; vocabulary/design choices; interpretability of fused views
Kang and Hwang (2025) [33]	Ethereum	Off-chain	Transformer with Random Traversal over the transaction graph + deep graph features	Large-scale Ethereum transaction networks	Higher AUC than baselines on large networks	Scalable traversal strategy to form sequences for a transformer while preserving graph context	Potential training cost; explaining decisions may be non-trivial
Gomes and Ferreira (2025)	Bitcoin	Off-chain	Dimensionality reduction + ML (PCA/UMAP + FS) with multiple classifiers; XGBoost best	Elliptic Bitcoin dataset	3-class: XGBoost around 0.926 acc; 2-class performs higher;	Clean pipeline highlighting explainability via reduced feature sets; comprehensive baseline sweep	Imbalance & label scarcity

2.2. Smart Contract Security

Smart contracts are automated contracts where the terms of the agreement between the parties are directly interpreted into code. They are built on top of blockchains such as Ethereum and can be used to create dApps in a world without middlemen. Yet, the immutable and autonomous nature of smart contracts exposes them to a range of security bugs, which have resulted in large financial losses. Figure 5 compares two Solidity smart contract implementations: Contract A (Vulnerable) and Contract B (Normal). Both contracts allow users to deposit and withdraw funds. However, Contract A demonstrates a classic reentrancy vulnerability by transferring Ether to the sender (`msg.sender.call.value(...)`) before updating the state variables (i.e., setting the lock and deducting the balance). This ordering allows malicious actors to re-enter the function before the state is updated, potentially draining the contract. In contrast, Contract B mitigates this vulnerability by updating the state before making the external call, following the checks-effects-interactions pattern, a recommended best practice in Ethereum smart contract development. Deep learning has been recognized as a promising technique to identify and suppress such vulnerabilities.

Contract A (Vulnerable)	Contract B (Normal)
<pre> 1 contract Ree { 2 mapping(address => uint256) public balances; 3 4 event WithdrawFunds(address _to,uint256 _value); 5 6 function depositFunds() public payable { 7 balances[msg.sender] += msg.value; 8 } 9 10 function WithdrawFunds (uint256 _weiToWithdraw) public { 11 require(balances[msg.sender] >= _weiToWithdraw) 12 require(!locked[msg.sender]); 13 14 msg.sender.call.value(_weiToWithdraw)(); 15 locked[msg.sender] = true; 16 balances[msg.sender] -= _weiToWithdraw; 17 locked[msg.sender] = false; 18 } 19 }</pre>	<pre> 1 contract Ree { 2 mapping(address => uint256) public balances; 3 4 event WithdrawFunds(address _to,uint256 _value); 5 6 function depositFunds() public payable { 7 balances[msg.sender] += msg.value; 8 } 9 10 function WithdrawFunds (uint256 _weiToWithdraw) public { 11 require(balances[msg.sender] >= _weiToWithdraw) 12 require(!locked[msg.sender]); 13 14 locked[msg.sender] = true; 15 msg.sender.call.value(_weiToWithdraw)(); 16 balances[msg.sender] -= _weiToWithdraw; 17 locked[msg.sender] = false; 18 } 19 }</pre>

Figure 5. An example of smart contracts [35]

The most high-profile example of a smart contract vulnerability is The DAO (Decentralized Autonomous Organization) hack in 2016. The DAO was an Ethereum-based investor-directed venture capital fund that raised more than \$150 million (worth of Ether). In June 2016, The DAO's smart contract code was allegedly attacked after vulnerable spot reentrancy, resulting into the theft of an estimated \$70 million worth of 3.6 million ether. This attack contravened a tremendous amount of monetary value and led to a forced hard fork of the Ethereum blockchain, causing Ethereum Classic [36]. The reentrant exploit helped the hacker repeatedly execute the withdraw function before the first transaction finished which eventually emptied the DAO's coffers.

Typical Vulnerabilities of Smart Contracts, open Issues of Smart contracts Beyond reentrancy, there are actually other ways that smart contracts can be vulnerable to:

Integer Overflow and Underflow: It is a type of error when the results of a calculation exceed the maximum value that the data type can store or fall below the minimum value.

Timestamp Dependence: Using block timestamps for critical operations can be abused by miners.

Unhandled Exceptions: Contracts are unpredictable or unable to execute when there is not proper exception handling.

These characteristics can be linked to the complexity of smart contract languages, such as Solidity, and the difficulties encountered for the secure development of smart contracts.

Researchers are using deep learning more and more to spot bugs in smart contracts. Li et al. (2023) combines several kinds of features (both static and dynamic) and feed them into GCNs and Bi-LSTMs to capture how the code is structured and what it means. Their system does well on common issues like reentrancy and timestamp dependence [37]. Li et al. (2022) takes a different route with Link-DC, which builds a contract graph and uses a Deep & Cross Network to turn simple, sparse features into richer ones, leading to strong accuracy across many vulnerability types [38]. Osei et al. (2024) introduce WIDENNET, which converts bytecode into opcode vectors and uses a wide-and-deep model. It strikes a good balance between speed and accuracy, though it covers fewer vulnerability categories

[39].

To fix the lack of labeled data, Hwang et al. (2024) propose CGGNet, a compiler-guided generator. Instead of a standard GAN, it uses the compiler as an oracle and Monte Carlo Tree Search to create millions of valid, varied contract examples, helping models avoid underfitting [40]. Han et al. (2023) merge syntax and semantics by fusing features from ASTs and CFGs using TextCNN plus graph neural networks, boosting both accuracy and recall across several vulnerability types [41]. Li et al. (2024) focus on DEX contracts with StateGuard. They build ASTs, extract five key dependency signals (like control and data dependencies), and apply a GCN to catch subtle state mistakes that can cause security problems [42].

Other recent work follows the same theme of richer representations. Kasula et al. (2024) build a unified graph that mixes structural and semantic cues and report better results than traditional approaches [43]. Deng et al. (2023) fuse multiple “views” of a contract (source code, opcodes, and CFGs) by training deep models per view and combining their outputs for stronger predictions [44]. Chen et al. (2024) present Clear, which uses contrastive learning to model how vulnerable and safe contracts relate to each other. Clear significantly improves performance on many metrics and beats 13 prior deep-learning methods [35].

Wang et al. (2025) studies push beyond single-view code modeling toward multimodal and Transformer-centric pipelines. TMF-Net fuses textual (code tokens) and structural features (AST/CFG) via a multi-scale transformer fusion network, consistently beating single-modality baselines—especially on reentrancy and timestamp-dependence—by learning complementary representations across modalities [45]. Balci et al. (2025) present VASCOT targets compiled contracts, performing sequential analysis of EVM bytecode with a Transformer, and evaluates on ~16.4k verified Ethereum contracts deployed in 2022; independent summaries report ~90% test accuracy and superior speed vs. LSTM baselines, highlighting the practicality of bytecode-level detection when source is unavailable. Choi et al. (2025) proposed LLM + Graph Structural Analysis (CMC) integrates CodeBERT with CFG/AST-derived graphs (nodes embedded with Sent2Vec, centrality-weighted) and prompts an LLM using the fused features, reaching Acc 86.70 / F1 84.46 and beating CodeBERT-only baselines—evidence that graph cues materially help text models. Table 4 provides a comparative analysis of recent deep learning techniques applied to smart contract vulnerability detection.

Table 4. Various Deep Learning Approaches Have Been Applied Blockchain smart contract

Criteria	Blockchain Focus	Blockchain Integration	Main Technique	Dataset	Performance	Key Contributions	Weaknesses/ Challenges
Li et al. (2022) [38]	Ethereum, VNTChain	Off-chain	Deep and Cross Network (Link-DC model) combining contract graphs and expert features	Smart contracts from Ethereum and VNTChain	Reentrancy: 94.37%, Timestamp Dependency: 92.11%, Infinite Loop: 85.29%	Fuses expert patterns with constructed graph features for richer context	Relies on manually defined expert patterns; possible overfitting to known patterns
Li et al. (2023) [37]	Ethereum	Off-chain	Multimodal feature fusion using GCN and Bi-LSTM	Ethereum contracts	Reentrancy: 85.73%, Timestamp Dependency: 85.41%, Precision: 96%, Recall: 90%	Combines static and dynamic features; uses deep learning for comprehensive multimodal analysis	May have limited generalization to unseen vulnerabilities; performance may vary with dataset balance
Han et al. (2023) [41]	Ethereum	Off-chain	Syntactic and Semantic Feature Fusion using TextCNN and GNN	Ethereum Smart Contracts (AST + CFG)		Fusion of AST and CFG features for better vulnerability detection	Limited to five vulnerability types, may not scale
Deng et al. (2023) [44]	Ethereum	Off-chain	Deep Learning with Multimodal Decision Fusion	Public Smart Contract Dataset	Accuracy: 94.8%, AUC: 0.886	Fuses multi-modal information (SC, OP, CFG) via DL	High computational overhead with multiple modalities
Hwang et al. (2024) [40]	Ethereum	Off-chain	Compiler-Guided Generation with Monte Carlo Tree Search	Generated via CGGNet (3.37M contracts)	3,369,397 contracts generated	First smart contract-focused augmentation using compiler-guided learning	Depends on compiler and MCTS configuration quality
Osei et al. (2024) [39]	Ethereum	Off-chain	Wide & Deep Neural Network (WIDENNET) using opcode vectors	Real-world smart contracts	Accuracy: 83.07%, Precision: 83.13%	Introduces wide and deep learning model for combining general and specific vulnerability patterns	Limited to detecting reentrancy and timestamp dependency; lower performance compared to other models
Li et al. (2024) [42]	Ethereum (DEXs)	Off-chain	Graph-based Deep Learning using GCN on AST Dependencies	46 DEX Projects Dataset (5,671 contracts)	Precision: 92.24%, F1-score: 94.25%	Detection of state derailment in DEXs using dependency-aware GCN	Focus only on DEX-related state issues

