Engineering Applications of Artificial Intelligence 163 (2026) 112968

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Survey paper

The role of transformer models in advancing blockchain technology: A systematic survey

Tianxu Liu [a], Yanbin Wang [b,*], Jianguo Sun [a,*], Ye Tian [a], Yanyu Huang [a], Tao Xue [a], Peiyue Li [c], Yiwei Liu [d]

[a] Hangzhou Institute of Technology, Xidian University, Hangzhou, 310000, Zhejiang, China
[b] Shenzhen MSU-BIT University, Shenzhen, 518172, China
[c] People's Public Security University of China, Beijing 100038, China
[d] Defence Industry Secrecy Examination and Certification, Beijing 100038, China

ARTICLE INFO

Keywords:
Transformers
Blockchain
Smart contract
Anomaly detection
Vulnerability detection
Cryptocurrency price prediction
Code summarization

ABSTRACT

As blockchain technology evolves, the demand for improved efficiency, security, and scalability increases, with Transformer models demonstrating significant potential to address these challenges. However, a systematic review of their blockchain applications is lacking. This paper fills this gap by surveying over 200 relevant studies, offering a comprehensive analysis of Transformer applications across four key areas: anomaly detection, smart contract vulnerability detection, cryptocurrency prediction, and code summarization. We adopt a domain-oriented classification framework that systematically organizes research progress and challenges, enhancing clarity and identifying trends. Furthermore, we offer granular sub-classification within each domain based on algorithmic types, data modalities, or information sources, delivering deeper insights into methodological advancements. Additionally, we conduct a dual-layered comparative analysis, contrasting Transformers with traditional deep learning methods and assessing variations among Transformer approaches within each domain to uncover best practices. We also explore challenges such as data privacy and model complexity, propose future research directions to tailor Transformers to blockchain-specific needs. We will continue to update the latest articles and their released source codes at https://github.com/LTX001122/Transformers-Blockchain.

1. Introduction

In the digital age, blockchain technology (Nakamoto, 2008) has emerged as one of the most revolutionary technologies (Tapscott and Tapscott, 2016), empowering various industries such as finance (Swan, 2015; McWaters et al., 2016), supply chain Kshetri (2017), Zheng et al. (2017), Xu et al. (2021), and healthcare (Mettler, 2016; Ekblaw et al., 2016; Khezr et al., 2019; Fiore et al., 2023) with its unique decentralized nature, immutability, and transparency. Since the advent of Bitcoin, various types of cryptocurrencies and smart contract platforms like Ethereum have demonstrated the extensive value of blockchain technology (Conoscenti et al., 2016; Guo and Liang, 2016; Wood et al., 2014). However, Blockchain technology generates highly complex, sequential, and dynamic data that traditional deep-learning architectures, such as CNNs and RNNs, struggle to process effectively due to their limitations in capturing long-range dependencies, handling irregular data structures, and scaling efficiently. Transformer-based models, with their self-attention mechanisms and parallel processing capabilities, offer superior performance in tasks such as anomaly detection, smart contract analysis, and cryptocurrency prediction by enabling deeper contextual understanding, improved scalability, and adaptability to evolving patterns in blockchain networks.

Since its introduction (Vaswani et al., 2017) in 2017, the Transformer model has achieved groundbreaking progress in the field of natural language processing (NLP) (Xu et al., 2023c; Chernyavskiy et al., 2021; Zhang and Shafiq, 2024). As shown in Fig. 1, research literature on transformers has significantly increased in recent years. This model, based on the self-attention mechanism, effectively handles long-range dependencies and, due to its efficient parallel computing capabilities, has been widely applied to complex sequential data tasks. Its unique structure has shown remarkable abilities in image recognition (Dosovitskiy et al., 2020), speech processing (Gulati et al., 2020), and multimodal learning (Li et al., 2021a). Given the sequential nature of blockchain data and the complex interrelations between transaction

* Corresponding authors.
E-mail addresses: 23151214343@stu.xidian.edu.cn (T. Liu), wangyanbin12@mails.ucas.ac.cn (Y. Wang), jgsun@xidian.edu.cn (J. Sun), tianye@xidian.edu.cn (Y. Tian), huangyanyu@xidian.edu.cn (Y. Huang), xuetao@xidian.edu.cn (T. Xue), lipeiyue@ppsuc.edu.cn (P. Li), yiweiliu@bit.edu.cn (Y. Liu).

https://doi.org/10.1016/j.engappai.2025.112968
Received 28 November 2024; Received in revised form 24 September 2025; Accepted 23 October 2025
Available online 30 October 2025

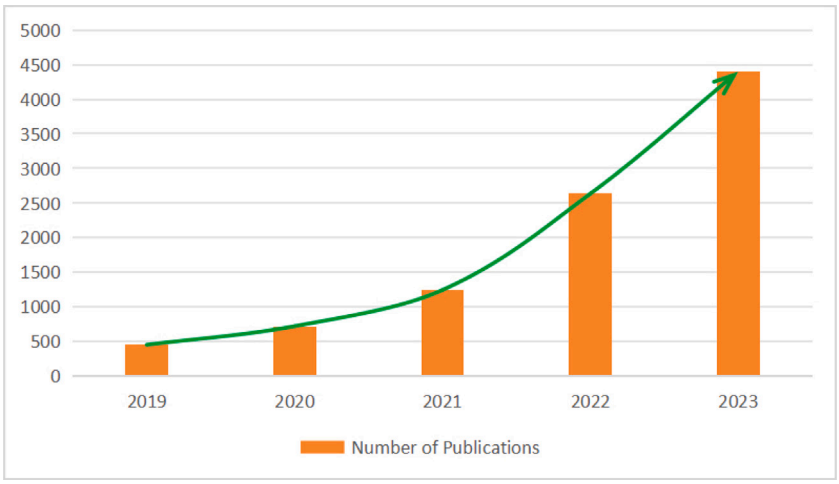

Fig. 1. Statistics on the number of research papers on transformer models.

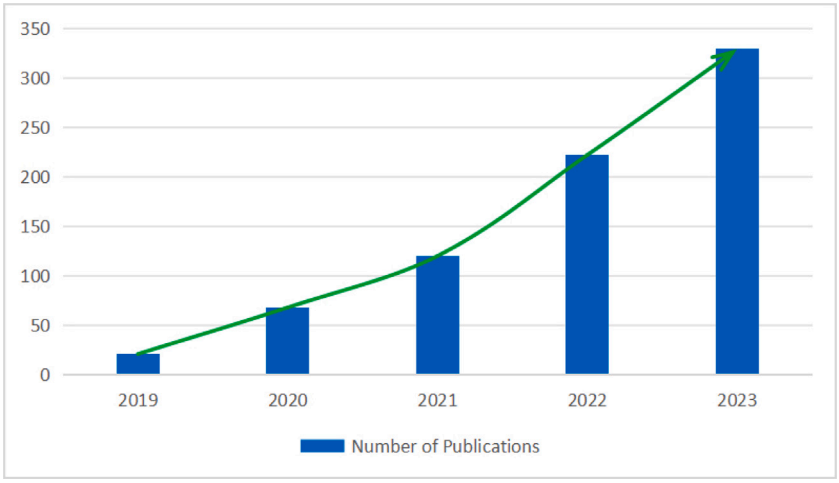

Fig. 2. Statistics on the number of research papers on the application of transformer models to blockchain.

data (Ju et al., 2020), applying Transformers to the blockchain domain, particularly in processing on-chain data, anomaly detection, and optimizing smart contracts, holds significant potential (Dolgui et al., 2020; Li et al., 2022; Weng et al., 2019; Shafay et al., 2023).

For example, in transaction monitoring and anomaly detection, traditional methods rely on simple rules or shallow machine learning models, which often fall short when dealing with complex, high-dimensional transaction data (Naseer et al., 2018; Sanjay Rai et al., 2023). The introduction of the Transformer model, with its excellent data correlation analysis capabilities, provides new solutions for identifying complex fraud patterns. As shown in Fig. 2, the research literature on the application of Transformer models to blockchain has steadily increased in recent years. Moreover, with the proliferation of smart contracts, the issue of contract security has become increasingly prominent (Luu et al., 2016; Atzei et al., 2017). Existing smart contract security analysis tools mostly rely on expert experience or traditional program analysis techniques, lacking sufficient automation and intelligence (Tikhomirov et al., 2018; Rizzo et al., 2024). The potential of the Transformer model in code semantic analysis and pattern recognition suggests promising applications in automated smart contract auditing and security analysis (Alon et al., 2019; Ahmad et al., 2021a).

In recent years, with the rapid development of blockchain technology, its applications have expanded across various fields, including finance, healthcare, and supply chains. However, as the complexity of blockchain applications increases, ensuring its security, efficiency, and scalability has become a critical challenge. Traditional blockchain data analysis methods, such as statistical approaches and shallow machine learning models, are no longer sufficient to meet the industry's demands for efficient data processing and real-time analysis. The self-attention mechanism of Transformer models offers unique advantages in handling sequential and high-dimensional data. Researchers have begun exploring the application of Transformer models in blockchain, with significant progress in areas such as transaction anomaly detection, smart contract security analysis, cryptocurrency trend prediction, and code summarization.

Blockchain technology encounters significant challenges in areas such as transaction security, anomaly detection, smart contract vulnerabilities, and cryptocurrency market prediction. Traditional AI models, including rule-based systems, recurrent neural networks (RNNs), and graph-based approaches, often struggle with long-range dependencies, high-dimensional data, and the dynamic nature of blockchain transactions. These limitations hinder the effectiveness of automated fraud detection, secure smart contract auditing, and accurate market trend forecasting. Transformer-based models, with their self-attention mechanisms, scalability, and contextual learning capabilities, offer a promising solution to address these challenges by enabling more accurate, efficient, and adaptable AI-driven blockchain applications.

Methods based on Graph Transformer Networks (GTNs) and Heterogeneous Graph Transformers (HGTs) have significantly improved fraud detection in blockchain transactions. These models can capture

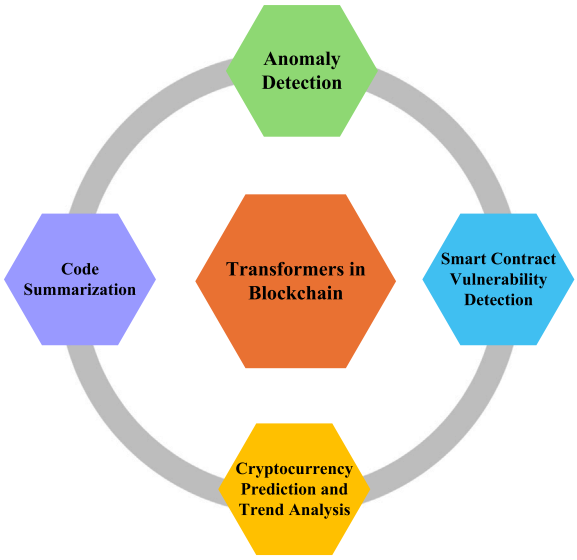

Fig. 3. A diverse set of application areas of Transformers in blockchain covered in this survey.

complex relationships in blockchain transaction graphs, enhancing the accuracy of fraud detection (Wang et al., 2023). Pretrained models like CodeBERT and GraphCodeBERT improve the accuracy and recall of smart contract vulnerability detection by understanding the deep semantic structures of smart contract code (Sun et al., 2023a). These advancements provide automated and efficient solutions for smart contract security. Recent studies indicate that combining Variational Autoencoders (VAE) with Transformers can identify anomalous behavior patterns in DeFi protocols in real-time, thus enhancing the security of the DeFi ecosystem (Song et al., 2023a). Researchers are also exploring the integration of Transformers with Zero-Knowledge Proofs (ZKP) to improve privacy-preserving transactions and enhance blockchain scalability (Ruj, 2024).

While previous surveys have discussed AI-driven blockchain advancements (Ressi et al., 2024), limited work has systematically examined the specific role of Transformer-based models in solving blockchain-related challenges. This paper provides a comprehensive review of Transformer applications, classifying them into key domains and analyzing their effectiveness in addressing blockchain's unique computational and security challenges. Based on the aforementioned background and research motivations, the primary objective of this paper is to comprehensively review the practices, challenges, and future directions of applying the Transformer model in blockchain technology. Specifically, this paper aims to:

1. Systematically introduce the Transformer model: Explain its fundamental principles, architectural features, and why it is effective in processing blockchain data.

2. Review the model's specific applications in blockchain: As shown in Fig. 3, conduct an in-depth analysis of the applications and research achievements of Transformers in blockchain transaction anomaly detection, smart contract vulnerability detection, cryptocurrency prediction and trend analysis, and generating code summaries.

3. Explore challenges and future directions: Identify the current challenges in these applications, discuss how to overcome them, and explore potential future research directions.

Through these efforts, the contributions of this paper are:

- conducting a systematic survey of more than 200 relevant studies to provide a comprehensive overview of Transformer model applications in blockchain, helping researchers better understand both the potential and limitations of this technology.

- Introducing a novel domain-oriented classification framework that clearly identifies major blockchain application domains enhanced by Transformer architectures, effectively promoting the integration of blockchain and artificial intelligence technologies, thus opening new avenues for research and practical application.

- Providing an in-depth discussion of existing challenges and offering concrete directions for future research, particularly focusing on enhancing blockchain data processing efficiency and addressing critical issues such as security, data privacy, computational complexity, and real-time processing constraints.

To ensure methodological rigor and transparency, this survey strictly follows the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) framework, which enhances reproducibility and reliability of the review process. Chapter 2 provides an overview of the Transformer model, introducing its architecture and its various derivatives, and analyzing their advantages and challenges. Chapter 3 introduces the fundamentals of blockchain technology. Chapter 4, through a summary and analysis of existing literature, details the current research progress and applications of Transformers in the blockchain domain. Chapter 5 discusses the challenges and future prospects of applying Transformers to blockchain. Finally, the paper concludes with a summary.

2. Overview of the transformer model and its derivatives

The Transformer model, introduced by Vaswani et al. (2017), is a groundbreaking advancement in deep learning. Unlike traditional Recurrent Neural Networks (RNNs) (Elman, 1990), it relies entirely on an attention mechanism (Bahdanau et al., 2016) to process sequential data. Its self-attention and multi-head attention mechanisms excel at parallelizing long sequences and capturing long-range dependencies, making it a cornerstone for complex sequence tasks in natural language processing (NLP) and beyond.

Table 1 presents a comparative analysis of Transformer models against other commonly used AI architectures, including RNNs, CNNs, and GNNs. The comparison highlights key factors such as handling sequential dependencies, scalability, computational efficiency, interpretability, and suitability for blockchain-related tasks. As shown, Transformers excel in modeling long-range dependencies and scalability due to their self-attention mechanisms, making them highly effective for blockchain applications such as smart contract analysis, anomaly detection, and NLP tasks. However, their higher computational cost and lower interpretability remain challenges compared to other architectures.

2.1. Attention mechanism and positional encoding

The self-attention mechanism computes relationships between sequence elements through three vectors: Query, Key, and Value. For each element, the model (1) calculates compatibility scores between its Query and all Keys, (2) uses these scores to weight the corresponding Values, and (3) sums the weighted Values to produce the output representation. Multi-head attention extends this process by employing multiple parallel attention heads, each learning distinct representation subspaces to capture diverse features and patterns. The working principle of the multi-head attention mechanism is shown in Fig. 4.

To compensate for the lack of recurrent structure, positional encoding injects sequential order information into the model. This is achieved by adding fixed sinusoidal signals to the input embeddings, where different frequencies of sine and cosine functions generate unique position representations. The chosen frequency progression allows the model to naturally learn both absolute positions and relative distances between elements in the sequence.

2.2. Encoder–decoder architecture

The detailed working principle of the Transformer model is shown in Fig. 5. The Transformer architecture consists of two core components: (1) The encoder stack contains N identical layers, each with two sub-layers: (i) multi-head self-attention that processes input sequences through parallel attention heads, and (ii) position-wise feed-forward

Table 1
Comparison of Transformers with other AI models for blockchain applications.