Kasula et al. (2024) [43]	Ethereum	Off-chain	Graph-based Deep Learning Approach	Ethereum explorer	N/R	Integrates structural and semantic contract features	Limited methodological detail;
Chen et al. (2024) [35]	Ethereum	Off-chain	Contrastive Learning with Transformer-based Architecture	40K+ Real-world Contracts	F1-score improved by 9.73% to 39.99%	Applies contrastive learning for improved contract context understanding	Requires pair-wise contract samples and correlation labels
Wang et al. (2025) [45]	Ethereum	Off-chain	Multimodal, multi-scale Transformer fusion over code tokens + AST/CFG	Public Ethereum contracts	N/R	Shows complementarity of text + structure; scalable fusion design	Potential compute overhead for graph + transformer; generalization beyond tested vulns not detailed
Balci et al. (2025) [46]	Ethereum	Off-chain	Transformer over EVM bytecode sequences	~16,4xx verified 2022 contracts	N/R	Works without source code; strong scalability claims; contemporary, large verified set	Potential label noise from auto-annotation; need reproducible release of data/code
Choi et al. (2025)	Ethereum	Off-chain	CodeBERT (text) + CFG/AST/opcode graphs, Sent2Vec + centrality	Not mentioned	Acc 86.70%, Prec 85.24%, Rec 84.87%, F1 84.46%	Clear text–structure fusion recipe; transparent metrics;	Moderate F1 vs newer multimodal systems; dependency on LLM prompting; compute for graph extraction;

2.3. *Intrusion Detection in Blockchain Networks*

Blockchain is decentralized and very secure, but it's not completely safe from cyberattacks [47]. Intrusions such as the well-known 51% attack have exposed real weaknesses. That's why strong protection is still essential, especially powerful intrusion detection systems that can spot and stop attacks quickly. Recent advances in deep learning have made these systems much better at detecting and responding to threats.

A clear example is the series of 51% attacks on the Ethereum Classic (ETC) network [48]:

- January 2019: Attackers gained control of over half of the network's hashing power, reorganized the chain, and double-spent about 219,500 ETC (worth over \$1.1 million).
- August 2020: ETC was hit by multiple 51% attacks in just one month, causing millions of dollars in losses and seriously undermining network security.

These incidents show why we urgently need advanced, real-time intrusion detection to protect blockchain networks.

Beyond 51% attacks, two attack families have come to dominate the threat landscape on smart-contract platforms: Maximal Extractable Value (MEV) exploitation and oracle manipulation. MEV refers to the profit a block proposer or sophisticated searcher can extract by reordering, inserting, or censoring transactions within a block — for example, sandwiching a victim's swap on a decentralized exchange, or front-running a profitable liquidation. Detecting MEV is fundamentally a temporal and graph-structured problem: candidate attacks manifest as suspicious transaction orderings in the mempool, often involving back-to-back trades against the same liquidity pool by related addresses. Oracle manipulation is the closely related problem of an attacker pushing a manipulated price into a DeFi protocol's price feed, typically via a flash loan that distorts a low-liquidity pool the oracle reads from. Detection here combines on-chain features (flash-loan call patterns, pool reserves, swap sizes) with cross-venue price signals from off-chain feeds to flag implausible deviations. Both MEV and oracle attacks share two characteristics the surveyed literature underweights: they are multi-block (attacks span several transactions and sometimes several blocks) and economically constrained (the attacker is profit-driven, so plausible attacks must remain economically rational).

The novel use of blockchain combined with deep learning in IDS as of the late has inspired the development of novel solutions to ensure the security of IoT and cloud networks [49]. In A deep blockchain framework-enabled collaborative intrusion detection system for securing IoT and cloud networks Al-Kadi et al. (2020) described a cooperative IDS which applies a BiLSTM-deep learning model for sequential network traffic, beside the help of Ethereum-blockchain and smart contracts for privacy and decentralized trust across IoT and cloud [50]. Elsewhere, Optimized blockchain-based deep learning model for cloud intrusion detection by Alasmari (2024) introduces a hybrid deep learning ensemble that consists of CNN, RBM, and GAN which is optimized by the hybrid seahorse and bat algorithm (HSHBA). This method could improve feature selection and prediction accuracy with blockchain as secure data storage and audibility [51]. Finally, Wang (2022) Embedded Architecture Protocol Network Intrusion Detection focuses on deep residual learning and adopted multi-threshold fusion, and spectral feature extraction, specifically for embedded blockchain-based protocols. This approach is focused more on accuracy of the abnormal traffic detection and no information regarding dataset normalization and larger blockchain integration [52].

Katib and Ragab (2023) introduced the H3SC-DLIDS model, which integrates Harris Hawks Optimization (HHO) with a sine cosine algorithm (SCA) for feature selection, followed by LSTM-based autoencoder (LSTM-AE) classification and hyperparameter tuning using the Arithmetic Optimization Algorithm (AOA). This hybrid methodology enhances DDoS attack detection accuracy in blockchain-enabled IoT environments [53]. Similarly, Mansour (2022) proposed the PRO-DLBIDCPS model for cyber-physical systems, employing an Adaptive Harmony Search Algorithm (AHSA) for feature selection and an attention-based BiGRNN classifier. The approach incorporates poor-rich optimization (PRO) for hyperparameter tuning and uses blockchain for securing CPS communications [54].

In a related study, Alkhonaini et al. (2024) developed SPOHDL-ID, which applies a sandpiper optimizer (SPO) for feature selection and combines CNN with a stacked autoencoder (SAE) for intrusion detection. The model further enhances accuracy through hyperparameter tuning using the Beetle Search Optimization Algorithm (BSOA), achieving notable performance on benchmark datasets [55]. Another hybrid system, BHS-ALOHD by Alamro et al. (2023), uses Ant Lion Optimizer (ALO) for selecting optimal features and integrates CNN with LSTM for deep learning classification. Flower Pollination Algorithm (FPA) is then applied for hyperparameter tuning, offering significant improvements in IoT healthcare intrusion detection [56].

Kamali Poorazad et al. (2023) addressed vulnerabilities in SDN-enabled IIoT by integrating a CNN-based IDS within the SDN control plane, complemented by a blockchain system at the network layer. This dual-layer approach

improves real-time threat mitigation and data integrity, particularly in SCADA environments [57]. Finally, Li, Sun, and Sun (2023) introduced a privacy-preserving intrusion detection system using federated deep learning that combines LSTM and GRU models. Blockchain ensures secure and private data transfer across distributed nodes, while the system maintains high accuracy and data integrity in sensitive domains like healthcare [58].

Alruwaili (2025) combine DL with blockchain-assisted infrastructure: LBCCD-GJO (GRU classifier; GJO feature selection; metaheuristic hyper-parameter tuning) on ToN-IoT, and MOBCF-ADDLM (DBN with evolutionary tuning) on BoT-IoT—both report $\approx 99\%$ -level accuracy and position blockchain for decentralization/tamper-proofing rather than on-chain inference [59]. Aliyu et al. (2025) introduces a continuously learning IDS whose LSTM-based core ingests intrusion logs; self-updating neural nets are anchored per block and executed inside TEEs to harden integrity. Blockchain provides decentralized coordination/provenance; experiments on Binance Smart Chain and Ethereum Classic datasets report 98.5% accuracy with 1.5% FPR. Integration is effectively hybrid (TEE inference/off-chain; model state anchored on-chain) [60].

Cumulatively, these studies underline various and complementary methodological progresses in blockchain-supported deep learning for intrusion detection. Table 5 presents a comparative analysis of recent studies that integrate deep learning techniques with blockchain technology for intrusion detection.

Table 5. Various Deep Learning Approaches Have Been Applied Blockchain Intrusion detection

Criteria	Blockchain Focus	Blockchain Integration	Main Technique	Dataset	Performance	Key Contributions	Weaknesses/ Challenges
Al-Kadi et al. (2020) [50]	Ethereum	off-chain	BiLSTM for detecting intrusions	UNSW-NB15 and BoT-IoT	Most attack types detected with over 95% accuracy	Integrates deep learning and blockchain for secure VM migration and collaborative IDS	Potential scalability issues and complexity in deployment
Wang (2022) [52]	Embedded within architecture protocol	off-chain	Deep Residual Learning with edge info fusion	Not clearly specified	N/R	Uses deep residual fusion and association rules for embedded systems	Lack of standard datasets, unclear integration with broader blockchain systems
Mansour (2022) [54]	Blockchain for CPS communication and integrity	off-chain	AHSA for feature selection, ABi-GRNN, PRO optimizer	NSL-KDD-2015 CICIDS-2017	N/R	Uses attention-based BiGRNN and hybrid optimization in a CPS blockchain context	generalization and reproducibility concerns
Katib & Ragab (2023) [53]	Generic blockchain framework	off-chain	HHO + SCA for feature selection, LSTM-AE, AOA for hyperparameter tuning	BoT-IoT	Accuracy: 99.05%	Hybrid metaheuristic optimization for feature selection; LSTM-AE for classification; Blockchain for secure IoT data flow	Model complexity and computation cost due to multiple optimization layers
Alamro et al. (2023) [56]	Used for secure medical data transmission	off-chain	ALO for FS, CNN + LSTM, FPA for tuning	ToN-IoT CICIDS-2017	N/R	Applies hybrid optimization and DL in healthcare; leverages CNN-LSTM synergy	lacks interpretability and energy efficiency assessment
Kamali Poorazad et al. (2023) [57]	Generic Blockchain	off-chain	CNN-based IDS + Blockchain integration in SDN	Custom SDN attack simulation dataset (version 3)	N/R	Dual-layer security (IDS + blockchain); focuses on SDN-IIoT integration	Does not benchmark blockchain overhead or real-time latency

Li, Sun & Sun (2023) [58]	Used for privacy-preserving federated learning	off-chain	Federated LSTM-GRU with blockchain-secured training	NSL-KDD 2015	Accuracy: 99.01%	Focuses on privacy-preserving training; hybrid RNN + blockchain	Model complexity and hardware heterogeneity limit scalability
Alasmari (2024) [51]	Ethereum	off-chain	Ensemble of CNN, RBM, GAN with HSHBA optimization	UNSW-NB15	Accuracy: 99.12%, Precision: 99%	Hybrid optimization (seahorse + bat), advanced ensemble DL models for IDS	High computational cost due to complex ensemble and optimization techniques
Alkhonaini et al. (2024) [55]	Blockchain for data sharing integrity	off-chain	Sandpiper optimizer, CNN-SAE, BSOA tuning	ToN-IoT, CICIDS-2017	Accuracy: 99.59% (ToN-IoT), 99.54% (CICIDS-2017)	Efficient FS and hybrid DL for high accuracy; robust feature learning and hyperparameter tuning	Requires multiple tuning layers, increasing computational cost
Alruwaili (2025) [59]	blockchain-assisted IoT environment	Hybrid	GRU classifier; GJO feature selection; PDO-DE hyper-param tuning (LBCCD-GJO)	ToN-IoT	Up to 99.67% Acc	Robust DDoS pipeline with evolutionary optimization; compares processing time vs baselines	On-chain design details light; need device-level constraints and reproducibility checks.
Aliyu et al. (2025) [60]	Ethereum	Hybrid	LSTM core; self-updating NNs per block;	Binance Smart Chain, Ethereum Classic	Acc. 98.5%, FPR 1.5%	Integrate TEEs with blockchain for tamper-resistant, self-adapting IDS;	Trust/attacker model for TEEs and on-chain cost not fully benchmarked;

2.4. *Enhancing Consensus Algorithms*

Consensus algorithms are the heart of blockchain, as they guarantee that all decentralized nodes come to an agreement on the transaction logic and maintain consensus of the distributed ledger. Primitive consensus mechanisms such as Proof of Work (PoW) and Proof of Stake (PoS) have been widely used for the security of blockchain systems. Nevertheless, these approaches are not without problems, including that of high energy consumption, vulnerability to centralization, and security attacks that are possible. Implementing deep learning methods with consensus algorithms is a new innovative exploring direction for improving security, efficiency and robustness.

A 51% attack happens when a single party or group of parties takes control of more than half of a blockchain network's mining or staking power, effectively allowing them to alter transactions, double spend coins, and degrade the network's reliability. Ethereum Classic (ETC) suffered several 51% attacks in the past. ETC experienced a major attack in January 2019 where around \$1.1M worth of double spend ETC was made. Despite deploying network upgrades, there were further attacks in August and October 2020, with estimated losses of \$5.6 million and \$1.68 million, respectively[61].

The content storage network is resilient against such Sybil attacks because it uses a stable-form of anonymous identifiers. Take IOTA network as an example, many attacks related to double-spending, e.g., parasite chain attack, becoming quite serious challenges of the new networks, especially for those young networks when they were still immature and were unable to defend [62].

The application and use of deep learning in the blockchain-based consensus protocol has already stimulated attention for new tactics to resolve the inefficiencies that characterize the convergence of classical PoW (Proof of Work). Zhi et al. (2025) propose the Blockchain Consensus based on Distributed Deep Learning Work (BCDDL), converting the distributed deep learning (DDL) training into a mining process. They propose a methodology that optimizes the allocation of training data according to the node's computing power, and uses the Model Aggregation Algorithm (MAA-TM) and the Dynamic Incentive Mechanism (DIM-TSCR) to maximize accuracy, security and maximize resource utilization [63]. In contrast, He et al. (2024), Proof of Privacy Preserving Machine Learning (PPMML) is proposed guarding confidential data and models at the consensus level utilizing encrypted deep learning and the hybrid approach to take the training procession. Their method is focused on confidentiality preserving while keeping model accuracy, and thus applicable to privacy-demanding blockchain applications [64].

Liu et al. (2021) propose Proof of Learning (PoLe), which swaps energy-heavy calculations for training neural networks to reach consensus; a statistical ML checker validates models and deters cheating, making the system more practical and sustainable [65]. Building on adaptive ideas, Islam et al. (2023) introduce MRL-PoS, a PoS method that uses multi-agent reinforcement learning with dynamic voting and a reward-penalty scheme to discourage malicious nodes and promote honest behavior [66]. For IoT settings, Goh et al. (2022) present TDCB-D3P, which relies on a Dueling Double Deep Q-Network with prioritized replay to choose trustworthy nodes, raise transactions per second, and tune parameters on the fly for better scalability and security [67]. In transport, Kim (2022) combines a blockchain consensus approach with LSTM-based anomaly detection for vehicle cybersecurity, showing promise against real-time attacks in a virtual testbed [68]. Badruddoja et al. (2022) split the workload: models are trained off-chain, while on-chain smart contracts handle predictions using Taylor-series approximations of activation functions, enabling accurate decentralized inference on MNIST despite blockchain limits [69]. Alruwaili (2024) introduced FGADL-DEVCA, a consensus protocol for IoT-cloud environments that integrates blockchain with a deep autoencoder for fault detection. The system utilizes a Football Game Algorithm for hyperparameter optimization and supports decentralized edge verification to ensure data integrity across distributed networks [70]. Amarnadh & Rao (2025) propose a deep-learning + blockchain pipeline for bank credit-risk evaluation: data are preprocessed and features selected via an enhanced Mountain Gazelle metaheuristic (pheromone-guided), then classified with a Multi-layer Residual coordinate-Boosted Sooty-tern Attention Network (MRBSAN) trained with focal loss; final credit decisions are written to a consortium blockchain that uses a Byzantine lightweight-fault-tolerant consensus to provide tamper-evident storage across nodes. Reported performance across three datasets is strong: Accuracy 97.2%, Precision 98.6%, Recall 98.6%, F1 98.0%. The figures show the off-chain ML pipeline with on-chain record storage and identical copies across nodes [71].

On-chain inference: gas and latency reality. A practical question that pervades on-chain inference is the cost-latency budget. On Ethereum mainnet, a simple ERC-20 transfer costs roughly 21,000 gas; a smart-contract function call invoking even a small fully-connected layer with fixed-point arithmetic can easily exceed 1–5 million gas per inference, translating to several USD per call at typical gas prices and to block-confirmation latencies in the tens of

seconds. These figures rule out on-chain inference for any task with mempool-speed deadlines (DEX front-running detection, MEV mitigation) and severely constrain the model size that fits within the block gas limit. The trade-off explains why the surveyed works overwhelmingly favour a hybrid pattern: heavy inference runs off-chain (or inside a TEE or zk-circuit), and only a compact commitment — a hash, a Merkle root, or a succinct proof — is anchored on-chain. Approximation techniques such as Taylor-series linearization of activation functions make small on-chain models possible, but their accuracy still trails their off-chain counterparts. Future deployments will need to publish gas-per-inference and end-to-end latency alongside accuracy if on-chain DL is to graduate from prototype to production. Table 6 presents a comparative analysis of recent blockchain consensus mechanisms that incorporate deep learning techniques.