Criteria	RNNs	CNNs	GNNs	Transformers
Sequential Dependencies	Handles short sequences but struggles with long-range dependencies	Not suited for sequential data	Captures graph structures but weak in sequential modeling	Excellent at modeling long-range dependencies with self-attention
Scalability	Limited due to sequential processing	Highly scalable with parallelization	Scalable but requires graph preprocessing	Highly scalable with parallel processing
Computational Efficiency	Moderate but slow for long sequences	Efficient for feature extraction	Moderate, depends on graph complexity	Computationally expensive but efficient with optimizations
Interpretability	Low interpretability	Moderate interpretability	More interpretable due to graph structures	Less interpretable due to complex attention mechanisms
Suitability for Blockchain Tasks	Used in transaction prediction but limited for complex tasks	Effective for blockchain image analysis but lacks sequential capabilities	Useful for blockchain network analysis but limited for NLP	Strong for blockchain tasks like smart contract analysis, anomaly detection, and NLP

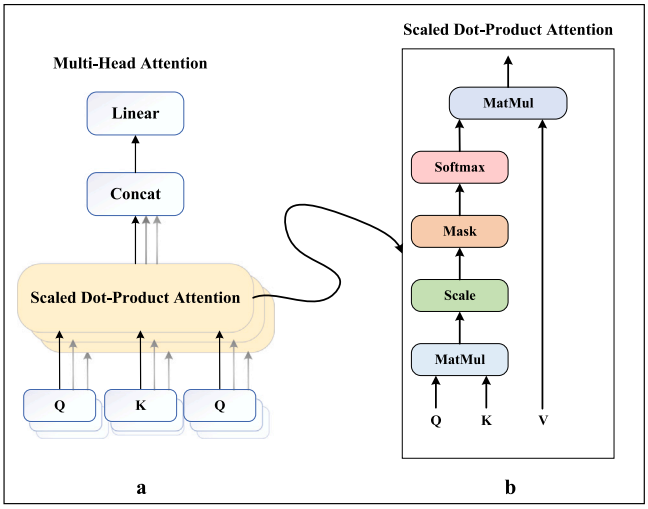

Fig. 4. Multi head attention mechanism. In the encoder and decoder, multiple attention heads are stacked together and their outputs are concatenated.

networks (FFN) with two linear transformations; (2) The decoder extends this design with three enhanced sub-layers: (i) masked multi-head attention preventing future information leakage, (ii) encoder–decoder cross-attention for output–input alignment, and (iii) position-wise FFN identical to the encoder's; (3) Both components employ identical stabilization mechanisms: residual connections with layer normalization after each sub-layer, effectively addressing vanishing gradients in deep networks while maintaining stable training dynamics.

2.3. Derivative models based on the transformer

The Transformer's superior performance and flexibility have spurred numerous derivative models that excel in NLP and beyond. By adapting the original architecture or training process, these variants optimize task-specific efficiency and effectiveness. This subsection briefly introduces key derivatives — including BERT (Devlin et al., 2019), GPT (Radford et al., 2018), RoBERTa (Liu et al., 2019), and T5 (Raffel et al., 2023) — analyzing their architectural innovations and potential applications in blockchain, particularly for smart contract analysis and transaction data processing.

BERT (Devlin et al., 2019) employs a stack of Transformer encoders with multi-head self-attention and feed-forward networks (Fig. 6). Its bidirectional pretraining through Masked Language Modeling (MLM) enables deep contextual understanding, making it particularly effective for analyzing smart contracts where code semantics depend on surrounding context. The model's ability to capture relationships across entire code segments helps identify vulnerabilities and optimize contract logic.

GPT (Radford et al., 2018) utilizes a unidirectional Transformer decoder architecture (Fig. 7), optimized for autoregressive text generation. Its two-phase training process (unsupervised pretraining followed by task-specific fine-tuning) enables applications like automated smart contract generation and natural language explanations of blockchain transactions. The model's causal attention mask ensures proper sequence generation while maintaining context awareness.

RoBERTa (Liu et al., 2019) enhances BERT through optimized training strategies (Fig. 8), including larger batch sizes, extended training duration, and removal of the Next Sentence Prediction task. These improvements yield superior performance for analyzing blockchain transaction patterns and monitoring cryptocurrency-related social media communications, where precise language understanding is crucial.

T5 (Raffel et al., 2023) implements a unified text-to-text framework using complete Transformer encoder–decoder architecture. Its flexible approach supports diverse blockchain applications including automated audit report generation, simplification of complex transaction data into natural language, and maintenance of smart contract documentation. The model's task-agnostic design allows seamless adaptation to various NLP requirements in blockchain ecosystems.

3. Fundamentals of blockchain technology

Blockchain technology, introduced by Satoshi Nakamoto in 2008 for Bitcoin, is a decentralized digital currency system. It enables distributed data storage and management across a global network of nodes without centralized control. Blockchain relies on distributed ledger technology, where transactions are grouped into "blocks," each cryptographically linked to the previous, forming an immutable chain. This ensures data security and permanence.

Transaction Mechanism: One of the core features of blockchain is its transaction mechanism (Antonopoulos and Harding, 2023). Each transaction must be verified by nodes in the network using a consensus algorithm. The verification process includes confirming the validity of the transaction and packaging it into a new block (Li et al., 2017; Regnath and Steinhorst, 2018; Nasrulin et al., 2018; Li et al., 2021b). Once a block is accepted by the network, it is added to the blockchain, a process that involves linking the hash of the new block to the hash of the previous block, ensuring the irreversibility of the chain and the immutability of the data. The core of this process is data processing and synchronization, where the application of Transformer can improve the speed and efficiency of data processing, especially in scenarios involving a large number of transactions and data verification (Jayabalan and N., 2021).

Smart Contracts: Smart contracts are self-executing codes stored on a blockchain that automatically enforce contract terms when predefined conditions are met, eliminating intermediaries, reducing costs, and minimizing errors (Cong and He, 2019; Kim and Ryu, 2020; Wang

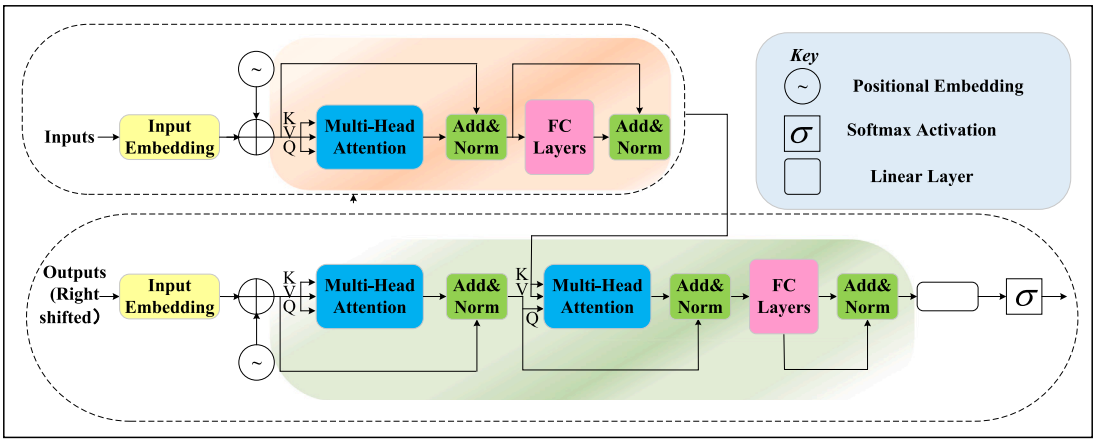

Fig. 5. Architecture of the Transformer model.

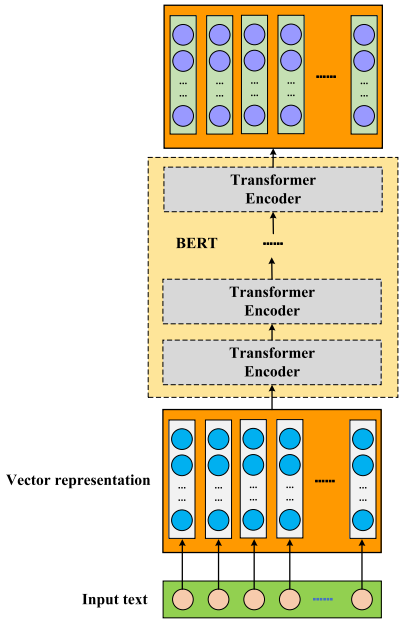

Fig. 6. Architecture of the BERT model.

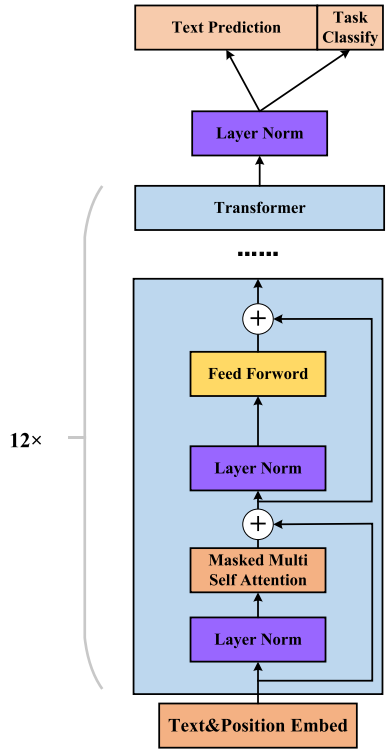

Fig. 7. Structure of the GPT-3.5 model.

et al., 2022; Alikhani and Hamidi, 2021). They are widely applied in financial transactions, supply chain management, automated legal processes, and identity verification (Prause, 2019; Dolgui et al., 2020; Trautmann and Lasch, 2020; Aejas and Bouras, 2021). Using Transformers for smart contract analysis can predict behavior, optimize complex logic, and enhance execution through natural language processing and pattern recognition (Charlier et al., 2017; Kushwaha et al., 2022; Zhou et al., 2023; Dosovitskiy et al., 2020; Amin and Neumann, 2021; Gabriel et al., 2023; Liang et al., 2023). Fig. 9 illustrates their working mechanism.

Cryptocurrencies: Cryptocurrency is a digital currency leveraging blockchain technology, employing encryption to secure transactions, prevent fraud, and eliminate double spending. It revolutionizes money storage and cross-border transactions, making them faster and more cost-effective (Yu et al., 2022; Ghosh et al., 2020). Cryptocurrencies introduce new investment tools and value storage methods to global financial markets (Baur et al., 2018; Corbet et al., 2019). Applying Transformer models optimizes trading algorithms, improves market prediction accuracy, and enhances cryptocurrency trading and market analysis (Sanju et al., 2023).

4. Related work

This section surveys pre-Transformer research in four major blockchain applications. For each domain, we (1) summarize the background, objectives, and traditional methods, (2) identify shared limitations of these methods relative to Transformers, and (3) highlight how Transformers offer superior solutions.

4.1. Anomaly detection

4.1.1. Background and objectives

Anomaly detection in blockchain identifies deviations from normal behavior to enhance network security and reliability. It uses machine learning and data analysis to monitor patterns, automatically detecting threats like double spending or 51% attacks (Signorini et al., 2018b)

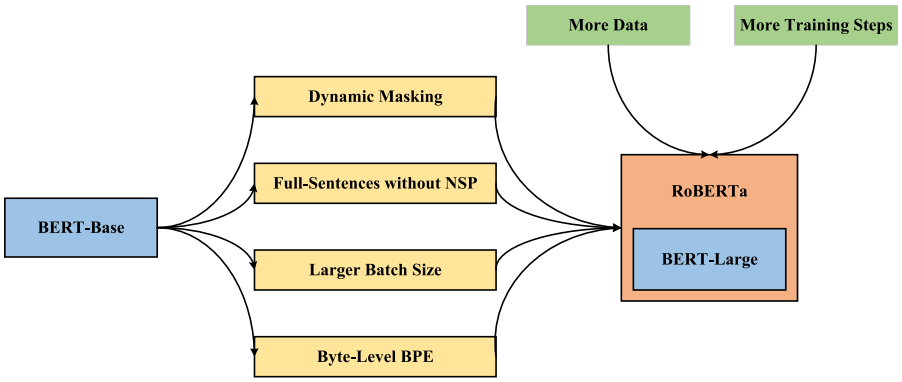

Fig. 8. RoBERTa's architectural improvements over BERT.

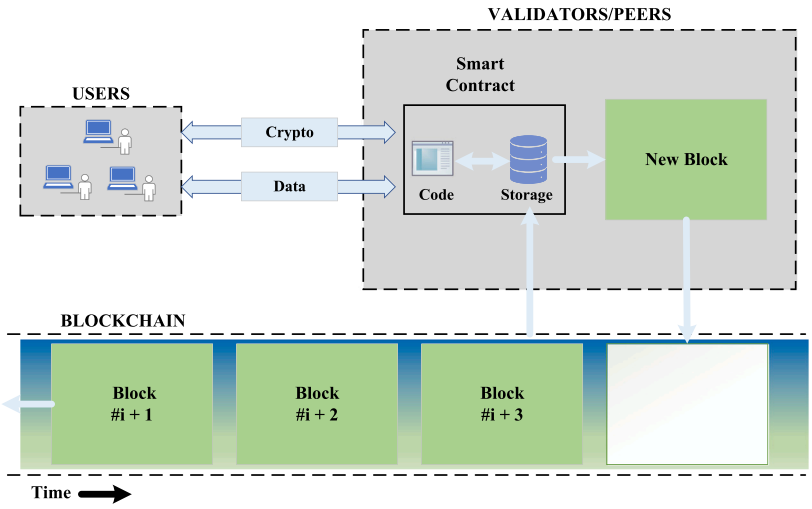

Fig. 9. Smart contract execution workflow.

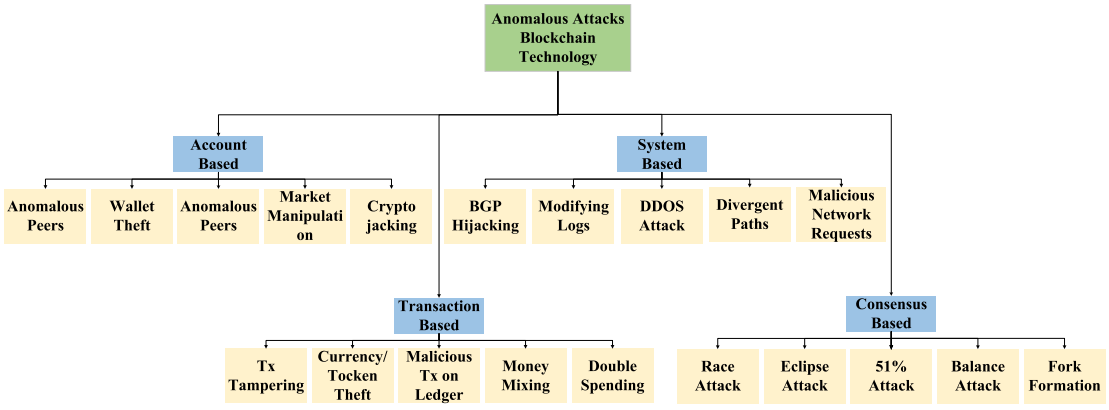

Fig. 10. Classification of anomalous attacks on blockchain technology.

and preventing fraud in cryptocurrency transactions by spotting unusual activities (Shayegan et al., 2022). This ensures data integrity and minimizes downtime from attacks, improving operational efficiency (Ahmad et al., 2021b; Kim et al., 2021b). A taxonomy of anomalies is shown in Fig. 10. By proactively addressing risks, anomaly detection protects user assets, prevents financial losses, and supports sustainable blockchain growth.

4.1.2. Applications of traditional methods

Traditional methods in blockchain anomaly detection employ diverse techniques to identify and mitigate security threats. For instance, metadata analysis, as applied in the "BAD" and "ADvISE" projects (Signorini et al., 2018b,a), leverages blockchain's immutability and transparency to detect abnormal activities (Liang et al., 2021). Network traffic monitoring, explored in Kim et al. (2021b), enhances real-time detection accuracy by capturing subtle anomalies in complex data flows. In Sayadi et al. (2019), One-Class Support Vector Machines (OCSVM) and K-means clustering were used to detect anomalies in cryptocurrency transactions. Federated and unsupervised learning, as studied in Saravanan et al. (2023), effectively protect data privacy and handle unlabeled data in blockchain environments. Additionally,

Table 2
Classification of vulnerabilities in Ethereum smart contracts.