Table 6. Various Deep Learning Approaches Have Been Applied Blockchain Consensus Mechanisms

Criteria	Blockchain Focus	Blockchain Integration	Main Technique	Dataset	Performance	Key Contributions	Weaknesses/ Challenges
Liu et al. (2021) [65]	PoLe (Proof of Learning)	Hybrid	Neural network training with Secure Mapping Layer (SML) to prevent cheating	MNIST, IRIS, and CIFAR-10; data released to all network nodes	Stable block generation and efficient transaction processing shown in experiments	Prevents cheating with SML; redirects PoW effort toward useful neural network training	SML adds extra computational overhead; assumes honest behavior for validation
Badruddoja et al. (2022) [69]	Ethereum-like Smart Contract Blockchain	On-chain	On-chain DL Prediction via Taylor Series	MNIST	99% accuracy; scalable on-chain prediction	Taylor-based approximation of activation functions for on-chain DL	Computational cost of on-chain operations; limited to simple models
Kim (2022) [68]	Custom LSTM Blockchain Framework	Hybrid	LSTM + Blockchain Consensus	Vehicle sensor and CAN data (simulated)	Real-time anomaly detection with secure propagation	Integrated LSTM for auto anomaly detection in vehicles	Specific to automotive use; lacks generalizability
Goh et al. (2022) [67]	Delegated PoS (e.g., EOS-like)	Hybrid	Deep Reinforcement Learning (D3P with TD-PBFT)	Simulation with varying node trust levels	High TPS, reduced computing cost, improved security	Trust-based node selection; DRL optimization; mitigates malicious influence	Focus on simulation; lacks real-world scalability validation
Islam et al. (2023) [66]	Generic PoS Blockchain	Hybrid	Multi-Agent Reinforcement Learning (MRL-PoS)	Simulated Blockchain Environment	Adaptive consensus; efficient malicious node detection	Introduced MRL-PoS; dynamic penalty-reward; RL-based voting system	No real-world deployment; complexity in multi-agent coordination
He et al. (2024) [64]	PPPML (Proof of Privacy-Preserving Machine Learning)	Hybrid	Privacy-preserving encrypted model training and hybrid learning	Confidential datasets; encryption techniques used to protect data privacy	Maintains model accuracy while preserving privacy; simulation validated	Introduces privacy protection for DL in blockchain consensus; hybrid approach balances privacy and efficiency	High computational cost of encrypted training; complex implementation
Alruwaili (2024) [70]	Custom IoT-Cloud Blockchain	Hybrid	FGADL-DEVCA (DL + Blockchain)	IoT-cloud simulation with injected faults	Enhanced fault detection;	DAE for fault detection; Football	Novelty in tuning may lack optimization

Zhi et al. (2025) [63]	BCDDL (Proof of Distributed Deep Learning Work)	Hybrid	+ Edge verification) Distributed Deep Learning used as mining task; dynamic incentives (DIM-TSCR) and model aggregation (MAA-TM)	Training data dynamically allocated based on node capability	efficient edge validation Increases resource utilization and reduces impact of malicious nodes	Game Algorithm for tuning Efficient use of resources in blockchain by integrating useful DL tasks; novel incentive and aggregation mechanisms	efficiency; experimental Requires reliable node performance estimation; potential vulnerabilities in data allocation
Amarnadh & Rao (2025) [71]	Byzantine lightweight fault-tolerant consensus	Hybrid	MRBSAN classifier; feature selection via enhanced Mountain Gazelle	Three credit datasets	Acc 97.2%, Prec 98.6%, Rec 98.6%, F1 98.0%	New MRBSAN architecture for credit-risk scoring	interpretability/XAI not discussed

2.5. *Data Privacy in Blockchain*

Privacy online is still a major concern, especially with blockchains, which are designed to be transparent by default. Decentralization can strengthen security, but it also makes it harder to keep sensitive information secret [72]. In this space, deep learning offers practical tools to help protect data and reinforce privacy within blockchain networks.

A well-known example of why privacy matters is the Facebook Cambridge Analytica scandal. In 2018, it was revealed that Cambridge Analytica had obtained data from up to 87 million Facebook users without their consent, largely through a personality-quiz app. Reports indicated that this information was used to influence voters during the 2016 U.S. presidential election and the Brexit referendum. The incident exposed the weaknesses of centralized data control and underscored the need for strong privacy safeguards.

Researchers have also explored ways to combine blockchain and deep learning to improve privacy and decentralization [73]. For instance, a 2019 study by Fadaeddini and colleagues proposed a decentralized deep-learning setup on the Stellar blockchain. In their approach, each participant trains the model locally and shares only the learned parameters (not the raw data) so collaborators can benefit from shared learning without exposing private information. The architecture leverages the native exchange of Stellar, along with custom tokens (DLC, VLM) to reward honest participation and to punish malice action - a mechanism is Know Your Customer (KYC). The approach escapes reliance on a central authority by employing a distributed ledger technology for organizing the training cooperation and rewards [74].

Mahmood et al. (2022) introduced secure protocols namely ZKPs, differential privacy and adversarial training in their blockchain-based federated learning system. Their approach is developed on the Ethereum blockchain and adopts a CNN model for training based on the Fashion-MNIST dataset. It combines local model updates with secure aggregation and encrypted communication to defend against adversary attacks including poisoning, inference and backdoor. There is also a transparent incentive mechanism to encourage the honest behavior of the node [75].

Instead of data integrity issue of decentralized FDL, Zhu and Li (2021) improved the data privacy of decentralized FDL by presenting a privacy-preserving framework (PDFDL). They used secure multiparty computation based on a variant of the Paillier encryption for aggregating encrypted gradients without performing decryption. Running on ethereum, the smart contract orchestrates the training rounds and rewards users according to their contribution. Their scheme deals securely with the model updates and achieves global model privacy through threshold decryption, dealing with both data leakage risk and malicious participant risk [76].

Awotunde et al. (2023) proposed a hybrid model integrating Kernel Principal Component Analysis (KPCA) with Convolutional Neural Networks (CNNs) and blockchain for smart city security. The method includes preprocessing, feature reduction, and classification stages, using ToN-IoT and BoT-IoT datasets for evaluation, achieving high intrusion detection accuracy [77]. Raju et al. (2023) developed a secure IoT-healthcare system using Optimal Key-based Hybrid Elliptic Curve Cryptography with Fully Homomorphic Encryption (OK-HECCFHE), combined with a deep learning prediction model based on Gated Recurrent Units (GRUs). Blockchain ensures secure data storage, while hybrid optimization algorithms improve encryption and prediction performance [78]. BlockDLO, introduced by Kokila and Reddy (2024), offers a five-phase IoT security architecture combining chaotic authentication, page rank-based clustering, smart contracts, and CNN-based anomaly detection. Blockchain ensures data integrity, while deep learning enhances intrusion detection in edge networks [79].

In digital governance, Malik et al. (2023) proposed a blockchain-based secure data exchange platform using bonobo optimization, a lightweight Feistel structure for encryption, and a deep reinforcement learning model for anomaly detection. The system was tested on BoT-IoT and ToN-IoT datasets to improve security and interoperability in governance systems [80]. Li et al. (2024) presented a privacy-preserving blockchain transaction framework using probabilistic encryption, zero-knowledge proofs, and Graph Neural Networks (GNNs). The model supports both privacy and regulatory oversight without compromising user identity [81]. Ghani et al. (2024) addressed biometric privacy by integrating Generative Adversarial Networks (GANs) and deep learning within a blockchain framework for facial recognition. This approach enhances data anonymity and security in decentralized environments [82]. Finally, Zhou et al. (2022) proposed a blockchain-based data-sharing system for brain-computer interface (BCI) applications using synthetic EEG data generation via GANs and transfer learning. The system enables privacy-aware storage and analysis sharing via Ethereum-based smart contracts [83].

Bhardwaj and Sumangali (2025) introduces PPFBXAIO, integrating FL (Flower-style orchestration) with a permissioned blockchain (SHA-256-backed logging + access control smart contracts), XAI (SHAP at the client), and optimization (Lévy Grasshopper Optimization Algorithm) for feature selection and FL hyper-parameter tuning. Uses

Entropy Deep Belief Network (EDBN) as the classifier. Evaluated on Breast Cancer Wisconsin and UCI Heart; reports ~95–97% class metrics (precision/recall/F1 around mid-90s), plus latency ↓ by ~81 ms and throughput ↑ by ~109 TPS vs FedAvg/FL-MPC/FL-RAEC/PEFL. Bezanjani et al. (2025) proposes a three-phase framework: (1) reputation-based trust estimation for IoT devices with off-chain storage; (2) a lightweight PoW blockchain for immutability and secure messaging; (3) a lightweight LSTM for anomaly detection/classification in real time. Reported simulation shows +2% in precision/accuracy/recall, +5% attack-detection rate, and −3% false-alarm rate vs. baselines. The design targets scalability and low latency for healthcare IoT [84]. Table 7 presents a comparative analysis of recent studies that integrate deep learning with blockchain technologies for data privacy.

Table 7. Various Deep Learning Approaches Have Been Applied Blockchain Data Privacy

Criteria	Blockchain Focus	Blockchain Integration	Main Technique	Dataset	Performance	Key Contributions	Weaknesses/ Challenges
Fadaeddini et al. (2019) [74]	Stellar	Hybrid	Decentralized federated learning using Stellar with Deep Learning Coin (DLC) incentive mechanism	General AI/Big Data (not specified)	Focus on incentivization and decentralization, no quantitative benchmarks provided	Introduced Stellar blockchain for deep learning model training with a novel token-based incentive mechanism (DLC)	Data availability, trust in validators, scalability of Stellar network
Zhu and Li (2021) [76]	Ethereum	Hybrid	Privacy-preserving federated deep learning with secure multi-party computation and Paillier encryption	not specified	Evaluated with respect to ciphertext size, throughput, training accuracy, and time	Proposed PDFDL model with encrypted gradient aggregation and smart contract-based model updates	Complex encryption adds overhead; does not fully guarantee privacy due to blockchain exposure risks
Zhou et al. (2022) [83]	Ethereum	Hybrid	GAN + Transfer Learning + Blockchain	EEG data (wearable sensors)	Secure, privacy-aware data sharing	BCI privacy preservation with synthetic data and blockchain	Limited application to small, specific use cases
Mahmood and Jusas (2022) [75]	Ethereum	Hybrid	Federated learning with multi-layered blockchain security and Zero-Knowledge Proofs (ZKPs)	Fashion-MNIST	Shows training accuracy and loss on CNN with Fashion-MNIST dataset	Integrated FL with blockchain to defend against multiple attacks and introduced multi-layered defense with ZKPs	FL still susceptible to attacks like poisoning and inference; complexity of securing large decentralized networks
Malik et al. (2023) [80]	Optimized Blockchain (Bonobo algorithm)	Hybrid	Feistel Encryption + DRL + Smart Governance	BoT-IoT, ToN-IoT	Enhanced security and fraud detection	Secure data exchange and governance using DRL and blockchain	Privacy vs. transparency trade-off
Awotunde et al. (2023) [77]	General Blockchain (ePoW)	Hybrid	Blockchain + CNN + KPCA	ToN-IoT, BoT-IoT	Improved accuracy	Hybrid deep learning and blockchain model for smart city security	High computational cost and complexity in multi-layer integration
Raju et al. (2023)[78]	Custom Blockchain	Hybrid	OK-HECCFHE + GRU + HPB-ASMO	IoT-Healthcare (custom data)	Secure prediction with encrypted cloud communication	Hybrid encryption and deep learning for healthcare IoT	Overhead from complex hybrid cryptographic scheme

Kokila & Reddy (2024) [79]	Ethereum	Hybrid	BlockDLO + CNN + Smart Contracts	Custom in-house testbed	High accuracy, reduced latency, lower energy use	Five-layer blockchain-deep learning model for IoT edge security	Complex architecture, limited generalizability
Li et al. (2024) [81]	Generic Blockchain	Hybrid	Probabilistic Encryption + ZKPs + GNN	Simulated transaction data	Balanced privacy and regulatory compliance	Framework for privacy-preserving, auditable blockchain transactions	Implementation complexity, not consensus-specific
Ghani et al. (2024) [82]	Decentralized Blockchain	Hybrid	GAN + Deep Learning + Blockchain	CelebA FFHQ HF	Improved privacy in biometric systems	Privacy-enhanced facial recognition on blockchain	Scalability and real-time performance concerns
Bhardwaj and Sumangali (2025) [85]	Generic Blockchain	Hybrid	Entropy Deep Belief Network (EDBN) within FL	Breast Cancer Wisconsin; UCI Heart	~95–96% (BCW) and ~93% (Heart);	First end-to-end FL+ Blockchain+ XAI+ Optimization pipeline with auditable explanations	Added complexity; permissioned ledger + XAI + optimizer overhead;
Bezanjani et al. (2025) [84]	Generic Blockchain	Hybrid	LSTM for anomaly detection.	Simulation-based evaluation	+2% precision, accuracy, recall	proves feasibility of lightweight DL	Simulation; PoW—even lightweight—may add energy/latency overhead

2.6. *Malware Detection in Blockchain Networks*

Although inherently so, as decentralized and immutable structures, blockchain networks are not immune to malware attacks. Adversaries are constantly finding new and advanced ways of attacking weaknesses in these networks resulting in huge financial losses and establishing doubts in blockchain system itself. Deep learning, a sub-field of machine learning, has become a powerful technology in identifying, and preventing those threats. This section focuses on the use of deep learning solutions for malware detection in blockchain networks, validating them on interesting case studies coming from real life.

On the heels of that revelation, cybersecurity researchers discovered an attack using more than 287 malicious packages published in the Node Package Manager (NPM) repository. In the attack, the malicious party used typo squatting methods — giving packages names similar to those of popular libraries — to trick developers into downloading them. Upon installation, these packages would run scripts that communicated with Ethereum smart contracts to obtain C2 server addresses, and then download and execute malware on the victim's machine [86].

Due to the explosion in cyber attacks via malware in the context of Internet of Things (IoT) in healthcare and governance, there is an increasing interest of the security researchers to propose blockchain based deep learning techniques [87]. Alotaibi (2021) presented a new BCMPB-RIDMPL (Biserial Miyaguchi–Preneel Blockchain-Based Ruzicka-Indexed Deep Perceptive Learning) for Internet of Medical Things (IoMT) malware detection. The model is based on multilayer perceptron and feature selection is performed by applying point biserial correlation as a first step, then the selected features are cryptographic hashed by Miyaguchi–Preneel blockchain in the second step. The hash integrity is checked for during classification with Ruzicka indices, in order to increase accuracy and speed of detection [88].

In related work, Nalinipriya et al. (2021) proposed a Vision-based Deep Stacked Autoencoder (DSAE) for detecting ransomware in blockchain networks. Their approach first extracts statistical features from opcode sequences (n-grams and TF-IDF) and then the extracted features are transformed using Box-Cox and the features are fused. Classification is conducted by a deep stacked autoencoder trained by a hybrid WMFO method on the fine-tuning of model weights, and demonstrates good performance for ransomware detection [89]. In addition to these attempts, Pawar et al. (2024) proposed a blockchain-based deep learning structure that is custom made for digital governance within the scope of IoT. They combine Bonobo optimization for SCDA and two-layered XGBoost model with a greedy search algorithm and bat optimization for parameter adjustment. This hybrid framework is developed to detect malware over distributed smart spaces, and interoperability and data security are also being considered [90].

Pastor et al. (2020) proposed a comprehensive system for detecting encrypted crypto mining malware traffic using machine and deep learning techniques without relying on payload inspection or sensitive metadata. Their methodology involved building a realistic experimental setup with encrypted Monero (XMR) mining traffic and designing two feature sets (Tstat and NetFlow), with Tstat-derived features proving significantly more informative. They evaluated classifiers such as Random Forests, fully connected neural networks, and logistic regression, highlighting the superior performance of complex models even with encrypted traffic [91]. In the domain of Android malware detection, Şafak et al. (2025) introduced BlockDroid, a lightweight convolutional neural network (CNN) ensemble model that processes DEX files converted into grayscale images. By combining EfficientNetB0, MobileNetV2, and a custom CNN through stacking, the system achieved 97.38% accuracy. Uniquely, it incorporates blockchain to securely store prediction results, preventing re-analysis and ensuring data immutability [92]. Finally, Denysiuk et al. (2023) also explored a blockchain-enhanced malware detection system but focused on using a deep learning model validated through a Proof-of-Action consensus mechanism. Their method enables parallel malware analysis across nodes, increasing detection robustness and reaching an accuracy of up to 99.33% [93]. Nannan et al. (2025) Andro-BCFL proposes a Federated Learning + blockchain framework so devices train locally and synchronize updates via a ledger/smart contracts, improving timeliness and privacy while avoiding raw data sharing [94]. Table 8 presents a comparative analysis of recent studies that integrate deep learning with blockchain technologies for malware detection.

Table 8. Various Deep Learning Approaches Have Been Applied Blockchain Malware Detection

Criteria	Blockchain Focus	Blockchain Integration	Main Technique	Dataset	Performance	Key Contributions	Weaknesses/Challenges
Pastor et al. (2020) [91]	None (focuses on ML for encrypted traffic)	Off-chain	Random Forest, DNN with Tstat-derived flow features	Custom encrypted Monero mining traffic	N/R	Detection of encrypted cryptomining flows via network behavior profiling	No use of blockchain; dependent on realistic setup
Alotaibi (2021) [88]	Miyaguchi–Preneel cryptographic hash-based blockchain	Hybrid	Deep multilayer perceptive learning with Ruzicka index and biserial correlation	Drebin dataset	Accuracy 93–95% depending on sample size	Combines biserial correlation, blockchain hashing, and Ruzicka index for secure and fast malware detection in IoMT	Potential challenges in scalability for very large, real-time datasets were not discussedcomparison benchmarks
Nalinipriya et al. (2021) [89]	Blockchain network for decentralized security layer	Hybrid	Deep Stacked Autoencoder with Water Wave–based Moth Flame Optimization (WMFO)	Custom dataset using 2-gram and 3-gram opcodes	Accuracy: 96.93%, Sensitivity: 96.90%, Specificity: 97.92%	Uses optimized feature fusion and transformation methods for effective ransomware detection	Optimization complexity; performance might vary on real-world datasets
Denysiuk et al. (2023)	Blockchain with Proof-of-Action consensus	Hybrid	Deep learning + Blockchain consensus validation	Not specified	Accuracy: 98.81% to 99.33%	Parallel malware analysis via blockchain; Proof-of-Action validation	High computational complexity due to consensus process
Pawar et al. (2024) [90]	Blockchain for smart city authentication & data exchange	Hybrid	Two-layer XGBoost model with Bonobo optimization and greedy search	Raspberry Pi, NVIDIA Jetson, plus other IoT devices	F1-score: 93%, Accuracy: 95%	Integrates optimized XGBoost and blockchain to support digital governance security	Increased power and memory usage on edge devices (13.5% more energy)
Şafak et al. (2025) [92]	Blockchain (unspecified, used to store predictions immutably)	Hybrid	CNN-based ensemble (EfficientNetB0, MobileNetV2, Custom CNN)	CICMalDroid 2020 (3,590 image samples)	Accuracy: 97.38%	Image-based Android malware detection + blockchain logging	Does not specify blockchain type; mobile feasibility implied but not deeply tested
Nannan et al. (2025) [94]	Coordination and integrity of FL rounds	Hybrid	Federated Learning across devices	Not specified	N/R	Get hitched FL with blockchain to eliminate raw-data centralization	ledger throughput/latency and system complexity

3. Discussion

3.1 Critical Analysis

Most papers report top-line scores (like F1 or AUC) on easy, static datasets things like transaction graphs, smart-contract code, or IoT logs using random splits. That makes models look more general than they really are and hides the real issue: data changes over time in the wild (DeFi markets, mempool activity, IoT botnets). When you evaluate on real timelines and incidents, the difference between in-distribution and out-of-distribution performance shows up fast. Models that seem tied on random splits start to pull apart under time-ordered tests. On top of that, class imbalance and messy labels are everywhere: blacklists, heuristics, and weak supervision create shaky “ground truth” and quietly hurt recall at practical alert limits. Few papers share PR-AUC, calibration, or operating points tied to business needs; even fewer track latency or costs (like gas fees for on-chain checks).