Vulnerability source	Vulnerability type
Solidity Code	Call to unknown address Gasless Send Reentrancy attack Exception disorders Type Casts
Blockchain	Generating randomness Unpredictable state Time Constraints
EVM (Ethereum Virtual Machine)	Immutable bugs Ether lost in the transfer Stack size limit

GPU acceleration, demonstrated in Morishima (2021), optimizes performance for large-scale blockchain applications requiring real-time processing.

Structural and topological analyses provide further insights. Topological Data Analysis (TDA), applied in Ofori-Boateng et al. (2021), identifies structural anomalies in dynamic multilayer blockchain networks, effectively capturing unconventional behaviors caused by attacks or misconfigurations (Idé, 2018). Additionally, "sketch" techniques — a data simplification method — introduced in Voronov et al. (2021), enhance scalability and efficiency by reducing dataset complexity while preserving critical structural information.

4.2. Smart contract vulnerability detection

4.2.1. Background and objectives

Smart contracts, a core component of blockchain technology, enable the automatic execution of coded terms, enhancing transaction efficiency and transparency while ensuring trust and compliance without intermediaries. However, they face security challenges, with vulnerabilities like reentrancy attacks (He et al., 2023), arithmetic overflow (Chu et al., 2023), timestamp dependence (Luo et al., 2024), and improper exception handling (Vani et al., 2022) risking funds theft, data corruption, or logic failure, leading to significant economic losses and trust issues.

Efficient vulnerability detection is vital to mitigate financial risks, protect user assets (Yang et al., 2022a), ensure blockchain ecosystem stability, meet regulatory compliance (Tang et al., 2021), and boost market acceptance and user trust (Wang and Xu, 2023). Modern techniques, such as deep learning and graph neural networks (Liu et al., 2021), improve detection accuracy and efficiency, supporting blockchain's healthy development. Fig. 11 shows the growing research on vulnerability detection, and Table 2 classifies common smart contract vulnerabilities.

4.2.2. Applications and limitations of traditional methods

See Figs. 12–14.

4.2.3. Applications of traditional methods

Traditional methods for smart contract vulnerability detection, excluding Transformer models, employ various machine learning techniques, each with distinct strengths and limitations. These methods can be categorized into structural, behavioral, and hybrid approaches.

Structural Analysis: Graph Neural Networks (GNNs) and Abstract Syntax Trees (ASTs) are commonly used to analyze code structure. Refs. (Luo et al., 2024) and Han et al. (2022) apply GNNs to capture complex relationships in smart contracts, with (Luo et al., 2024) focusing on critical structures via attention mechanisms and Han et al. (2022) targeting vulnerabilities in execution paths using control flow graphs. Additionally, Yang et al. (2022b) leverages ASTs (illustrated in Fig. 12) to detect structural vulnerabilities by representing code as a tree, enabling intuitive and effective analysis. Semantic analysis in Yan et al. (2022) further complements these methods by examining the functional intent of code, enhancing the depth of vulnerability detection.

Behavioral Analysis: Dynamic behavior analysis focuses on runtime vulnerabilities. Ref. (Sui et al., 2023) identifies issues by analyzing opcodes during execution, while (Cao et al., 2023) monitors real-time execution using data flow and attention mechanisms, improving dynamic vulnerability detection. Deep learning methods also contribute: (Zhang et al., 2022a) employs Bi-LSTM (process shown in Fig. 13) to identify complex dependencies in time-series data, and Deng et al. (2023) uses multimodal decision fusion (process in Fig. 14) to enhance predictions by integrating multiple data sources.

Hybrid and Optimization Techniques: Hybrid approaches combine multiple methods for improved detection. Ref. (Wu et al., 2023) introduces a model with a hybrid attention mechanism, merging self-attention and convolutional attention to focus on long-distance dependencies and local features, boosting accuracy in complex code structures. Additionally, Yang and Zhu (2023) integrates Support Vector Machines (SVM) with deep learning to optimize the detection process, providing robust classification and prediction capabilities.

4.3. Cryptocurrency prediction and trend analysis

4.3.1. Background and objectives

Cryptocurrency prediction and trend analysis employ methods like technical, fundamental, and sentiment analysis to forecast price movements and market dynamics by analyzing historical data, supply–demand relationships, global economic events, and investor behavior. Research in this field has grown steadily, as shown in Fig. 15, reflecting its increasing importance. The cryptocurrency market is characterized by extreme volatility, influenced by factors such as market sentiment, policy changes, and global economic events (Bariviera, 2017), which complicates accurate prediction. Additionally, blockchain innovations like smart contracts and DeFi, as explored in Cong and He (2019), introduce new trading mechanisms that impact market prices and participant behavior. Regulatory variations across regions, discussed in Armour et al. (2016), further affect market stability and trading patterns.

Understanding the behavior of diverse market participants — retail and institutional investors, miners, and developers — is essential for effective trend analysis (Catalini and Gans, 2020). For instance, Fig. 16 provides a 2024–2030 price prediction for Flow coin, aiding investment decisions. These analyses help investors and analysts better navigate market dynamics, optimize investment strategies, and manage risks.

4.3.2. Applications of traditional methods

Prior methods for cryptocurrency prediction can be categorized into statistical, machine learning, deep learning, sentiment-based, and multi-model approaches, each addressing the market's volatility and complexity.

Statistical Methods: Traditional statistical techniques like time series analysis and regression models predict price trends using historical data. Methods such as Autoregressive (AR), Moving Average (MA), ARMA, and ARIMA models are widely applied. For instance, Yang et al. (2019) evaluates ARIMA's performance in cryptocurrency price prediction, noting its limitations with volatile data. Moving averages, explored in Pilipchenko et al. (2021) and Pronchakov and Bugaienko (2019), help identify trends and reversal points, while (Catania et al., 2019) enhances adaptability using Dynamic Model Averaging (DMA) to integrate multiple time series models.

Machine Learning Methods: Machine learning algorithms capture nonlinear patterns in dynamic markets. Fig. 17 illustrates the process: data collection, visualization, and model training. Studies like (Mittal et al., 2018) apply linear regression and support vector machines,

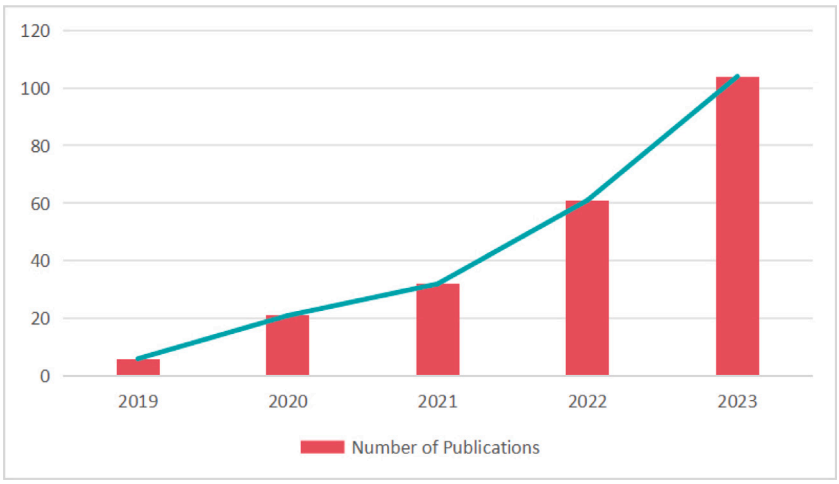

Fig. 11. Statistics on the publication of papers on smart contract vulnerability detection.

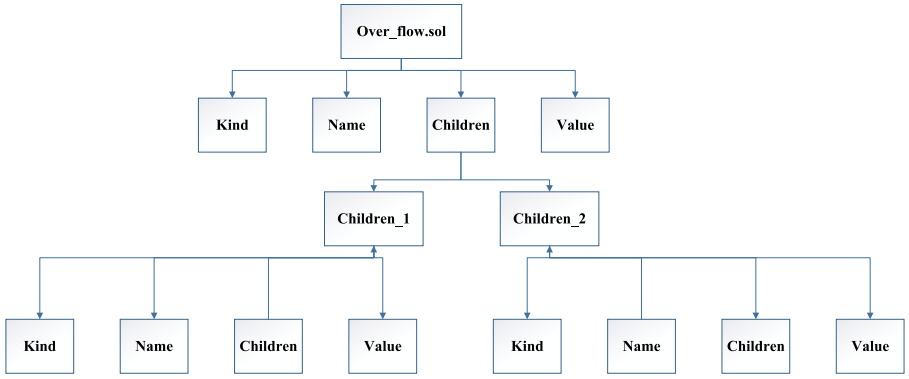

Fig. 12. The graph of AST.

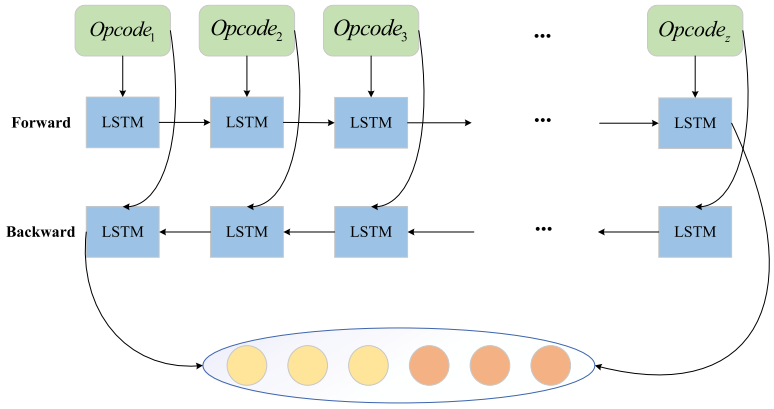

Fig. 13. Bi-LSTM data processing process.

while (Sun et al., 2020) uses LightGBM for trend prediction via ensemble learning. Stochastic neural networks in Jay et al. (2020) model market uncertainty, and Alamery (2023) compares algorithms like random forests and gradient boosting, guiding model selection. Additionally, Abdul Rashid and Ismail (2023) analyzes linear and nonlinear patterns for trend forecasting.

Deep Learning Techniques: Deep learning excels in handling large-scale, nonlinear datasets. LSTM models, as in Patel et al. (2020) and Lahmiri and Bekiros (2019), capture long-term dependencies in volatile markets, with (Gunarto et al., 2023) confirming LSTM's superiority over traditional RNNs. GRU, a simpler LSTM variant, is explored in Kim et al. (2021a) (structure in Fig. 18), showing comparable performance with faster training. Hybrid models like LSTM-GRU in Liu et al. (2023) and CNN for time series in Kim et al. (2022) further improve accuracy. However, deep learning faces challenges: reliance on large, high-quality data, high computational costs, and overfitting risks, especially with noisy cryptocurrency data.

Sentiment Analysis: Cryptocurrency markets are heavily influenced by public sentiment. Social media sentiment analysis, using platforms like Twitter and Google Trends, predicts short-term price changes. Fig. 19 outlines a sentiment-based prediction model. Studies such as (Steinert and Herff, 2018), Inamdar et al. (2019), and Wołk (2020)

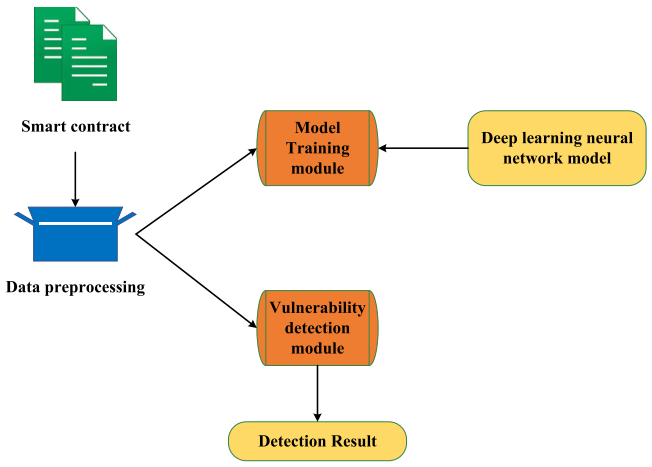

Fig. 14. Smart contract vulnerability detection frameworks based on deep learning.

show sentiment's impact on price volatility, particularly for altcoins and Bitcoin. Advanced methods in Pathak and Kakkar (2020), Prajapati (2020), Oikonomopoulos et al. (2022), Koltun and Yamshchikov (2023), and Bhatt et al. (2023) integrate sentiment with market data, achieving high accuracy (e.g., 99.67% for Ethereum in Oikonomopoulos et al. (2022)). Limitations include data quality, sentiment diversity, and market manipulation risks.

Multi-Model Approaches: Combining multiple models enhances prediction robustness. Chalkiadakis et al. (2022) uses multi-output Gaussian processes to link sentiment and prices, while (Derbentsev et al., 2020) applies Random Forest and Gradient Boosting for short-term forecasts. Hybrid models like (Du et al., 2022) (using VMD and CEEMDAN) and Boukhers et al. (2022) (AdaBoost-LSTM with multimodal data) improve accuracy by 19.29% over benchmarks. Kim et al. (2022) leverages on-chain data with SAM-LSTM, and Akila et al. (2023) combines LSTM with Change Point Detection, outperforming baseline models in MSE, MAE, and RMSE. These approaches integrate diverse data sources, enhancing adaptability to market volatility.

4.4. Code summarization

4.4.1. Background and objectives

Smart contracts are self-executing programs on a blockchain that automate contract terms, digital asset transactions, and operations, characterized by transparency, trustworthiness, and immutability. As shown in Fig. 20, research on smart contract code summarization has increased annually, reflecting growing interest. However, these features make security and accuracy critical—immutable code means vulnerabilities or unclear logic can cause irreversible losses (Bhargavan et al., 2016). With smart contracts widely applied in finance, insurance, and supply chains, the need for security and auditability has surged.

Smart contract code summaries, brief descriptions of functionality, logic, permissions, and risks, help users, developers, auditors, and regulators quickly understand contracts, reducing risks from misinterpretation (Grech et al., 2018; Liu et al., 2018). They also enhance audit efficiency and compliance checks (Bartoletti and Pompianu, 2017). Research on summarization splits into manual and automatic approaches (Delmolino et al., 2016). Manual methods, where developers write summaries, are accurate but time-consuming and subjective, unfit for large-scale use (Grech et al., 2018). Automatic methods, like NatSpec and SmartSummary, use static and dynamic analysis to extract key information and generate concise, human-readable summaries (Nikolić et al., 2018).

4.4.2. Applications of traditional methods

Smart contract code summarization methods can be categorized into manual, template-based, and NLP-based approaches, each aiming to enhance transparency, readability, and security while addressing scalability challenges.

Manual Summarization: Manual methods involve developers or auditors writing summaries, often using standardized annotations like NatSpec to describe contract functionality, permissions, and logic (Grech et al., 2018). While accurate, as demonstrated by Grech et al. (2018) with the MadMax tool, this approach is time-consuming and lacks scalability for large contract volumes. To address complex vulnerabilities like "honey pot" attacks, Yu et al. (2019) proposed static analysis to generate protective summaries, highlighting the inefficiency of manual methods in such scenarios.

Template-Based Summarization: Template-based methods use predefined structures to generate standardized summaries. Bartoletti and Pompianu (2017) analyzed 870 Ethereum contracts, finding that 70% of functionalities could be extracted using templates based on common design patterns. Nikolić et al. (2018) extended this with NatSpec, extracting functional and permission details to create secure summaries. Mendoza and Gu (2018) further refined the approach by incorporating execution graphs to generate behavioral summaries, providing structured insights into contract behavior.

NLP-Based Summarization: NLP methods automate summary generation through feature extraction and semantic analysis. Grech et al. (2018) introduced MadMax, which uses control flow analysis to identify gas consumption issues and generate summaries. Mou et al. (2016) developed a deep learning model with convolutional neural networks, combining code structure and semantics for improved accuracy. Xu et al. (2020) proposed an intelligent summarization method using semantic understanding and natural language generation, while (Hu et al., 2018b) aligned code descriptions with function behavior, enhancing summary precision through NLP techniques.

4.5. Limitations of traditional methods across blockchain applications

Traditional methods in blockchain applications — spanning anomaly detection, smart contract vulnerability detection, cryptocurrency prediction, and code summarization — rely on techniques like machine learning, network monitoring, data mining, topological analysis, and rule-based approaches. While effective, they exhibit shared limitations compared to Transformer models, which can be grouped into two primary categories: limited handling of complex dependencies and semantics and constrained processing efficiency and adaptability.