Model choice also fits the data unevenly. GNNs are strong on relational fraud and contract-interaction graphs, while Transformers shine on time-based mempool patterns and code-like data (opcode sequences, logs). But cross-modal signals are rarely combined: DeFi fraud models often ignore code meaning; bytecode models rarely add transaction context; IoT IDS models skip blockchain telemetry even when a ledger is used for attestation. Robustness checks are shallow, too: simple evasion (small feature tweaks or bytecode padding), poisoning of graphs or crowd labels, and backdoors in pretrained embeddings are rarely stress-tested. Finally, deployment details are fuzzy. Teams don’t clearly document end-to-end latency (from mempool event to action), model update governance, human-in-the-loop review, or rollback for false positives. Reproducibility suffers from missing code, seeds, and fixed, time-aware splits. The real-world cases in the manuscript highlight these gaps and argue for moving beyond static benchmarks to evaluations that account for cost and latency.

3.2 Future Directions

Future work should focus on evaluations that mirror how these systems actually operate. In DeFi fraud detection, mempool analytics, smart-contract analysis, and IoT malware detection, time-ordered or rolling-window splits should replace random splits so we measure performance on evolving data. Building versioned, incident-anchored datasets linking transactions, contracts, block ranges, and logs to publicly documented attacks and responses will let researchers replay real crises and test resilience to new tactics. Results should go beyond accuracy to routinely include PR-AUC, calibration error, precision/recall at fixed alert rates, end-to-end decision latency, and a clear accounting of costs like on-chain gas, rollout TPS impact, and edge-device energy.

While within-domain multimodal fusion is already well established in the surveyed literature, these efforts remain confined to code-level or transaction-level modalities in isolation. A practical approach is late fusion or cross-attention between: (i) GNNs over wallet/contract interaction graphs; (ii) code-aware encoders over bytecode/AST/CFG to capture semantics and resist simple obfuscations; and (iii) temporal modules for mempool/DEX flow sequences. Self-supervised pretraining on large unlabeled ledgers and logs can reduce labeling needs and improve transfer to new protocols and attack families, while lightweight adapters can quickly specialize the backbone to emerging contexts.

Deployed systems must be shift-aware and continuously adaptive. Embedding-space drift detectors and change-point tests on alert streams can trigger online fine-tuning or safe fallbacks to preserve precision when adversaries pivot. Parameter-efficient updates (e.g., adapters, LoRA-style layers) and few-shot prototype mechanisms can absorb new exploit patterns with minimal downtime and compute. These updates should be guided by explicit service-level objectives, milliseconds for mempool gating versus minutes for post-trade forensics, and protected with guardrails to prevent catastrophic forgetting.

The community should also standardize security evaluations for ML pipelines. Shared red-team suites ought to test constrained evasion (feature edits that remain economically plausible), semantics-preserving bytecode transformations, and localized graph poisoning. Dataset lineage must be auditable through hashed snapshots, labeled-source provenance, and release notes that record merges and deprecations. Backdoor screening and robust-training baselines such as adversarial augmentation, consistency regularization, and confidence calibration, should be considered minimum requirements for production claims.

Deployment patterns should be hybrid by design. Heavy inference can run off-chain or at the edge, with on-chain commitments, succinct proofs, or TEE attestations providing auditability without excessive gas costs. Serving stacks should separate ultra-low-latency filters on the mempool path from slower, high-confidence analytics used for escalation and forensic review. Clear governance covering model-update approvals, incident playbooks, and reversible

mitigation tools (pause/resume guards, quarantine lists) is essential to turn model scores into accountable actions.

Meaningful adoption will also depend on privacy-preserving collaboration. Federated learning can let exchanges, wallets, and IoT operators share signals without sharing raw data. Secure aggregation and differential privacy reduce leakage, while confidential inference (using TEEs or proof-carrying computations) sustains trust when models influence on-chain policy. These techniques should be evaluated together for utility, leakage risk, and computational overhead to avoid “security theater.”

Finally, interpretability and reproducibility must be treated as deliverables, not extras. Local explanations that highlight salient edges, counterparty clusters, or opcode spans can justify decisions like blacklisting or pausing a contract, while global dashboards that track drift, failure modes, and sector-level alert distributions help operators calibrate trust. Public releases should include code, configs, fixed temporal splits, and model cards that state intended use, limits, and known failure modes. To make incident-driven research cumulative, each dataset should ship with a machine-readable event index timestamp, block heights, transaction hashes, and remediation notes so rigorous, event-aligned evaluation becomes routine.

4. Conclusion

This paper brings together deep learning and blockchain to tackle real security problems like data breaches, and reviews the latest solutions in the literature. Because blockchain is now used in sensitive areas such as finance, healthcare, and government, fixing its weaknesses is crucial. Despite helpful traits like decentralization, immutability, and transparency, blockchains can still be attacked through online fraud, smart-contract abuse, intrusions, and malware.

Deep learning helps address these risks. Using models such as CNNs, LSTMs, GNNs, and autoencoders, it improves fraud detection, intrusion prevention, smart-contract auditing, and malware detection with greater accuracy and at scale. Integrating deep learning into consensus mechanisms and privacy features also points to a smarter, more adaptive, and more resilient blockchain.

However, challenges remain. Deep learning applications are computationally intensive and suffer from adversarial attacks. The inability to explain decisions of AI, the availability of data, and privacy issues also hinder the widespread implementation. Next steps in this field should focus on interpretable models, robust decentralized learning, and energy-efficient architectures designed for blockchain applications.

Author Contributions

Conceptualization, O.M.A. and S.A.; methodology, O.M.A. and S.A.; formal analysis, O.M.A.; investigation, O.M.A.; resources, S.A.; data curation, O.M.A.; writing—original draft preparation, O.M.A.; writing—review and editing, O.M.A. and S.A.; visualization, O.M.A.; supervision, S.A.; project administration, S.A. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Conflict of Interest Statement

The authors declare no conflicts of interest.

Data Availability Statement

Dataset or special materials that need to refer to availability have not been used.

Acknowledgment

The authors gratefully acknowledge the Technical College of Zakho, Duhok Polytechnic University, and the Erbil Technical Engineering College, Erbil Polytechnic University, for the administrative and technical support provided during the preparation of this survey.

References

1. N. Z. Benisi, M. Aminian, and B. Javadi, “Blockchain-based decentralized storage networks: A survey,” *Journal of Network and Computer Applications*, vol. 162, p. 102656, 2020.
2. R. Bin Sulaiman, “Applications of Block-Chain Technology and Related Security Threats,” *Rejwan, Applications of Block-Chain Technology and Related Security Threats (June 29, 2018)*, 2018.

3. N. El Madhoun and B. Hammi, "Blockchain technology in the healthcare sector: overview and security analysis," In *2024 IEEE 14th annual computing and communication workshop and conference (CCWC)*, IEEE, pp. 439–446, 2024.
4. B. Putta and D. C. Kar, "Securing Blockchain Technology: A Comprehensive Analysis of Vulnerabilities and Mitigation Strategies," in *2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE)*, IEEE, pp. 2486–2493, 2023.
5. H. Hasanova, U. Baek, M. Shin, K. Cho, and M. Kim, "A survey on blockchain cybersecurity vulnerabilities and possible countermeasures," *International Journal of Network Management*, vol. 29, no. 2, p. e2060, 2019.
6. Alkhalifah, A. Ng, A. S. M. Kayes, J. Chowdhury, M. Alazab, and P. A. Watters, "A taxonomy of blockchain threats and vulnerabilities," in *Blockchain for Cybersecurity and Privacy*, CRC Press, London, pp. 3–28, 2020.
7. T. Varshney, N. Sharma, I. Kaushik, and B. Bhushan, "Authentication & encryption based security services in blockchain technology," in *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, IEEE, pp. 63–68, 2019.
8. T. Laurence, *Blockchain for dummies*. John Wiley & Sons, New Jersey, 2023.
9. S. Goyal, N. Sharma, I. Kaushik, and B. Bhushan, "Blockchain as a solution for security attacks in named data networking of things," In *Security and privacy issues in IoT devices and sensor networks*, Academic Press, Fribourg-Freiburg, pp. 211–243, 2021.
10. K. Sharma and D. Jain, "Consensus algorithms in blockchain technology: A survey," In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–7, 2019.
11. S. Zhou, K. Li, L. Xiao, J. Cai, W. Liang, and A. Castiglione, "A systematic review of consensus mechanisms in blockchain," *Mathematics*, vol. 11, no. 10, p. 2248, 2023.
12. S. Solat, "Rdv: An alternative to proof-of-work and a real decentralized consensus for blockchain," in *Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems*, pp. 25–31, 2018.
13. Li, X. Wei, and Z. He, "Robust proof of stake: A new consensus protocol for sustainable blockchain systems," *Sustainability*, vol. 12, no. 7, p. 2824, 2020.
14. Q. Wang, M. Xu, X. Li, and H. Qian, "Revisiting the fairness and randomness of delegated proof of stake consensus algorithm," in *2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pp. 305–312, 2020.
15. W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, and M. A. Imran, "A scalable multi-layer PBFT consensus for blockchain," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1146–1160, 2020.
16. M. M. Hammad, "Artificial Neural Network and Deep Learning: Fundamentals and Theory," *arXiv preprint arXiv:2408.16002*, 2024.
17. Goel, A. Agarwal, M. Vatsa, R. Singh, and N. Ratha, "DeepRing: Protecting deep neural network with blockchain," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2019.
18. Z. Rojan, "Financial fraud detection based on machine and deep learning: A review," *The Indonesian Journal of Computer Science*, vol. 13, no. 3, 2024.
19. A.-A. Monirah and M. Ykhlef, "DeepBlock: a Collaborative Intrusion Detection Framework Based on Blockchain and Deep Learning," in *2023 Fifth International Conference on Blockchain Computing and Applications (BCCA)*, pp. 180–185, 2023.
20. K. Demertzis, L. Iliadis, N. Tziritas, and P. Kikiras, "Anomaly detection via blockchained deep learning smart contracts in industry 4.0," *Neural Comput. Appl.*, vol. 32, no. 23, pp. 17361–17378, 2020.

21. Ancelotti and C. Liason, "Review of blockchain application with graph neural networks, graph convolutional networks and convolutional neural networks," *arXiv preprint arXiv:2410.00875*, 2024.
22. Tommaso Gagliardoni, "The Poly Network Hack Explained," Kudelski Security Research. Available at: <https://research.kudelskisecurity.com/2021/08/12/the-poly-network-hack-explained/> (Accessed on date: 3 Mar 2025)
23. K. S. Roy, M. E. Karim, and P. B. Udas, "Exploiting deep learning based classification model for detecting fraudulent schemes over ethereum blockchain," In *2022 4th International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pp. 1–6, 2022.
24. C. Ghosh, A. Chowdhury, N. Das, and B. Sadhukhan, "Enhancing financial fraud detection in bitcoin networks using ensemble deep learning," In *2023 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)*, pp. 1–6, 2023.
25. R. Olusegun and B. Yang, "Improved Ethereum Fraud Detection Mechanism with Explainable Tabular Transformer Model," In *2024 IEEE 6th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*, pp. 59–68, 2024.
26. A. Lakhan, M. A. Mohammed, D. A. Ibrahim, S. Kadry, and K. H. Abdulkareem, "ITS based on deep graph convolutional fraud detection network blockchain-enabled fog-cloud," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8399–8408, 2022.
27. S. S. Taher, S. Y. Ameen, and J. A. Ahmed, "Advanced fraud detection in blockchain transactions: An ensemble learning and explainable ai approach," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 12822–12830, 2024.
28. A. Asiri and K. Somasundaram, "Graph convolution network for fraud detection in bitcoin transactions," *Sci. Rep.*, vol. 15, no. 1, p. 11076, 2025.
29. T. Baabdullah, A. Alzahrani, D. B. Rawat, and C. Liu, "Efficiency of federated learning and blockchain in preserving privacy and enhancing the performance of credit card fraud detection (CCFD) systems," *Future Internet*, vol. 16, no. 6, p. 196, 2024.
30. Z. Gu and O. Dib, "Enhancing fraud detection in the Ethereum blockchain using ensemble learning," *PeerJ Comput. Sci.*, vol. 11, p. e2716, 2025.
31. G. Airlangga, "Deep Learning for Anomaly Detection and Fraud Analysis in Blockchain Transactions of the Open Metaverse," *Jurnal Informatika Ekonomi Bisnis*, pp. 324–329, 2024.
32. J. Sun, Y. Jia, Y. Wang, Y. Tian, and S. Zhang, "Ethereum fraud detection via joint transaction language model and graph representation learning," *Information Fusion*, vol. 120, p. 103074, 2025.
33. H. Kang and K. Hwang, "FORTRESS: Fraud-Oriented Transformer with Random Traversal for Ethereum Security Surveillance," *Inf. Sci. (N. Y.)*, p. 122534, 2025.
34. N. G. and A. J. Ferreira, "Bitcoin Fraud Detection: A Study with Dimensionality Reduction and Machine Learning Techniques," In *Proceedings of the 14th International Conference on Data Science, Technology and Applications*, 2025. DOI:10.5220/0013647400003967
35. Y. Chen, Z. Sun, Z. Gong, and D. Hao, "Improving smart contract security with contrastive learning-based vulnerability detection," In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pp. 1–11, 2024.
36. Andrew Loo, "What Was the Famous DAO Heist?", Available at: <https://corporatefinanceinstitute.com/resources/cryptocurrency/dao-heist/> (Accessed on date: 03 Mar 2025)
37. J. Li, G. Lu, Y. Gao, and F. Gao, "A smart contract vulnerability detection method based on multimodal feature fusion and deep learning," *Mathematics*, vol. 11, no. 23, p. 4823, 2023.

38. N. Li, Y. Liu, L. Li, and Y. Wang, "Smart contract vulnerability detection based on deep and cross network," In *2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA)*, pp. 533–536, 2022.
39. S. B. Osei, Z. Ma, and R. Huang, "Smart contract vulnerability detection using wide and deep neural network," *Sci. Comput. Program.*, vol. 238, p. 103172, 2024.
40. S.-J. Hwang, S. Ho Ju, and Y.-H. Choi, "CGGNet: Compiler-Guided Generation Network for Smart Contract Data Augmentation," *IEEE Access*, 12, 97515-97532, 2024.
41. D. Han, Q. Li, L. Zhang, and T. Xu, "A smart contract vulnerability detection model based on syntactic and semantic fusion learning," *Wirel. Commun. Mob. Comput.*, vol. 2023, no. 1, p. 9212269, 2023.
42. Z. Li, W. Li, X. Li, and Y. Zhang, "StateGuard: Detecting State Derailment Defects in Decentralized Exchange Smart Contract," In *Companion Proceedings of the ACM Web Conference 2024*, pp. 810–813, 2024.
43. V. K. Kasula, A. R. Yadulla, M. Yenugula, and B. Konda, "Enhancing Smart Contract Vulnerability Detection using Graph-Based Deep Learning Approaches," In *2024 International Conference on Integrated Intelligence and Communication Systems (ICIICS)*, pp. 1–6, 2024.
44. W. Deng, H. Wei, T. Huang, C. Cao, Y. Peng, and X. Hu, "Smart contract vulnerability detection based on deep learning and multimodal decision fusion," *Sensors*, vol. 23, no. 16, p. 7246, 2023.
45. T. Wang, X. Zhao, and J. Zhang, "TMF-Net: Multimodal smart contract vulnerability detection based on multiscale transformer fusion," *Information Fusion*, vol. 122, p. 103189, 2025.
46. E. Balci, T. Aydede, G. Yilmaz, and E. G. Soyak, "Examining the effectiveness of transformer-based smart contract vulnerability scan," *Journal of Systems and Software*, p. 112593, 2025.
47. N. Sharma and P. G. Shambharkar, "Multi-layered security architecture for IoMT systems: integrating dynamic key management, decentralized storage, and dependable intrusion detection framework," *International Journal of Machine Learning and Cybernetics*, vol. 16, no. 9, pp. 6399–6446, 2025.
48. Neptune Mutual, "Ethereum Classic 51% Attacks," Available at: <https://neptunemutual.com/blog/ethereum-classic-51-attacks/> (Accessed on date: 03 Mar 2025)
49. W. Serrano, "The blockchain random neural network for cybersecure IoT and 5G infrastructure in smart cities," *Journal of Network and Computer Applications*, vol. 175, p. 102909, 2021.
50. O. Alkadi, N. Moustafa, B. Turnbull, and K.-K. R. Choo, "A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9463–9472, 2020.
51. S. Alasmari, "Optimized Blockchain-Based Deep Learning Model for Cloud Intrusion Detection.," *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 9, 2024.
52. R. Wang, "Embedded architecture protocol network intrusion detection based on deep residual learning," In *Third International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI 2022)*, pp. 629–636, 2023.
53. I. Katib and M. Ragab, "Blockchain-assisted hybrid harris hawks optimization based deep DDoS attack detection in the IoT environment," *Mathematics*, vol. 11, no. 8, p. 1887, 2023.
54. R. F. Mansour, "Artificial intelligence based optimization with deep learning model for blockchain enabled intrusion detection in CPS environment," *Sci. Rep.*, vol. 12, no. 1, p. 12937, 2022.
55. M. A. Alkhonaini *et al.*, "Sandpiper optimization with hybrid deep learning model for blockchain-assisted intrusion detection in iot environment," *Alexandria Engineering Journal*, vol. 112, pp. 49–62, 2025.