Limited Handling of Complex Dependencies and Semantics: Traditional methods struggle to capture long-term dependencies, global context, and deep semantics in complex blockchain data. In anomaly detection, methods like OCSVM, K-means (Sayadi et al., 2019), and topological analysis (Ofori-Boateng et al., 2021) handle simple patterns but fail to address long-range dependencies or contextual relationships in transaction data. Similarly, in cryptocurrency prediction, LSTM and RNN models (Patel et al., 2020) are less effective at capturing long-term dependencies and global market dynamics compared to Transformers. For smart contract vulnerability detection, Bi-LSTM (Zhang et al., 2022a) and graph-based approaches (Luo et al., 2024; Han et al., 2022) lose information over long sequences and lack global context understanding. In code summarization, rule-based and shallow machine learning methods (Bartoletti and Pompianu, 2017) cannot capture deep logical relationships or complex semantics, often missing critical code interactions. In contrast, Transformers leverage self-attention to globally capture dependencies, semantics, and context across long sequences, enabling better recognition of intricate patterns, fraud, vulnerabilities, and code logic.

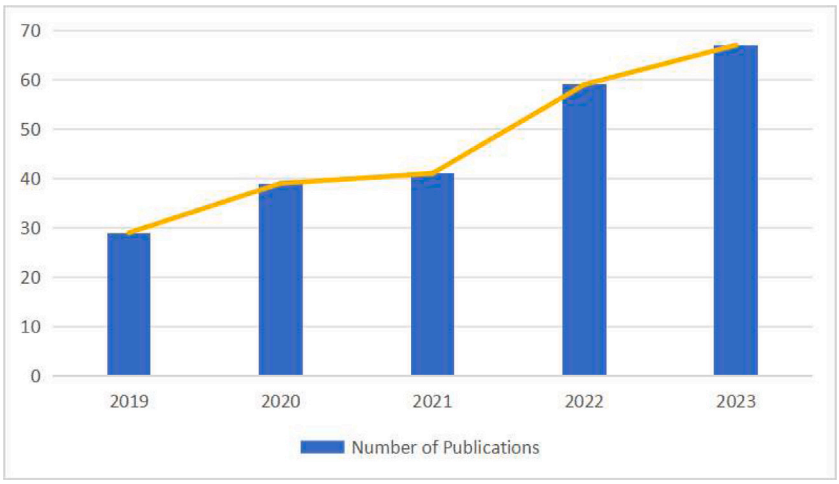

Fig. 15. Statistics on the publication of papers on Cryptocurrency Prediction.

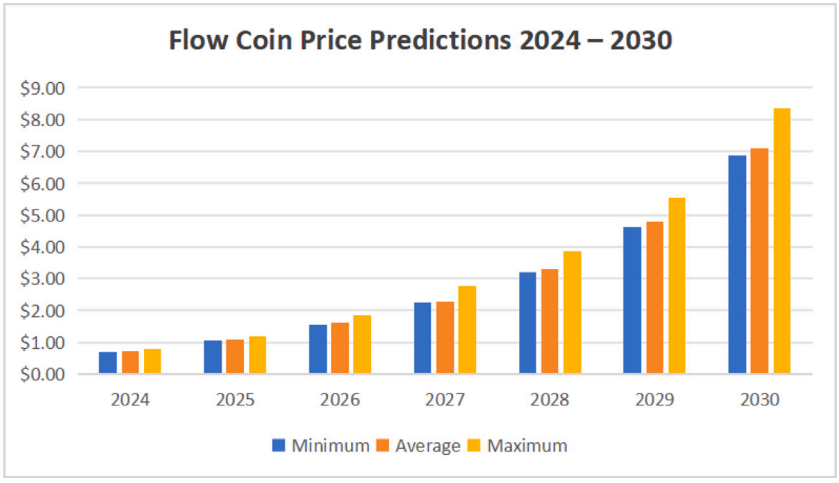

Fig. 16. Flow coin price predictions 2024–2030.

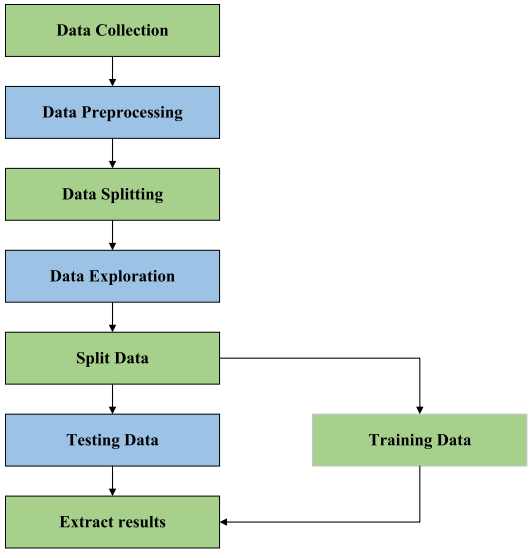

Fig. 17. Methodology of processing data and model selection.

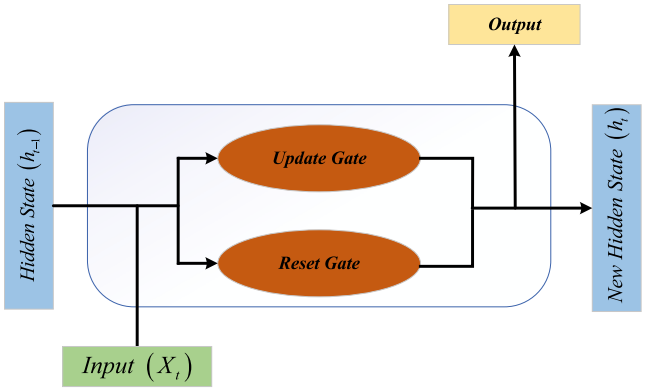

Fig. 18. The cell model of a GRU block diagram.

Constrained Processing Efficiency and generalization: Traditional methods face challenges in scalability, real-time processing, and adaptability to new scenarios. In anomaly detection, network traffic monitoring (Kim et al., 2021b) and real-time execution analysis (Sui et al., 2023; Cao et al., 2023) encounter bottlenecks with large-scale data, while in cryptocurrency prediction, RNN-based models lack parallel

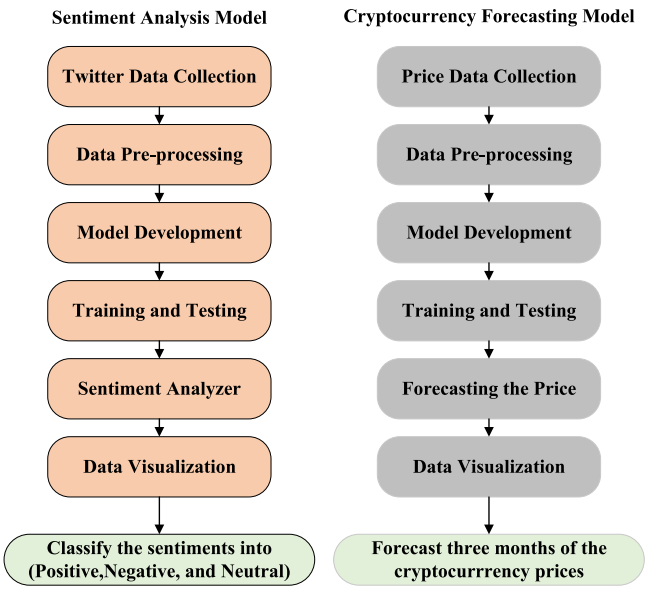

Fig. 19. Forecasting Model Structure.

processing capabilities, limiting efficiency. Smart contract vulnerability detection methods like hybrid attention models (Wu et al., 2023) and SVM (Yang and Zhu, 2023) struggle to adapt to new attack patterns and scale poorly. Similarly, code summarization methods reliant on manual features or templates (Grech et al., 2018) exhibit limited generalization across diverse codebases. Transformers address these issues with parallel processing, efficient handling of large-scale datasets, and adaptability through fine-tuning, making them ideal for real-time security monitoring, dynamic market prediction, and scalable code analysis.

4.6. Opportunities with transformers

Transformers address these limitations by leveraging their self-attention mechanism and multi-layer architecture, offering significant improvements across blockchain applications:

Enhanced Handling of Complex Dependencies and Semantics: Transformers globally capture long-term dependencies, semantics, and context across long sequences, enabling better recognition of intricate patterns, fraud, vulnerabilities, and code logic. This improves accuracy in anomaly detection (e.g., identifying complex transaction patterns), cryptocurrency prediction (e.g., capturing global market dynamics), smart contract vulnerability detection (e.g., understanding complex code interactions), and code summarization (e.g., reflecting deep logical relationships).

Improved Processing Efficiency and Adaptability: Transformers' parallel processing capabilities and scalability allow efficient handling of large-scale datasets, benefiting real-time security monitoring, dynamic market prediction, and scalable code analysis. Their adaptability through fine-tuning ensures flexibility in responding to new attack patterns, market trends, or diverse codebases, making them a promising solution for addressing the complexity and volatility of blockchain applications.

5. Methodology

This survey paper follows a structured and systematic methodology to ensure a comprehensive review of the literature related to the application of Transformer models in blockchain technology. The methodology involves multiple steps, including literature search, data collection, analysis, and integration of findings.

5.1. Literature search

To ensure methodological rigor and transparency, we conducted a systematic literature review based on PRISMA guidelines. The search was carried out between March 1st and November 30th, 2023, across six major academic databases: Google Scholar, IEEE Xplore, SpringerLink, ScienceDirect, ACM Digital Library, and Web of Science. We used a series of targeted Boolean search queries — such as "Transformer" AND "Blockchain", "Transformer model" AND "Smart contract", and "Transformer" AND "Cryptocurrency prediction" — to identify relevant studies. These queries were applied to titles, abstracts, and keywords, covering literature published between January 2017 and November 2024. The initial search yielded 687 results, which were screened in multiple stages: duplicate entries were removed, followed by title and abstract screening, and then full-text eligibility assessment.

We included studies that were peer-reviewed, written in English, and focused on the application of Transformer-based architectures to blockchain problems, providing either empirical results or theoretical contributions. Articles were excluded if they were opinion-based, lacked technical depth, or addressed unrelated domains. After applying these criteria, a total of 236 studies were retained for detailed analysis. This selection process is illustrated in the PRISMA flowchart (Fig. 21), and served as the foundation for subsequent classification and synthesis across four thematic areas: anomaly detection, smart contract vulnerability analysis, cryptocurrency trend prediction, and code summarization.

5.2. Data collection

Following the initial literature search, we employed a structured data collection process to ensure that only high-quality and thematically relevant studies were included in this survey. All retrieved articles were first imported into a reference management system for deduplication. We then performed two-stage manual filtering: (1) Title and abstract screening to exclude clearly irrelevant works, and (2) Full-text eligibility assessment to determine whether the study specifically focused on Transformer models in blockchain contexts.

We applied the following inclusion criteria: (1) peer-reviewed publications between 2017 and 2024, (2) written in English, (3) explicit application of Transformer-based architectures to blockchain-related problems (e.g., anomaly detection, smart contracts, cryptocurrency), and (4) containing empirical evaluations, theoretical contributions, or review-based insights. Exclusion criteria included: (1) non-peer-reviewed or informal publications (e.g., white papers, editorials), (2) articles lacking sufficient methodological details, and (3) works only tangentially referencing blockchain or Transformers. After the screening and validation stages, we retained a final corpus of 236 qualified studies, forming the basis for our analysis. The full screening pipeline is visualized in Fig. 21 using the PRISMA model.

5.3. Data analysis

To extract meaningful insights from the selected studies, we employed a thematic analysis approach, organizing the final corpus of 236 papers into four primary domains based on their core research objectives and application focus: (1) anomaly detection, (2) smart contract vulnerability analysis, (3) cryptocurrency prediction and trend analysis, and (4) blockchain-related code summarization. Each paper was assigned to a category by at least two independent reviewers based on its stated aims, methodology, and experimental focus. In cases of overlap or ambiguity, the dominant contribution area was used for classification.

Within each domain, we further examined the papers in terms of their methodological design (e.g., sequence modeling, graph representation, hybrid models), data sources (e.g., Ethereum, transaction logs, code repositories), and evaluation metrics (e.g., accuracy, F1

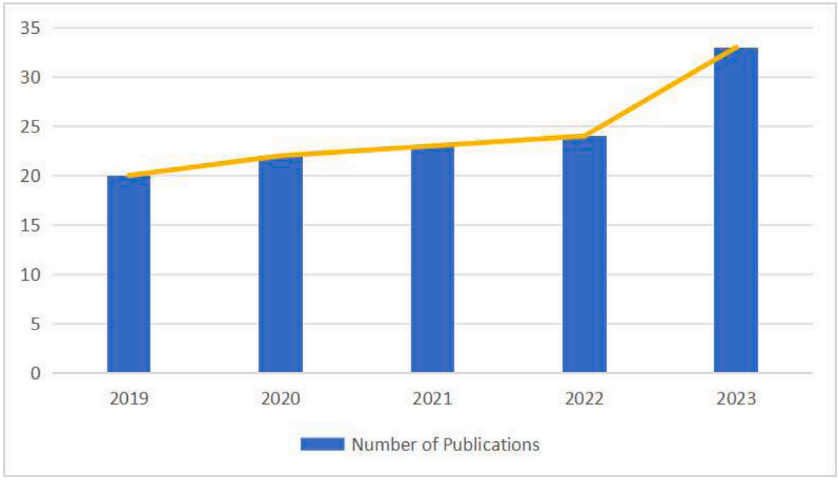

Fig. 20. Statistics on the publication of papers on the summarization of smart contract.

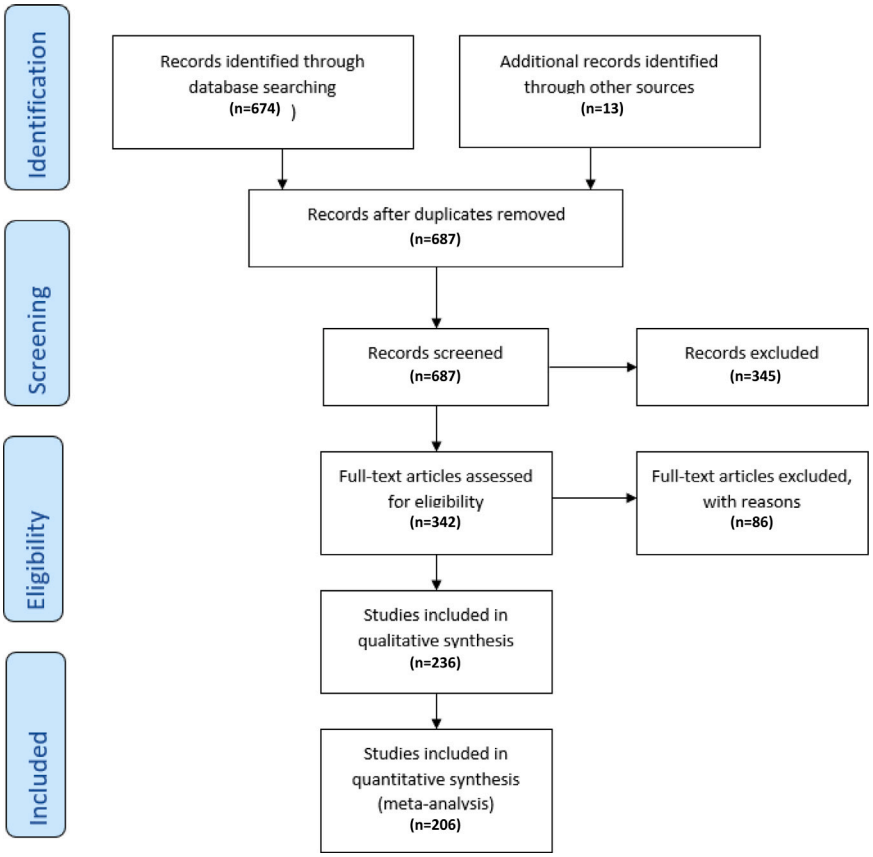

Fig. 21. PRISMA flowchart.

score, AUC). Particular attention was given to the role of Transformer architectures—whether as standalone models (e.g., BERT, GPT, T5) or components in composite frameworks (e.g., GraphCodeBERT, HGTs, CNN-Transformer hybrids). This allowed us to compare traditional methods with Transformer-based approaches, analyze performance gains, and identify common bottlenecks and research gaps within each subfield. To further contextualize the scope of the selected literature, we categorized each study based on its research type. This classification helps distinguish between empirical work, conceptual proposals, systematic reviews, and hybrid approaches that blend methodologies. The distribution is summarized in Table 3.