56. H. Alamro *et al.*, “Modeling of blockchain assisted intrusion detection on IoT healthcare system using ant lion optimizer with hybrid deep learning,” *IEEE Access*, vol. 11, pp. 82199–82207, 2023.
57. S. K. Poorazad, C. Benzaïd, and T. Taleb, “Blockchain and deep learning-based ids for securing sdn-enabled industrial iot environments,” In *GLOBECOM 2023-2023 IEEE Global Communications Conference*, pp. 2760–2765, 2023.
58. J. Li, Q. Sun, and F. Sun, “Enhancing Privacy-Preserving Intrusion Detection in Blockchain-Based Networks with Deep Learning,” *Data Sci. J.*, vol. 22, no. 1, 2023.
59. F. F. Alruwaili, “Leveraging blockchain for cybersecurity detection using hybridization of prairie dog optimization with differential evolution on internet of things environment,” *Sci. Rep.*, vol. 15, no. 1, p. 31673, 2025.
60. A. A. Aliyu, M. Ibrahim, and S. Abdulkadir, “A Blockchain-Enhanced Deep Learning Approach for Intrusion Detection in Trusted Execution Environments,” *Digital Technologies Research and Applications*, vol. 4, no. 1, pp. 135–157, 2025.
61. Wikipedia, “Ethereum Classic,” Available at: https://en.wikipedia.org/wiki/Ethereum_Classic (Accessed on date: 3 Mar 2025)
62. Dehvcurtis, “Attacking the Heart of Blockchain: Critical Vulnerabilities in Blockchain Consensus Protocols,” Available at: <https://medium.com/%40dehvcurtis/attacking-the-heart-of-blockchain-critical-vulnerabilities-in-blockchain-consensus-protocols-f8ec04fd7f64> (Accessed on date: 3 Mar 2025)
63. H. Zhi, H. Wu, Y. Huang, C. Tian, and S. Wang, “Blockchain Consensus Scheme Based on the Proof of Distributed Deep Learning Work,” *IET Software*, vol. 2025, no. 1, p. 3378383, 2025.
64. H. He, J. Shen, Z. Cao, X. Dong, and H. Wu, “Proof of Privacy-Preserving Machine Learning: A Blockchain Consensus Mechanism with Secure Deep Learning Process,” In *2024 IEEE International Conference on Blockchain (Blockchain)*, pp. 193–200, 2024.
65. Y. Liu, Y. Lan, B. Li, C. Miao, and Z. Tian, “Proof of Learning (PoLe): Empowering neural network training with consensus building on blockchains,” *Computer Networks*, vol. 201, p. 108594, 2021.
66. T. Islam, F. H. Bappy, T. S. Zaman, M. S. I. Sajid, and M. M. A. Pritom, “MRL-PoS: A Multi-agent Reinforcement Learning based Proof of Stake Consensus Algorithm for Blockchain,” In *2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 409–413, 2024.
67. Y. Goh, J. Yun, D. Jung, and J.-M. Chung, “Secure trust-based delegated consensus for blockchain frameworks using deep reinforcement learning,” *IEEE access*, vol. 10, pp. 118498–118511, 2022.
68. S.-K. Kim, “Automotive vulnerability analysis for deep learning blockchain consensus algorithm,” *Electronics (Basel)*, vol. 11, no. 1, p. 119, 2021.
69. S. Badruddoja, R. Dantu, Y. He, M. Thompson, A. Salau, and K. Upadhyay, “Smarter contracts to predict using deep-learning algorithms,” In *2022 Fourth International Conference on Blockchain Computing and Applications (BCCA)*, pp. 280–288, 2022.
70. F. F. Alruwaili, “Ensuring data integrity in deep learning-assisted iot-cloud environments: Blockchain-assisted data edge verification with consensus algorithms,” *Aims Mathematics*, vol. 9, no. 4, pp. 8868–8884, 2024.
71. V. Amarnadh and M. N. Rao, “A Consensus Blockchain-Based Credit Risk Evaluation and Credit Data Storage Using Novel Deep Learning Approach,” *Comput. Econ.*, pp. 1–34, 2025.
72. Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, “A survey on privacy protection in blockchain system,” *Journal of network and computer applications*, vol. 126, pp. 45–58, 2019.
73. S. Rathore, B. W. Kwon, and J. H. Park, “BlockSecIoTNet: Blockchain-based decentralized security architecture for IoT network,” *Journal of Network and Computer Applications*, vol. 143, pp. 167–177, 2019.

74. A. Fadaeddini, B. Majidi, and M. Eshghi, "Privacy preserved decentralized deep learning: A blockchain based solution for secure ai-driven enterprise," In *High-Performance Computing and Big Data Analysis: Second International Congress, TopHPC 2019, Tehran, Iran, April 23–25, 2019, Revised Selected Papers 2*, pp. 32–40, 2019.
75. Z. Mahmood and V. Jusas, "Blockchain-enabled: Multi-layered security federated learning platform for preserving data privacy," *Electronics (Basel)*, vol. 11, no. 10, p. 1624, 2022.
76. X. Zhu and H. Li, "Privacy-preserving decentralized federated deep learning," In *Proceedings of the ACM Turing Award Celebration Conference-China*, pp. 33–38, 2021.
77. J. B. Awotunde, T. Gaber, L. V. N. Prasad, S. O. Folorunso, and V. L. Lalitha, "Privacy and security enhancement of smart cities using hybrid deep learning-enabled blockchain," *Scalable Computing: Practice and Experience*, vol. 24, no. 3, pp. 561–584, 2023.
78. K. Raju, N. Ramshankar, J. A. Shathik, and R. Lavanya, "Blockchain assisted cloud security and privacy preservation using hybridized encryption and deep learning mechanism in iot-healthcare application," *J. Grid Comput.*, vol. 21, no. 3, p. 45, 2023.
79. M. Kokila and K. S. Reddy, "BlockDLO: Blockchain computing with deep learning orchestration for secure data communication in IoT Environment," *IEEE Access*, vol. 12, pp. 134521–134540, 2024.
80. V. Malik *et al.*, "Building a secure platform for digital governance interoperability and data exchange using blockchain and deep learning-based frameworks," *Ieee Access*, vol. 11, pp. 70110–70131, 2023.
81. J. Li, C. Zhang, J. Zhang, and Y. Shao, "Research on Blockchain Transaction Privacy Protection Methods Based on Deep Learning," *Future Internet*, vol. 16, no. 4, p. 113, 2024.
82. M. Ghani *et al.*, "Toward robust and privacy-enhanced facial recognition: A decentralized blockchain-based approach with GANs and deep learning," *Mathematical Biosciences and Engineering*, vol. 21, no. 3, pp. 4165–4186, 2024.
83. X. Zhou and P.-C. Liao, "A privacy-preserving data storage and service framework based on deep learning and blockchain for construction workers' wearable IoT sensors," *arXiv preprint arXiv:2211.10713*, 2022.
84. B. R. Bezanjani, S. H. Ghafouri, and R. Gholamrezaei, "Privacy-preserving healthcare data in IoT: a synergistic approach with deep learning and blockchain," *J. Supercomput.*, vol. 81, no. 4, p. 533, 2025.
85. T. Bhardwaj and K. Sumangali, "An explainable federated blockchain framework with privacy-preserving AI optimization for securing healthcare data," *Sci. Rep.*, vol. 15, no. 1, p. 21799, 2025.
86. C. Yehuda Gelb, "Supply Chain Attack Using Ethereum Smart Contracts to Distribute Multi-Platform Malware–Global Security Mag Online," Available at: <https://www.globalsecuritymag.com/supply-chain-attack-using-ethereum-smart-contracts-to-distribute-multi-platform.html> (Accessed on date: 3 Mar 2025).
87. T. McGhin, K.-K. R. Choo, C. Z. Liu, and D. He, "Blockchain in healthcare applications: Research challenges and opportunities," *Journal of network and computer applications*, vol. 135, pp. 62–75, 2019.
88. A. S. Alotaibi, "Biserial Miyaguchi–Preneel blockchain-based Ruzicka-indexed deep perceptive learning for malware detection in IoMT," *sensors*, vol. 21, no. 21, p. 7119, 2021.
89. G. Nalinipriya, B. Maram, C. Vidyadhari, and R. Cristin, "Optimized deep stacked autoencoder for ransomware detection using blockchain network," *Int. J. Wavelets Multiresolut. Inf. Process.*, vol. 19, no. 06, p. 2150022, 2021.
90. P. P. Pawar, D. Kumar, M. K. Meesala, P. K. Pareek, S. R. Addula, and S. KS, "Securing Digital Governance: A Deep Learning and Blockchain Framework for Malware Detection in IoT Networks," In *2024 International Conference on Integrated Intelligence and Communication Systems (ICIICS)*, pp. 1–8, 2024.

91. A. Pastor *et al.*, “Detection of encrypted cryptomining malware connections with machine and deep learning,” *IEEE Access*, vol. 8, pp. 158036–158055, 2020.
92. E. Şafak, İ. A. Doğru, N. Barışçı, and İ. Atacak, “BlockDroid: detection of Android malware from images using lightweight convolutional neural network models with ensemble learning and blockchain for mobile devices,” *PeerJ Comput. Sci.*, vol. 11, p. e2918, 2025.
93. D. Denysiuk, O. Geidarova, M. Kapustian, S. Lysenko, and A. Sachenko, “Blockchain-based Deep Learning Algorithm for Detecting Malware.,” In *IntelITSIS*, pp. 529–538, 2023.
94. X. Nannan, M. Linyang, W. Yangfan, and M. Yubo, “Andro-BCFL: Blockchain and federated learning based Android malware detection,” *Computers and Electrical Engineering*, vol. 122, p. 109948, 2025.