5.4. Evaluation criteria

To ensure the academic rigor and relevance of the selected literature, we applied a multi-criteria evaluation framework during the review process. Each included study was assessed across four key dimensions:

1. Scientific contribution — Whether the paper introduced novel methods, architectures, or frameworks involving Transformer models applied to blockchain tasks.

2. Methodological soundness — The clarity and reproducibility of the research design, including details on datasets, model configurations, and evaluation metrics.

Table 3
Classification of selected literature by research type.

Type	Count	Description
Empirical Studies	109	Experiments with Transformer models applied to real blockchain data
Theoretical Frameworks	42	Architectural proposals, algorithm design, or formal models
Systematic Reviews	21	Literature reviews or comparative analyses focusing on Transformer use in blockchain
Hybrid/System Approaches	34	Studies combining Transformer models with other methods (e.g., GNNs, rule-based systems)
Total*	**206**	**Some studies span multiple types; total is not strictly additive*

3. Impact and visibility — Considered through publication venue, citation count (as of July 2024), and recognition within the research community.

4. Practical relevance — The applicability of the proposed methods to real-world blockchain scenarios, including deployment feasibility and scalability.

Each dimension was qualitatively assessed, and papers scoring low across multiple criteria were excluded during the full-text screening phase described in Section 5.2. In addition, particular preference was given to studies that offered open-source implementations, used publicly available datasets, or included comparative baselines against traditional models.

This evaluation process ensured that the final set of reviewed studies represents a balanced sample of both foundational and state-of-the-art contributions, reflecting the current landscape of Transformer applications in blockchain.

5.5. Integration of findings

The final stage of the methodology involved synthesizing the insights extracted from the 236 selected studies into a cohesive, domain-specific narrative. After categorizing papers into four key application areas — anomaly detection, smart contract analysis, cryptocurrency prediction, and code summarization — we conducted a cross-domain comparison to identify overlapping methodologies, shared challenges, and recurring architectural patterns.

We observed that Transformer-based models are increasingly used as both standalone architectures (e.g., BERT, GPT-2/3, T5) and integrated components within hybrid systems (e.g., CNN-Transformer pipelines, Graph-based Transformers). Across domains, common challenges emerged, such as data sparsity, limited interpretability, and computational costs associated with large-scale models. In contrast, strengths like language understanding, temporal modeling, and cross-modal learning consistently contributed to performance gains.

The integration process also highlighted research gaps—such as the underutilization of Transformers in smart contract optimization and the limited adaptation of pretrained models to blockchain-specific vocabularies. These observations informed our discussion in Section 6 and guided the future research agenda presented in Section 7.

By consolidating diverse findings through a comparative lens, we provide a unified perspective on how Transformer models are reshaping blockchain research, enabling the development of more intelligent, secure, and scalable systems.

6. Transformer in blockchain applications

This chapter reviews the current research on Transformer applications in key blockchain areas—anomaly detection, smart contract vulnerability detection, cryptocurrency prediction and trend analysis, and code summarization. It evaluates the model's effectiveness, its ability to address existing challenges, and its potential to enhance blockchain security and efficiency through its unique processing capabilities. The Table 4 below summarizes the main application areas, including task descriptions, representative methods, and example applications.

We synthesize findings across the four key application areas discussed above and offer a critical reflection on the broader role of Transformer models in blockchain research. While these models have shown considerable promise — particularly in terms of contextual understanding, sequence modeling, and semantic representation — they are not without limitations.

We aim to move beyond descriptive summarization and provide a structured discussion along four central dimensions: (1) performance improvements versus computational cost, (2) robustness and generalizability across different blockchain tasks, (3) interpretability and auditability in high-stakes environments such as smart contract verification, and (4) scalability and data efficiency, especially in domains with limited labeled data. In addition, we discuss reported failure cases, implementation challenges, and potential biases, which are often underexplored in the current literature.

6.1. Anomaly detection

This section reviews the application of Transformer models in blockchain anomaly detection, emphasizing their role in addressing diverse challenges across different research themes. The studies are categorized by their focus areas: DeFi and financial transaction anomalies, smart contract and account fraud detection, IoT and permissioned blockchain monitoring, and advanced transaction pattern analysis. We will illustrates how these methods leverage Transformer models to enhance detection capabilities.

DeFi and Financial Transaction Anomalies: Song et al. (2023b) developed a framework for anomaly detection in decentralized finance (DeFi) systems, combining Variational Autoencoders (VAE) and Transformers. The VAE encodes time-series data into low-dimensional embeddings to capture short-term patterns, and then the method utilizes the Transformer to analyze long-term dependencies and handle complex temporal relationships. By reconstructing the data and comparing it with the original, the framework identifies anomalies like structural changes and malicious attacks. Tested on the Olympus DAO dataset, it demonstrated high accuracy, improving DeFi platform transparency and user asset security. (Wang et al., 2024) introduced the 1D SA-Inception model for detecting abnormal blockchain transactions, integrating 1D CNN, Inception structures, and Transformers. The CNN extract low-scale features, while the framework employs the Transformer's self-attention mechanism to prioritize critical data interactions, enhancing sensitivity to subtle patterns. With an AUC of 91.05%, G-mean of 84.95%, and F1 score of 83.52%, it outperforms traditional 1D CNNs, especially in imbalanced datasets. By leveraging the Transformer, this method improves precision in identifying rare anomalies. Wang et al. (2024) further highlights the role of Transformers in transaction analysis. The 1D SA-Inception model utilizes the Transformer's self-attention mechanism to optimize the recognition of complex patterns by focusing on key data interactions, improving the detection of rare anomalies in financial transactions.

Smart Contract and Account Fraud Detection: Liu et al. (2022) proposed Heterogeneous Graph Transformer Networks (HGTNs) to detect abnormal behaviors in Ethereum smart contracts. A Heterogeneous Information Network (HIN) is constructed with nodes (smart contracts, accounts, transactions) and edges (function calls, fund transfers). The method leverages the Transformer's self-attention mechanism to dynamically identify influential nodes and relationships, achieving 92% accuracy, 89% recall, and 90.5% F1 score, surpassing traditional methods. Xu et al. (2023b) applied HGTNs to detect illegal accounts on Ethereum, using an HIN of accounts, transactions, and blocks. It excels

Table 4
Applications of transformer models in blockchain.

Application area	Description	Representative methods
Anomaly Detection	Detecting fraudulent transactions and abnormal behaviors within blockchain systems.	Transformer models for transaction monitoring, fraud detection, and anomaly detection. For example: Identifying fraudulent transactions or abnormal network activity in blockchain networks.
Smart Contract Security	Enhancing the security of smart contracts through auditing and vulnerability detection.	Using Transformer models for code analysis and vulnerability detection in smart contracts. For example: Auditing Ethereum smart contracts for vulnerabilities such as reentrancy or gas overflow.
Cryptocurrency Prediction	Forecasting cryptocurrency trends and market behavior using historical and real-time data.	Transformer models for time-series prediction of cryptocurrency prices and trends. For example: Predicting cryptocurrency market changes based on historical price data and market sentiment.
Code Summarization	Automatically generating summaries of blockchain-related code for better understanding and debugging.	Transformer models for summarizing complex blockchain code. For example: Summarizing Ethereum smart contract code for easier comprehension and review.

in uncovering hidden patterns within complex networks, achieving 95.57% accuracy.

Chen et al. (2021) combined Abstract Syntax Trees (ASTs), Multi-Channel TextCNN, and Transformers to detect Ponzi schemes in Ethereum smart contracts. TextCNN extract local code features, while the Transformer is utilized to enhance the identification of sophisticated Ponzi schemes through understanding intricate, distributed logic, and capture code dependencies in contract logic, achieving 95.57% accuracy.

TLMG4Eth (Sun et al., 2025), a hybrid framework combining a transaction language model and graph-based methods to detect fraud in Ethereum transactions, enriched with semantic and similarity features. The model converts transaction data into sentences and constructs similarity and account interaction graphs to capture semantics, similarity patterns, and network structures. A Multi-Head Attention Network fuses semantic and similarity embeddings, complementing the account interaction graph's structural analysis. Experimental evaluations demonstrate performance improvements of 9.62% to 13.2% over state-of-the-art methods across three datasets. This study highlights the synergy of semantic modeling and graph-based learning in addressing Ethereum's fraud landscape.

Sheng et al. (2025) proposes a dynamic feature fusion model, combining graph-based representation learning and semantic feature extraction to detect blockchain fraud, enriched with structural and contextual features. The model constructs global graph representations and extracts local transaction semantics to capture network relationships and fraud patterns. A dynamic multimodal fusion mechanism integrates these features, complementing graph-based structural analysis. Experimental evaluations demonstrate superior accuracy, F1 score, and recall over benchmarks on real-world blockchain datasets.

IoT and Permissioned Blockchain Monitoring: Batool et al. (2022) introduced Block-FeST, integrating Federated Learning (FL), Split Learning (SL), Transformers, and blockchain for IoT anomaly detection. FL and SL ensure data privacy and reduce client computation, while the framework leverages the Transformer to process time-series data, capturing complex patterns and long-term dependencies. With 86% accuracy across test sets, Block-FeST uses smart contracts for automation and transparency, advancing data security standards in IoT environments. This method ensures efficient anomaly detection in resource-constrained settings by utilizing the Transformer to manage dynamic data streams. Zhou et al. (2022) proposed a CNN-Transformer hybrid for anomaly detection in permissioned blockchains. Operation logs are segmented into time-windowed blocks, with CNN extracting local features and the method employing the Transformer to analyze long-range patterns across blocks. This approach detects non-linear anomalies like hidden fraud, supporting near-real-time monitoring and improving system security. By leveraging the Transformer to process extensive log sequences, this framework excels in high-demand blockchain environments where rapid anomaly resolution is critical.

Table 5 summarizes recent Transformer-based approaches for anomaly detection in blockchain, highlighting their model architectures, data types, performance metrics, and key innovations across diverse contexts such as DeFi transactions, heterogeneous graphs, and smart contract behaviors. Transformers consistently outperform traditional methods across these themes by leveraging self-attention to capture long-range dependencies, complex patterns, and multi-level associations in blockchain data. Their parallel processing capabilities enable real-time detection, as seen in Liu et al. (2022) and Zhou et al. (2022), while their adaptability to diverse data types (time-series, graphs, code) enhances accuracy, as evidenced by high metrics in Xu et al. (2023b) and Chen et al. (2021). Key advantages include improved transparency (Song et al., 2023b), enhanced security (Batool et al., 2022), and sensitivity to rare anomalies (Wang et al., 2024). Compared with traditional methods such as rule-based anomaly filters, statistical time series models, and isolation forests, Transformer-based models demonstrate significantly improved ability to capture long-range dependencies in transaction behavior. As shown in Table 6, several studies report AUC gains of 5%–15% when using architectures like BERT or Gated Transformer Networks (GTNs). However, challenges remain, such as the computational complexity of Transformers, which may require optimization for resource-constrained environments like IoT. Moreover, false positives remain a concern, particularly when pretraining is not domain-specific. Future research could explore lightweight Transformer variants, integration with emerging blockchain technologies, and broader applications in privacy-preserving anomaly detection, further advancing blockchain security and efficiency.

6.2. Smart contract vulnerability detection

We categorize studies by the primary data type processed — source code, bytecode/opcodes, graph Transformer, and mixed data — offering detailed insights into Transformer applications, their performance, and unique contributions to enhancing smart contract security.

6.2.1. Source code analysis

These methods leverage Transformers to process smart contract source code (e.g., Solidity), extracting semantic and contextual features to identify vulnerabilities.

Peculiar (Wu et al., 2021) generates a Crucial Data Flow Graph (CDFG) by filtering security-relevant flows from a complete Data Flow Graph (DFG), reducing noise in source code analysis. GraphCodeBERT, pre-trained on code, enhances reentrancy detection by modeling code relationships, achieving 91.80% precision and 92.40% recall

Table 5
Comparison of transformer-based anomaly detection methods in blockchain.

Ref	Model structure	Technical pipeline	Data type	Performance	Key highlights
Signorini et al. (2018a)	VAE + Transformer	VAE encodes DeFi time series; Transformer captures long-range dependencies; Reconstruction-based anomaly detection	DeFi transaction sequences (Olympus DAO)	High detection accuracy across anomaly types	Joint local–global modeling; suitable for volatile DeFi environments
Li et al. (2017)	Heterogeneous Graph Transformer (HGT)	Heterogeneous Information Network (HIN) construction; Multi-type node interaction modeling; Transformer detects abnormal paths	Contract-account-transaction graphs	Accuracy 92%, F1-score 90.5%	Robust anomaly identification in complex network structures
Batool et al. (2022)	Federated + Split Learning + Transformer + Blockchain	Federated and split learning for distributed training; Transformer for time-series modeling; Blockchain ensures transparency	IoT edge-streaming data	Accuracy 86%	Privacy-preserving, lightweight client processing with decentralized learning
Weng et al. (2019)	Graph Transformer	Account-centric heterogeneous graph; Meta-path learning; Transformer extracts fraudulent behavior patterns	Ethereum account network	Accuracy 95.57%	Superior in illegal account detection with fast processing
Chen et al. (2022)	Multi-Channel CNN + Transformer	Abstract Syntax Tree (AST) to SBT sequences; Multi-channel CNN for local pattern extraction; Transformer captures cross-function dependencies	Ponzi scheme smart contract code	Accuracy 95.57%	Enhanced for structural fraud schemes with long-range logic modeling
Zhang et al. (2022a)	CNN + Transformer	Time-based log segmentation; CNN for local state extraction; Transformer models behavioral sequences for anomaly detection	Blockchain operation logs	Not specified	Real-time anomaly detection with complex behavior modeling
Wan et al. (2018)	1D SA-Inception + Transformer	Multi-scale feature extraction with 1D CNN and Inception; Transformer self-attention for temporal modeling	Blockchain transaction sequences	AUC 91.05%, F1-score 83.52%, G-mean 84.95%	Effective on imbalanced datasets and rare anomalies

Table 6
Comparison of anomaly detection methods on ethereum dataset.

Method	AUC	Latency (ms)	Interpretability
Isolation Forest	0.76	5.2	High
LSTM Autoencoder	0.82	12.4	Medium
Transformer (BERT)	0.89	35.1	Low
Gated Transformer Network	0.91	44.3	Low

on Ethereum contracts, outperforming static analysis tools due to its targeted data flow representation.

SmartConDetect (Jeon et al., 2021) extracts Solidity snippets via static analysis and employs a pre-trained BERT model to analyze code semantics. Its deep contextual understanding yields 98.7% precision, 86.2% recall, and a 90.9% F1 score on 10,000 Ethereum contracts, surpassing SVM (45.2% F1), Eth2Vec (57.5% F1), and DR-GCN (78.1% F1) by adapting to diverse coding styles.

ASSBert (Sun et al., 2023b) combines BERT with active learning to select impactful samples and semi-supervised learning to leverage unlabeled data, optimizing training with limited annotations. This approach achieves 78.6% accuracy for timestamp vulnerabilities using 20% labeled data, outperforming BERT-AL (79.1%) and BERT-SSL (54.5%) due to its efficient data utilization.

Xu et al. (2023a) pre-trains SolBERT on smart contract code for semantic feature extraction, paired with BiGRU for sequential analysis and hierarchical attention to prioritize critical code segments. This hybrid model achieves 93.85% accuracy and a 94.02% F1 score on Ethereum contracts, with a 4.5-second detection time, ideal for large-scale audits (Feist et al., 2019).

Tang et al. (2023) uses CodeBERT, pre-trained on diverse programming languages, to extract semantic features from Solidity code, feeding them to classifiers for vulnerability detection. Tested on thousands of contracts, it achieves 91% accuracy, 89% recall, and a 90% F1 score, a 10% F1 improvement over traditional methods, due to its robust feature extraction.

Lê Hông et al. (2023) employs a custom tokenizer to preserve code structure, then fine-tunes DistilBERT for reentrancy detection, integrating LSTM and MLP for enhanced classification. Its lightweight architecture delivers 99.71% performance on 101,082 samples, outperforming traditional deep learning models due to targeted optimization.

He et al. (2024) integrates BERT for semantic embedding, an attention mechanism to focus on key code parts, and BiLSTM for temporal analysis. This three-layer design achieves 98.58% accuracy and a 98.26% F1 score, surpassing AWD-LSTM (Merity et al., 2017) (90% F1) and MulCas (Ma et al., 2013) (78.9% F1) by prioritizing critical features.

MEVD (Guo et al., 2024) combines Transformers with multi-scale CNN encoders and Surface Feature Encoders (SFE) to capture global and local code features, augmented by Deep Residual Shrinking Networks (DRSN) to reduce redundancy. It achieves 92.13% accuracy for reentrancy and 90.85% for timestamp vulnerabilities, outperforming ReChecker (Qian et al., 2020) (69.41% F1) and CGE (85.43% F1) due to its multi-resolution analysis.

Jain and Tripathi (2024) uses a Transformer-BiGRU-TextCNN pipeline for progressive code analysis. The Transformer extracts contextual features, BiGRU processes temporal sequences, and TextCNN captures local patterns, achieving 96.1% accuracy, 97.8% recall, and a 96.5% F1 score on 49,552 contracts, surpassing static tools.

Chen et al. (2021) integrates Abstract Syntax Trees (ASTs) with Transformers to detect Ponzi schemes, leveraging ASTs to represent code logic. It achieves 95.57% accuracy on Ethereum contracts, excelling in identifying complex fraud patterns.

6.2.2. Bytecode and opcode analysis

These approaches process compiled bytecode or opcode sequences, enabling vulnerability detection without source code access.

VASCOT (Balcı et al., 2023) processes EVM bytecode with a Transformer model, identifying operational risks without source code access. Tested on 16,363 contracts, it achieves 90% accuracy, a 29% improvement over LSTM, with 6% of LSTM's training time, due to its efficient parallel processing, ideal for scalable audits.

TrapFormer (Gu et al., 2023) extracts opcode sequences from bytecode and applies a multi-layer Transformer to identify malicious behaviors like traps. Its stacked architecture enhances behavioral analysis, achieving a 98.1% F1 score for regular contracts, 96.5% for Ponzi contracts, and 98.4% for honey pot contracts, outperforming SCSGuard (Hu et al., 2022) (96.8% F1) and PSD-OL (87.5% F1).

VDDL (Jiang et al., 2022) uses a multi-layer bidirectional Transformer to analyze code sequences after standard preprocessing, identifying high-risk patterns through bidirectional context analysis. It achieves 92.35% accuracy, 81.43% recall, and an 86.38% F1 score, surpassing random forest (by 2.52% F1) and TextRNN (Hu et al., 2019) (by 2.74% F1).

SCGformer (Gong et al., 2023a) constructs control flow graphs (CFGs) from bytecode, using Transformers to analyze execution paths. Its graph-based approach achieves 94.36% accuracy and a 93.58% F1 score on 50,000 contracts, outperforming Slither and Oyente by adapting to new vulnerability patterns.

6.2.3. Graph transformer analysis

These methods apply Transformers to graph-based data (e.g., control flow or heterogeneous graphs) to model complex code or transaction interactions.

SCVDIE (Zhang et al., 2022b) constructs information graphs to represent code interactions, using an ensemble of seven neural networks, including Transformers, to analyze features from multiple perspectives. Ensemble learning enhances detection, achieving 95.46% accuracy, 96.81% precision, and 97.26% recall on 21,667 contracts, outperforming Oyente (46.1% F1), Mythril (Mueller, 2017) (45.6% F1), and Securify (Tsankov et al., 2018) (43.9% F1).

MANDO-HGT (Nguyen et al., 2023) transforms code into Heterogeneous Contract Graphs (HCGs) with nodes for functions, variables, and flows. Heterogeneous Graph Transformers model diverse relationships, achieving F1 scores from 0.7% to 76% across 55,000 contracts, excelling in multi-granularity detection.

GRATDet (Gong et al., 2023b) converts code into graph representations, abstracting functions and interactions as nodes and edges. Transformers analyze these graphs, achieving 95.22% accuracy, 95.59% precision, and a 95.16% F1 score, surpassing Mythril (Mueller, 2017) (50.44% F1) and SmartCheck due to its detailed interaction modeling.

Liu et al. (2022) uses Heterogeneous Graph Transformer Networks (HGTNs) to model contracts, accounts, and transactions in a Heterogeneous Information Network (HIN). It identifies key nodes, achieving 92% accuracy and a 90.5% F1 score on Ethereum data, outperforming traditional graph methods.

Xu et al. (2023b) applies HGTNs to account-transaction graphs for illegal account detection, leveraging graph structure analysis to achieve 95.57% accuracy on Ethereum datasets, excelling in uncovering hidden patterns.

Song et al. (2023b) processes DeFi transaction graphs with Transformers to detect structural anomalies, achieving high accuracy on Olympus DAO datasets by focusing on irregular transaction patterns, complementing vulnerability detection.

6.2.4. Mixed data analysis

These methods integrate source code with bytecode/opcodes or execution traces.

Zhang et al. (2022a) pairs BERT with a Multi-Objective Detection Neural Network (MODNN) to process source code and opcode-derived Critical Operation Sequences (COS). BERT extracts semantic features, while MODNN uses opcode co-occurrence matrices for multi-objective classification, achieving a 94.8% F1 score on 18,000 contracts, outperforming standard machine learning models.

Jie et al. (2023) integrates word2vec for semantic embeddings, BERT for code analysis, and GCN for bytecode-derived graphs, processing source code, bytecode, and execution traces. This multimodal approach achieves 99.71% performance on 101,082 functions from SmartEmbed, leveraging comprehensive feature fusion.

Transformer-based methods for smart contract vulnerability detection consistently outperform traditional approaches, such as SVM, CNN, and RNN, achieving 5%–15% higher AUC by effectively modeling intricate data relationships. This superior performance stems from pretraining on large datasets, as seen in BERT and CodeBERT, which enhances generalization across diverse contract structures, exemplified by SmartConDetect's robust semantic analysis and Peculiar's precise reentrancy detection. Additionally, the parallel processing capabilities of Transformers enable rapid analysis, as demonstrated by VASCOT's efficient bytecode audits and TrapFormer's swift trap identification.

Table 7 provides a detailed comparison of Transformer-based approaches for smart contract vulnerability detection. It outlines the architectural components, including hybrid models combining pretrained Transformers with graph structures, recurrent networks, or static analysis techniques. Each method is evaluated based on the types of contract data it processes, the specific vulnerabilities it targets, its detection performance (e.g., precision, recall, F1-score), and its advantages in terms of generalizability, interpretability, and structural awareness. Traditional vulnerability detection in smart contracts often relies on symbolic execution, static analysis, or SVM-based classifiers. While effective at detecting known patterns, these methods struggle with complex semantic relationships across contract components. Transformer-based models, particularly those using pretrained code models like CodeBERT or GraphCodeBERT, have achieved higher recall and F1 scores in recent benchmarks (Table 8). Nonetheless, Transformers are prone to overfitting on limited labeled data, and may fail in the presence of obfuscated code or adversarial syntax manipulation. Explainability also remains limited, making practical auditing difficult.

6.3. Cryptocurrency prediction and trend analysis

Transformer effectively captures complex temporal patterns and external influences in cryptocurrency price prediction, outperforming traditional methods like LSTM and RNN. This section examines studies that integrate Transformer architectures with sentiment analysis, technical market indicators, and hybrid neural networks to forecast prices for cryptocurrencies such as Bitcoin, Ethereum, and decentralized storage tokens. Each study is analyzed for its methodology, data sources, performance metrics, and contributions.

Zhao et al. (2022) develops a Transformer encoder model augmented with Time2Vec embeddings to forecast Bitcoin and Ethereum prices, incorporating sentiment analysis from social media. The model processes six years of Bitcoin and five years of Ethereum historical price data from CoinAPI, combined with sentiment scores derived from Twitter using the VADER tool. Time2Vec embeddings enable the model to capture both periodic and non-periodic patterns in price data, while multi-head attention enhances the integration of sentiment and price features. Transfer learning from Bitcoin to Ethereum data improves Ethereum predictions. Experimental results demonstrate significant improvements with sentiment integration, reducing the mean squared error (MSE) from 0.00137 to 0.00037, mean absolute percentage error (MAPE) from 0.18096 to 0.05816, and mean absolute error (MAE) from 0.02900 to 0.01435, outperforming LSTM models, which showed negligible gains from sentiment data. This approach underscores the Transformer's ability to fuse heterogeneous data for financial forecasting.

Penmetsa and Vemula (2023) proposes a hybrid framework combining Long Short-Term Memory (LSTM) and Transformer networks to predict prices of Bitcoin, Ethereum, and Litecoin, enriched with technical

Table 7
Comparison of transformer-based methods for smart contract vulnerability detection.

Ref	Model structure	Technical pipeline	Data type	Performance	Key highlights
Wei et al. (2019)	GraphCodeBERT + CDFG	Critical Data Flow Graph extraction; Semantic embedding with GraphCodeBERT; Transformer classification	Solidity smart contract code	Precision 91.8%, Recall 92.4%	Effective in capturing control-flow semantics and variable dependencies
Jain and Tripathi (2024)	Static Analysis + BERT	Extracts key contract segments using static analysis; BERT for semantic feature extraction	Solidity code snippets	F1-score 90.9%	Lightweight and interpretable, suitable for modular analysis
Yang and Zhu (2023)	BERT + MODNN	BERT encodes contract input sequences; Multi-objective deep neural network detects multiple vulnerabilities	Contract interaction sequences	F1-score 94.8%	Robust multi-vulnerability detection with strong generalization
Mueller (2017)	MANDO-HGT	Constructs Heterogeneous Contract Graph (HCG); Applies multi-head attention in HGT for structure-aware detection	Graph-based contract structure (HCG)	Significantly outperforms baselines	Structure-preserving learning for complex control flows
Wołk (2020)	SolBERT + BiGRU + Attention	Pre-trained SolBERT for token representation; BiGRU captures sequential logic; Hierarchical attention for key features	Solidity source code	Accuracy 93.85%, F1-score 94.02%	High interpretability and training efficiency
Sun et al. (2023a)	CodeBERT + Classifier	Embeds contract semantics using CodeBERT; Classifier detects vulnerability classes	Solidity contracts	Accuracy 91%, F1-score 90%	Generalizable across different code patterns and vulnerability types
Jayabalan and N. (2021)	Word2Vec + BERT + GCN	Multi-modal feature extraction from bytecode, source code, and EVM traces; GCN aggregates token interactions	Smart contract + bytecode + trace data	F1-score 99.71%	High-dimensional fusion improves detection granularity
Jay et al. (2020)	Transformer + Static Graphs	Leverages control and data flow graph; Transformer-based analysis for vulnerability patterns	SC semantic graph structures	Not specified	Effective for variable-dependent detection (e.g., VDD)
Kushwaha et al. (2022)	DistilBERT + LSTM + MLP	Compresses semantic representations using DistilBERT; LSTM for sequential modeling; MLP for classification	Solidity source code	Not specified	Lightweight model suitable for deployment scenarios

Table 8
Comparison of vulnerability detection models on smart contracts.

Method	F1 score	Training data required	Explainability
Mythril (Static Analysis)	0.68	None	High
SVM Classifier	0.71	1k samples	Medium
CodeBERT	0.82	10k+ labeled contracts	Low
GraphCodeBERT	0.85	10k+ labeled contracts	Very Low

market indicators. The model incorporates the Relative Strength Index (RSI), Bollinger Bands, and Moving Average Convergence Divergence (MACD) as input features to capture market momentum, volatility, and trend dynamics. The Transformer's self-attention mechanism processes multiple time points concurrently, complementing LSTM's ability to model long-term dependencies. Experimental evaluations demonstrate that the inclusion of technical indicators enhances prediction accuracy, with the Transformer outperforming standalone LSTM models across multiple test scenarios. This study highlights the synergy of traditional financial analysis and advanced neural networks in addressing the volatility of cryptocurrency markets.

Davoudi et al. (2023) presents a multi-faceted architecture for predicting price trends of decentralized storage cryptocurrencies, such as Filecoin, Storj, and Arweave, by integrating network analysis, text analysis, and market data. Network analysis identifies key entities associated with the cryptocurrency, forming a relational network. Text analysis employs the T5 model to summarize news articles and FinBERT to extract sentiment from articles and tweets, generating feature vectors. A Transformer encoder processes these vectors alongside market data, leveraging self-attention to model complex dependencies. The model achieves prediction accuracies of 76%, 83%, 61%, and 74% for Filecoin, Storj, Arweave, and other tokens, respectively, surpassing LSTM and traditional sentiment-based methods. This approach demonstrates the Transformer's strength in synthesizing multi-source data for niche cryptocurrency markets.

Khaniki and Manthouri (2024) introduces a model combining Transformer networks, Bidirectional LSTM (BiLSTM), and technical market indicators to forecast prices of Bitcoin, Ethereum, and Litecoin. The model integrates RSI and moving averages as input features to capture market trends and dynamics. The Transformer's multi-head attention mechanism models both short- and long-term temporal dependencies, while BiLSTM's bidirectional processing enhances the capture of sequential patterns. Experimental results show a 5%–10% reduction in MSE and RMSE compared to standalone LSTM and artificial neural network (ANN) models, highlighting the model's robustness in volatile markets. This study illustrates the value of hybrid architectures in improving prediction reliability.

Sridhar and Sanagavarapu (2021) employs a Transformer model with multi-head self-attention and Time2Vec embeddings to predict hourly Dogecoin prices. The Time2Vec layer converts temporal features into continuous vectors, enabling the model to capture periodic and non-periodic market patterns. The multi-head attention mechanism processes multiple data sequences in parallel, enhancing the model's ability to model complex temporal relationships. The model achieves a prediction accuracy of 98.46% and an R-squared value of 0.8616, outperforming LSTM and simple neural networks. Evaluations using MSE and MAE further confirm the model's stability in handling high-volatility data, demonstrating the efficacy of time-aware Transformer architectures.

Murray et al. (2023) proposes the Temporal Fusion Transformer (TFT) for price prediction of Bitcoin, Ethereum, Litecoin, Ripple, and Monero. The TFT integrates multi-head attention with gating layers, variable selection networks, and temporal processing components to manage dynamic, multi-source data. Experimental results show RMSEs

of 0.02353 for Bitcoin and 0.0181 for Ethereum, competitive with LSTM (0.02224, 0.0173) and GRU (0.02285, 0.0173), with superior flexibility in volatile markets. The TFT's ability to handle static and dynamic inputs makes it particularly suited for integrating diverse market influences, offering a versatile tool for financial forecasting.

Singh and Bhat (2024) forecasts Ethereum prices using a Transformer model that integrates sentiment data from Twitter and Reddit, processed by FinBERT, with price and trading volume data from Ethereum and correlated cryptocurrencies (Polkadot, Cardano). The model employs multiple Transformer encoder blocks with multi-head attention to capture temporal and cross-currency correlations. Using 730 days of data, it achieves an MSE of 0.068, higher than LSTM (0.0051) but outperforming ANN and MLP models. The inclusion of multi-currency data enhances prediction robustness, demonstrating the Transformer's capability to model complex market interactions.

Son et al. (2022) develops a framework that uses RoBERTa for stance detection on 2500 manually labeled Bitcoin-related tweets, achieving 81% accuracy, 92% recall, and an F1 score of 0.86, and a two-layer LSTM for price prediction. Tweets are cleaned to remove special characters, stopwords, and punctuation, and sentiment scores are paired with Yahoo Finance Bitcoin price data (2018–2022) using a 15-day input window. The LSTM model, normalized via Min-Max scaling, predicts prices outperforming GRU, linear regression, and ARIMA. This approach leverages RoBERTa's precise sentiment analysis and LSTM's robust time series modeling to enhance prediction accuracy in volatile markets.

Table 9 presents a comparative analysis of Transformer-based approaches for cryptocurrency prediction, highlighting models that integrate temporal encoding, sentiment analysis, technical indicators, and multi-currency interactions. These methods demonstrate improved predictive performance in volatile markets by capturing both sequential dependencies and cross-domain features. Forecasting cryptocurrency trends traditionally involves statistical methods (e.g., ARIMA), sentiment analysis, or shallow neural networks. While these models offer interpretable and fast predictions, they often fail to capture volatile nonlinear dynamics. Transformer-based time series models like Temporal Fusion Transformer (TFT) and Informer have demonstrated superior forecasting accuracy, particularly in capturing long-term dependencies. However, the gains are inconsistent across coins and time windows, and are highly sensitive to training window size, learning rate, and market shocks. Table 10 summarizes recent findings. Model robustness under real-world noise remains an unresolved issue.

6.4. Code summarization

Transformer enhances smart contract code summarization by modeling complex semantic and structural relationships, surpassing traditional rule-based, template-based, or shallow neural network methods. This section reviews studies applying Transformer architectures, often augmented with graph-based or multi-modal techniques, to generate concise, coherent summaries of smart contract code, typically in Solidity, Java, or Python. Each study is analyzed for its methodology, data processing, performance metrics, and contributions to code understanding.

Hu et al. (2021) proposes SMARTDOC, a Transformer-based model for generating user-oriented comments for Solidity smart contract functions, enhancing functional clarity and risk awareness. Source code is tokenized, and a Java-pre-trained encoder, leveraging syntactic similarities with Solidity, is fine-tuned on 7878 function–comment pairs from 54,739 Solidity contracts. A pointer generation mechanism copies keywords from source code, addressing dynamic expressions, while transfer learning mitigates the limited Solidity dataset size. Evaluated on Solidity datasets, SMARTDOC achieves a BLEU score of 47.39 and a ROUGE-L score of 51.86, outperforming attendgru (BLEU 29.01, ROUGE-L 38.48), ast-attendgru (LeClair et al., 2019) (26.01, 34.51), and Re2Com (Wei et al., 2020) (29.37, 34.55), due to its robust pre-training and precise keyword integration.

Ahmad et al. (2020) develops a Transformer encoder–decoder model for code summarization, processing tokenized source code to generate natural language summaries. The encoder employs relative position encoding to capture semantic relationships, while the decoder uses a copy mechanism to include rare tokens from the source code, ensuring summary completeness. Tested on Java and Python datasets, the model achieves BLEU scores of 44.58 (Java) and 32.52 (Python), METEOR scores of 26.43 and 19.77, and ROUGE-L scores of 54.76 and 46.73, respectively, surpassing the Dual Model (Wei et al., 2019) (Java: BLEU 42.39, Python: BLEU 21.80) by 2.19 and 10.72 in BLEU, due to its effective dependency modeling and parallel processing (Eriguchi et al., 2016).

Yang et al. (2021) introduces MMTrans, a multi-modal Transformer architecture integrating a graph encoder, Sequence-Based Tree (SBT) encoder, and joint decoder. Source code is parsed into an Abstract Syntax Tree (AST) and an SBT sequence via Structural Traversal. The graph encoder uses Graph Convolutional Networks (GCNs) to extract local semantic features, while the SBT encoder embeds the SBT sequence with positional encoding to capture global semantics. The joint decoder fuses these outputs to generate summaries, leveraging cross-modal attention. On Java and Python datasets, MMTrans improves BLEU by 3.5 and 2.7, ROUGE-L by 2.8 and 2.3, and METEOR by 2.2 and 1.9 over the Dual Model, due to its comprehensive structural and semantic feature extraction.

Gong et al. (2022) proposes SCRIPT, a Structural Relative Position Guided Transformer, comprising a Relative Distance Weighted Transformer (RDW-Transformer) and a Structural Relative Position Encoding Transformer (SRPEi-Transformer). Source code is tokenized and parsed into an AST, with shortest path lengths between nodes forming a structural relative position matrix. The RDW-Transformer weights structural dependencies, and the SRPEi-Transformer integrates AST-based position encoding, enhancing structural semantics. The stacked encoder, paired with a standard decoder, generates summaries. On Java and Python datasets, SCRIPT improves BLEU by 1.19 and 0.54, ROUGE-L by 1.15 and 0.65, and METEOR by 0.93 and 0.56 over the Dual Model (Wei et al., 2019), outperforming SiT due to its advanced structural modeling.

Gao and Lyu (2022) presents M2TS, a multi-scale multi-modal Transformer model featuring multi-scale AST feature extraction, cross-modal feature fusion, and a Transformer decoder. Source code is parsed into AST and SBT sequences, with a GCN extracting multi-scale features by computing the power matrix of the AST adjacency matrix. A cross-modal fusion method integrates GCN-extracted graph features with encoded token sequences, using attention to prioritize key features. The decoder generates summaries, capturing local and global semantics. On Java and Python datasets, M2TS achieves a BLEU-4 score of 46.84% (Java, +1.63% over SG-Trans (Gao et al., 2021)) and outperforms CODE-NN (Iyer et al., 2016), DeepCom (Hu et al., 2018a), and others in BLEU, METEOR, ROUGE-L, and CIDER, due to its robust feature fusion.

Shi et al. (2023) introduces CoSS, combining a Transformer encoder with a Graph Attention Network (GAT) encoder for code summarization. The Transformer encoder processes tokenized code with positional encoding, capturing semantic dependencies. A bidirectional LSTM generates initial statement embeddings, which the GAT encoder processes via a control flow graph (CFG) to model structural relationships. The joint decoder fuses both outputs to generate summaries, emphasizing semantic and structural coherence. On Java, Python, and Solidity datasets, CoSS achieves BLEU-4 scores of 46.84% (Java), 33.84% (Python), and 30.15% (Solidity), METEOR scores of 28.93%, 21.83%, and 18.77%, and ROUGE-L scores of 57.87%, 47.92%, and 42.34%, surpassing CODE-NN (e.g., Java: 26.07%), RL+Hybrid2Seq (Wan et al., 2018) (e.g., Java: BLEU 38.22, Python: BLEU 19.28), and Hybrid-DeepCom (e.g., Java: 38.55%) due to its integrated feature extraction.

Table 9
Comparison of transformer-based methods for cryptocurrency prediction and trend analysis.

Ref	Model structure	Technical pipeline	Data type	Performance	Key highlights
Signorini et al. (2018b)	Transformer + Time2Vec	Multi-head self-attention with Time2Vec encoding for temporal modeling	Dogecoin hourly price data	Accuracy 98.46%, R^2 = 0.8616	Captures periodic/non-periodic patterns; efficient parallel learning
Zhang and Shafiq (2024)	Transformer + VADER + Transfer Learning	Sentiment scoring via VADER; combined with historical price; transfer learning from BTC to ETH	BTC and ETH prices + Twitter sentiment	MSE ↓ 0.00137 to 0.00037; MAE ↓ 0.029 to 0.01435	Enhances prediction with sentiment; improves ETH forecast via BTC model
Tang et al. (2021)	Transformer + FinBERT	FinBERT extracts sentiment scores; Transformer models cross-currency relations	ETH price, volume + DOT/ADA data + social sentiment	RMSE 0.2608; MAPE 18.14%	Fuses multi-currency data; interpretable attention layers
Sun et al. (2023b)	Transformer + BiLSTM + Technical Indicators	RSI, MACD, etc. fed into BiLSTM + Performer Transformer	BTC/ETH/LTC historical prices + indicators	R^2 = 0.9997; RMSE 18.3	Low complexity; outperforms LSTM on volatility sensitivity
Sun et al. (2025)	Transformer + 1D CNN + Multi-source Fusion	Combines social media, market, and network data; Transformer learns trend features	Texts, sentiment, network features	Accuracy 83%	Applies to decentralized storage crypto; highly interpretable design
Nikolić et al. (2018)	LSTM + Transformer + Momentum/Volatility Indicators	Uses BiLSTM for sequential trends and Transformer for long-range dependencies	BTC/ETH/LTC prices + RSI, BB%b, MACD	Not specified	Hybrid deep learning with financial signals; robust short-term prediction
Shayegan et al. (2022)	RoBERTa + Transformer + RNN	RoBERTa classifies tweet stance; combined with RNN price predictor	Bitcoin price + Twitter stance-labeled data	Accuracy 81%, F1 = 0.86	Stance improves trend sensitivity; useful for short-term shifts
Shafay et al. (2023)	Transformer + FinBERT + Cross-currency modeling	ETH price modeled alongside DOT, ADA, sentiment data from Reddit/Twitter	ETH price/volume + multi-token + social media	MSE = 0.068	Strong multi-feature fusion; outperforms ANN/MLP baselines
Mettler (2016)	Temporal Fusion Transformer (TFT)	Combines gating, variable selection, and multi-head attention for series fusion	BTC/ETH/LTC time series	RMSE for BTC = 0.02353, ETH = 0.0181	Dynamic weighting of indicators; strong interpretability

Table 10
Forecasting accuracy of different models on bitcoin price data.

Model	RMSE	Sensitivity to market shocks	Training time (min)
ARIMA	412.6	High	<1
LSTM	328.9	Medium	14
Informer	291.2	Low	27
TFT	265.3	Medium	33

Gao et al. (2023) proposes SG-Trans, enhancing the Transformer by injecting local symbol information (tokens, statements) and global syntactic structures (data flow graphs) as adjacency matrices. These matrices constrain attention to structural relationships, with a hierarchical attention mechanism prioritizing local structures at lower layers and global structures at higher layers. Experimental results show that SG-Trans significantly outperforms existing methods on Java and Python benchmark datasets. On the Java dataset, SG-Trans achieved BLEU-4, METEOR, and ROUGE-L scores of 45.89, 27.85, and 55.79, respectively, while the state-of-the-art baseline method NeuralCodeSum (Ahmad et al., 2020) scored 45.15, 27.46, and 54.84, respectively. Similarly, on the Python dataset, SG-Trans achieved BLEU-4, METEOR, and ROUGE-L scores of 33.04, 20.52, and 47.01, respectively, compared to NeuralCodeSum's scores of 32.19, 19.96, and 46.32. In comparison to other methods, such as GREAT (Hellendoorn et al., 2019) and Transformer+GNN (Choi et al., 2021), SG-Trans also demonstrated significant advantages.

Table 11 compares Transformer-based models for smart contract code summarization, highlighting their architectural innovations, input representations, and summarization quality. These methods integrate control flow graphs, structural encoding, and transfer learning to enhance both the semantic accuracy and fluency of generated summaries. Traditional code summarization methods based on rule templates or sequence-to-sequence models often fail to capture semantic relationships in smart contracts. Transformer-based models, particularly those pretrained on code corpora (e.g., CodeT5, PLBART), significantly improve BLEU and ROUGE scores as shown in Table 12. However, these models require large-scale pretraining and often struggle with rare code patterns or poorly documented codebases. In addition, many models do not generalize well across languages, and hallucination of function names or contract intents is a known failure case.

Finally, Table 13 provides a comparison of traditional and Transformer-based approaches across four key blockchain research domains, outlining their methods, limitations, and advantages. Traditional techniques, such as statistical models, rule-based methods, and classical machine learning, often face challenges in scalability, contextual understanding, and capturing long-range dependencies. In contrast, Transformer-based models utilize self-attention mechanisms, deep contextual learning, and parallel processing, enabling enhanced adaptability, predictive accuracy, and efficiency in handling complex blockchain tasks. This table highlights the gaps in conventional approaches and demonstrates how Transformers offer more robust solutions for blockchain applications.

7. limitations, challenges, and future directions

While this survey has provided a comprehensive synthesis of the applications of Transformer models in blockchain, it is essential to present a balanced perspective by explicitly acknowledging the limitations of the review itself. Recognizing these boundaries enhances the transparency and credibility of our work and aligns with the PRISMA framework for systematic reviews. Beyond the scope of this review, it is also important to reflect on the broader technical challenges faced by Transformer applications in blockchain and to highlight potential future research directions.

Accordingly, this chapter is divided into two main parts. Section 7.1 Limitations discusses the inherent constraints of this survey, including issues such as publication bias, the limited availability of high-quality and standardized datasets, the uneven distribution of research topics,

Table 11
Comparison of transformer-based methods for smart contract code summarization.

Ref	Model structure	Technical pipeline	Data type	Performance	Key highlights
Sanju et al. (2023)	CoSS: Transformer + GAT + BiLSTM	Transformer encoder for tokens, BiLSTM for statements, GAT for CFG, joint decoding	Java, Python, Solidity	BLEU-4: 46.84/33.84/30.15; ROUGE-L: 57.87/47.92/42.34	Combines semantic and structural features; consistent cross-language performance
Xu et al. (2023c)	MMTrans: Multi-modal Transformer + GCN + SBT	AST→ graph, SBT→ sequence, encoded separately, fused via Transformer decoder	Java, Python	BLEU +3.5, ROUGE-L +2.8 over baseline	Fuses graph and token sequence; strong global and local semantic capture
Gong et al. (2023a)	SCRIPT: RDW + SRPEi Transformer	Adds structural relative positional encoding; uses distance weighting; Transformer decoder	Java, Python	BLEU-4: 45.89, ROUGE-L: 55.79, METEOR: 27.85	Improves structure modeling with AST paths and token positions
Yan et al. (2022)	SMARTDOC: Transformer + Pointer + Transfer Learning	Java pretraining + Solidity finetuning + pointer generator for copying	Solidity	BLEU: 47.39, ROUGE-L: 51.86	Excellent generalization to smart contracts; fluent summaries with real-world usability
Wang et al. (2023)	Full Transformer (baseline)	Self-attention encoder–decoder with relative position encoding	Java, Python	BLEU: 44.58/32.52; ROUGE-L: 54.76/46.73	Stronger than Dual Model; effective long-sequence and context modeling

Table 12
Performance comparison for code summarization on solidity contracts.

Model	BLEU	Pretraining required	Cross-language generalization
Template-based Rules	14.6	No	Low
Seq2Seq + Attention	21.3	No	Medium
CodeT5	32.8	Yes	Medium
PLBART	34.5	Yes	Low

and the challenges of comparing heterogeneous Transformer architectures. These limitations define the boundaries of our conclusions and provide context for interpreting the findings.

Section 7.2 Challenges and Future Directions shifts the focus from the review itself to the technical landscape of Transformer-based blockchain applications. Here, we analyze pressing research challenges — such as data scarcity, computational overhead, interpretability, and integration with blockchain-specific mechanisms — and propose promising directions for future exploration. By addressing both the methodological constraints of the survey and the technical challenges in the field, this chapter aims to provide a holistic understanding of the state of the art and to chart a clearer path for subsequent research.

7.1. Limitations

Although this survey aims to provide a comprehensive and systematic overview of Transformer applications in blockchain, several limitations must be acknowledged to ensure transparency and provide readers with a balanced perspective. These limitations primarily arise from the nature of the available literature, the scope of the review, and the heterogeneity of the studies included.

Publication Bias. As in many systematic reviews, this work may be subject to publication bias. Research that reports novel or positive results tends to be more frequently published and cited, while studies yielding negative, inconclusive, or non-significant outcomes are often underrepresented. Consequently, the findings synthesized in this survey may unintentionally overstate the effectiveness or maturity of Transformer models in blockchain applications.

Dataset Availability and Coverage. The conclusions of this review are shaped by the limited availability of high-quality, standardized, and publicly accessible blockchain datasets. Many primary studies rely on proprietary, domain-specific, or small-scale datasets, which limits its reproducibility and generalizability. Moreover, the lack of shared benchmarks across studies makes it challenging to draw robust, cross-comparable insights. As a result, the survey's synthesis is constrained by the uneven landscape of empirical evidence.

Scope Dependence. Despite covering more than 200 studies, the scope of this review is inevitably influenced by the dominant research themes in the literature. Areas such as smart contract vulnerability detection, cryptocurrency price prediction, and anomaly detection receive substantial coverage, while emerging applications — such as blockchain governance, interoperability, and cross-chain communication — remain relatively underexplored. This imbalance restricts the comprehensiveness of the conclusions and highlights the evolving nature of the field.

Heterogeneity of Transformer Approaches. The reviewed studies employ diverse Transformer architectures (e.g., BERT, GPT, RoBERTa, GraphCodeBERT, and hybrid frameworks), training strategies, datasets, and evaluation metrics. Such heterogeneity complicates direct cross-study comparisons and precludes the establishment of unified performance baselines. In some cases, variations in experimental settings or reporting standards may have led to inconsistencies in the comparative analysis presented here.

Temporal and Methodological Constraints. Given the rapid pace of developments in both blockchain and natural language processing, the findings of this survey represent only a snapshot of the current state of the art. Studies published after the search cutoff date are not included, and the dynamic evolution of Transformer models (e.g., GPT-4, LLaMA, domain-specific fine-tuned models) may soon shift the landscape. Moreover, although we adhered to PRISMA guidelines to ensure methodological rigor, the process of categorization and interpretation inevitably involved subjective judgment, which may influence the synthesis.

In summary, while this survey provides valuable insights into the intersection of Transformers and blockchain, its conclusions should be interpreted within the context of these limitations. Recognizing these boundaries not only strengthens the transparency and reliability of the review but also highlights opportunities for future research—such as mitigating publication bias, curating open benchmark datasets, broadening the scope of applications, and fostering standardized evaluation protocols.

7.2. Challenges and future directions

Applying Transformer models to blockchain technology presents a range of intricate challenges, particularly in the domains of cryptocurrency price prediction, smart contract vulnerability detection,

Table 13
Comparison of traditional vs. transformer-based approaches in blockchain research.

Domain	Traditional methods	Limitations	Transformer-based approaches	Advantages
Anomaly Detection	SVM, K-means, network monitoring, data fusion	Manual feature engineering, poor long-term dependency modeling, low scalability	Self-attention for deep contextual analysis, parallel processing for real-time monitoring	Captures complex patterns, adapts to evolving attacks, scalable for large datasets
Smart Contract Security	Graph Neural Networks (GNN), Abstract Syntax Trees (AST) analysis	Limited semantic understanding, poor handling of long-range dependencies	Transformer-based vulnerability detection (e.g., GRATDet), self-attention for execution flow analysis	Higher detection accuracy, adapts to evolving attack vectors
Cryptocurrency Prediction	Time-series models (AR, MA, ARMA, ARIMA), technical indicators	Weak at long-term dependencies, poor adaptability to market volatility	Transformer-based trend analysis, integrating market sentiment	Captures short- and long-term dependencies, improved prediction accuracy
Code Summarization	Rule-based, AST feature extraction, RNNs, LSTMs	Limited contextual understanding, struggles with long dependencies, poor adaptability	Transformer-based summarization (e.g., M2TS, SCRIPT), multi-modal AST-token embeddings	Better long-range dependency modeling, higher BLEU, METEOR, and ROUGE scores

and code summarization. These challenges stem from the computational demands of Transformers, their data processing requirements, and their integration with blockchain's decentralized, security-critical infrastructure (Bernabe et al., 2019; Golait et al., 2023).

- **Weak Local Semantic Capture:** Transformers struggle to model fine-grained local semantics, such as specific code statements in summarization or short-term market fluctuations in cryptocurrency price prediction. This limitation reduces precision in localized blockchain tasks, where CNNs or Mamba models may excel due to their focus on local feature extraction (Penmetsa and Vemula, 2023; Shi et al., 2023).
- **Cross-Function/Cross-Contract Vulnerability Modeling:** In smart contract vulnerability detection, Transformers face difficulties capturing cross-function or cross-contract dependencies, such as reentrancy vulnerabilities. Sequence segmentation in long contracts disrupts contextual understanding, lowering recall for complex issues (Aggarwal et al., 2019; Wang et al., 2020).
- **Structural Complexity in ASTs/CFGs:** For code summarization, Transformers risk attention collapse when processing complex Abstract Syntax Trees (ASTs) or Control Flow Graphs (CFGs), particularly in deeply nested code, limiting their ability to capture structural semantics essential for coherent summaries (Zhang et al., 2023).
- **Domain-Specific Adaptation:** Transformers exhibit reduced performance in domain-specific languages like Solidity compared to general languages (e.g., Java), due to limited training data and unique syntactic structures, impacting summarization and vulnerability detection in blockchain contexts (Hu et al., 2021; Shi et al., 2023).
- **High Computational Demands:** The resource-intensive nature of Transformer models increases node burdens, potentially causing network latency and degrading blockchain performance, particularly in decentralized environments (Khan et al., 2021; Bhutta et al., 2021; Kneissler and Oelbracht, 2023).
- **Data Privacy and Centralized Training Needs:** Transformers often require large, centralized datasets for training, posing challenges for data collection and privacy protection, especially when sharing data across blockchain nodes, complicating efficient processing while ensuring confidentiality (Rana et al., 2021; Chen et al., 2022).

Addressing these challenges requires advancements in algorithm design, hardware optimization, distributed computing, security frameworks, and system integration. Future research directions include:

- **Efficient Transformer Architectures:** Investigate how advanced optimization techniques, such as quantization, knowledge distillation, and FlashAttention, can be tailored to develop resource-efficient Transformer models for resource-constrained blockchain nodes (Laroiya et al., 2020).
- **Privacy-Preserving Training in Heterogeneous Blockchain Environments:** Examine the feasibility of federated learning and differential privacy for collaborative Transformer training across blockchain nodes with diverse data structures and consensus mechanisms. Investigating lightweight secure multi-party computation (MPC) and homomorphic encryption could balance privacy and efficiency, addressing data privacy challenges in code summarization and price prediction.
- **Robust Security Mechanisms for Hybrid Architectures:** Investigate AI-driven threat detection frameworks to identify and mitigate adversarial attacks and vulnerabilities in hybrid Transformer architectures (e.g., combined with graph neural networks) (Zhang et al., 2019).
- **Explainable Transformer Models for Trust and Auditability:** Explore interpretable Transformer architectures, leveraging attention visualization, post-hoc explanation methods, or hybrid CNN-Transformer designs, to enhance decision transparency in financial prediction and smart contract applications. Investigating how explainability impacts user trust could address domain-specific adaptation challenges.
- **Energy-Efficient Transformer Training and Inference:** Examine green AI techniques, such as carbon-aware scheduling, low-power hardware, and energy-optimized algorithms, to reduce the energy footprint of Transformers in energy-intensive blockchain networks (e.g., proof-of-work systems). Research into trade-offs between efficiency and performance could mitigate high computational demands.

8. Conclusion

This survey synthesizes the role of Transformer models in blockchain, revealing their strengths in anomaly detection, smart contract vulnerability detection, cryptocurrency prediction, and code summarization, while highlighting key challenges. Transformers enhance pattern recognition and automation across domains through their exceptional ability to absorb large-scale data and leverage parallel processing, but face practical hurdles like high computational demands, data privacy concerns, and limited interpretability, which hinder trust and scalability. Future research should focus on developing lightweight, privacy-preserving, and interpretable Transformer variants — potentially through efficient architectures, federated learning, and domain-specific adaptations — to unlock secure, efficient, and intelligent Web3 systems.

Table 14
List of acronyms.

Acronym	Full form
AUC	Area Under the Curve (ROC AUC)
AR	Autoregressive
ARMA	Autoregressive Moving Average
ARIMA	Autoregressive Integrated Moving Average
AST	Abstract Syntax Tree
BERT	Bidirectional Encoder Representations from Transformers
BiGRU	Bidirectional Gated Recurrent Unit
BiLSTM	Bidirectional Long Short-Term Memory
BLEU	Bilingual Evaluation Understudy score
CDFG	Crucial Data Flow Graph
CFG	Control Flow Graph
CNN	Convolutional Neural Network
COS	Critical Operation Sequences
DFG	Data Flow Graph
DeFi	Decentralized Finance
EVM	Ethereum Virtual Machine
F1	F1 score (harmonic mean of precision and recall)
FL	Federated Learning
G-mean	Geometric mean
GAT	Graph Attention Network
GCN	Graph Convolutional Network
GNN	Graph Neural Network
GPT	Generative Pre-trained Transformer
GTN	Gated Transformer Network
GRU	Gated Recurrent Unit
GPU	Graphics Processing Unit
HCG	Heterogeneous Contract Graph
HIN	Heterogeneous Information Network
HGT/HGTN	Heterogeneous Graph Transformer/HGT Networks
IoT	Internet of Things
LSTM	Long Short-Term Memory
MACD	Moving Average Convergence Divergence
MAPE	Mean Absolute Percentage Error
MAE	Mean Absolute Error
METEOR	Metric for Evaluation of Translation with Explicit ORdering
MLP	Multi-Layer Perceptron
MODNN	Multi-Objective Detection Neural Network
MPC	Multi-Party Computation
MLM	Masked Language Modeling
NLP	Natural Language Processing
OCSVM	One-Class Support Vector Machine
PLBART	PLBART (pretrained model for code/text)
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
RMSE	Root Mean Squared Error
RoBERTa	Robustly Optimized BERT Approach
RNN	Recurrent Neural Network
RSI	Relative Strength Index
SBT	Structure-Based Traversal
SFE	Surface Feature Encoder
SL	Split Learning
SVM	Support Vector Machine
TFT	Temporal Fusion Transformer
T5	Text-to-Text Transfer Transformer
VAE	Variational Autoencoder
VASCOT	VASCOT (Transformer-based model for EVM bytecode analysis)

CRediT authorship contribution statement

Tianxu Liu: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Yanbin Wang:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Jianguo Sun:** Investigation, Funding acquisition. **Ye Tian:** Validation, Supervision. **Yanyu Huang:** Project administration, Methodology. **Peiyue Li:** Validation, Software. **Yiwei Liu:** Visualization, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Characteristic Innovation Project of Ordinary Universities, Guangdong Provincial Department of Education, grant number 2025KTSCX190.

Appendix A

A.1. acronyms

See Table 14.

Data availability

Data will be made available on request.

References

Abdul Rashid, N., Ismail, M.T., 2023. Modelling and forecasting the trend in cryptocurrency prices. J. Inf. Commun. Technol. 22 (3), 449–501.
Aejas, B., Bouras, A., 2021. Effective smart contracts for supply chain contracts. Build. Resil. Univ.: Role Innov. Entrep. http://dx.doi.org/10.29117/quarfe.2021.0160.
Aggarwal, S., Chaudhary, R., Aujla, G., Kumar, N., Choo, K.-K.R., Zomaya, A.Y., 2019. Blockchain for smart communities: Applications, challenges and opportunities. J. Netw. Comput. Appl. 144, 13–48. http://dx.doi.org/10.1016/J.JNCA.2019.06.018.
Ahmad, W.U., Chakraborty, S., Ray, B., Chang, K.-W., 2020. A transformer-based approach for source code summarization. ArXiv preprint arXiv:2005.00653.
Ahmad, W.U., Chakraborty, S., Ray, B., Chang, K.-W., 2021a. Unified pre-training for program understanding and generation. ArXiv preprint arXiv:2103.06333.
Ahmad, Z., Shahid Khan, A., Nisar, K., Haider, I., Hassan, R., Haque, M.R., Tarmizi, S., Rodrigues, J.J., 2021b. Anomaly detection using deep neural network for IoT architecture. Appl. Sci. 11 (15), 7050.
Akila, V., Nitin, M., Prasanth, I., Reddy, S., Kumar, A., 2023. A cryptocurrency price prediction model using deep learning. In: E3S Web of Conferences. vol. 391, EDP Sciences, p. 01112.
Alamery, F.M.S., 2023. Cryptocurrency analysis using machine learning and deep learning approaches. J. Comput. Electr. Electron. Eng. Sci. 1 (2), 29–33.
Alikhani, A., Hamidi, H.-R., 2021. Regulating smart contracts: An efficient integration approach. Intell. Decis. Technol. 15, 397–404. http://dx.doi.org/10.3233/idt-200180.
Alon, U., Zilberstein, M., Levy, O., Yahav, E., 2019. Code2vec: Learning distributed representations of code. Proc. ACM Program. Lang. 3 (POPL), 1–29.
Amin, S., Neumann, G., 2021. T2NER: Transformers based transfer learning framework for named entity recognition. pp. 212–220. http://dx.doi.org/10.18653/v1/2021.eacl-demos.25.
Antonopoulos, A.M., Harding, D.A., 2023. Mastering Bitcoin. O'Reilly Media, Inc.
Armour, J., Awrey, D., Davies, P.L., Enriques, L., Gordon, J.N., Mayer, C.P., Payne, J., 2016. Principles of Financial Regulation. Oxford University Press.
Atzei, N., Bartoletti, M., Cimoli, T., 2017. A survey of attacks on ethereum smart contracts (sok). In: Principles of Security and Trust: 6th International Conference, POST 2017, Held As Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22–29, 2017, Proceedings 6. Springer, pp. 164–186.
Bahdanau, D., Cho, K., Bengio, Y., 2016. Neural machine translation by jointly learning to align and translate. arXiv:1409.0473.
Balcı, E., Yılmaz, G., Uzunoğlu, A., Soyak, E.G., 2023. Accelerating smart contract vulnerability scan using transformers. In: 2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering. CSDE, IEEE, pp. 1–6.
Bariviera, A.F., 2017. The inefficiency of Bitcoin revisited: A dynamic approach. Econom. Lett. 161, 1–4.
Bartoletti, M., Pompianu, L., 2017. An empirical analysis of smart contracts: platforms, applications, and design patterns. In: Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers 21. Springer, pp. 494–509.
Batool, Z., Zhang, K., Zhu, Z., Aravamuthan, S., Aivodji, U., 2022. Block-FeST: A blockchain-based federated anomaly detection framework with computation offloading using transformers. In: 2022 IEEE 1st Global Emerging Technology Blockchain Forum: Blockchain & beyond (IGETblockchain). IEEE, pp. 1–6.

Baur, D.G., Hong, K., Lee, A.D., 2018. Bitcoin: Medium of exchange or speculative assets? J. Int. Financ. Mark. Inst. Money 54, 177–189.
Bernabe, J.B., Cánovas, J.L., Hernández-Ramos, J.L., Moreno, R.T., Skarmeta, A., 2019. Privacy-preserving solutions for blockchain: Review and challenges. IEEE Access 7, 164908–164940. http://dx.doi.org/10.1109/ACCESS.2019.2950872.
Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Gollamudi, A., Gonthier, G., Kobeissi, N., Kulatova, N., Rastogi, A., Sibut-Pinote, T., Swamy, N., et al., 2016. Formal verification of smart contracts: Short paper. In: Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security. pp. 91–96.
Bhatt, S., Ghazanfar, M., Amirhosseini, M., 2023. Sentiment-driven cryptocurrency price prediction: A machine learning approach utilizing historical data and social media sentiment analysis. Mach. Learn. Appl.: Int. J. (MLAIJ) 10 (2/3), 1–15.
Bhutta, M.N.M., Khwaja, A., Nadeem, A., Ahmad, H.F., Khan, M., Hanif, M., Song, H., Alshamari, M.A., Cao, Y., 2021. A survey on blockchain technology: Evolution, architecture and security. IEEE Access 9, 61048–61073. http://dx.doi.org/10.1109/ACCESS.2021.3072849.
Boukhers, Z., Bouabdallah, A., Lohr, M., Jürjens, J., 2022. Ensemble and multimodal approach for forecasting cryptocurrency price. ArXiv preprint arXiv:2202.08967.
Cao, Y., Jiang, F., Xiao, J., Chen, S., Shao, X., Wu, C., 2023. Scheck: A novel graph-driven and attention-enabled smart contract vulnerability detection framework for web 3.0 ecosystem. IEEE Trans. Netw. Sci. Eng.
Catalini, C., Gans, J.S., 2020. Some simple economics of the blockchain. Commun. ACM 63 (7), 80–90.
Catania, L., Grassi, S., Ravazzolo, F., 2019. Forecasting cryptocurrencies under model and parameter instability. Int. J. Forecast. 35 (2), 485–501.
Chalkiadakis, I., Zaremba, A., Peters, G.W., Chantler, M.J., 2022. On-chain analytics for sentiment-driven statistical causality in cryptocurrencies. Blockchain: Res. Appl. 3 (2), 100063.
Charlier, J., State, R., Hilger, J., 2017. Modeling smart contracts activities: A tensor based approach. ArXiv arXiv:1905.09868.
Chen, Y., Dai, H., Yu, X., Hu, W., Xie, Z., Tan, C., 2021. Improving Ponzi scheme contract detection using multi-channel TextCNN and transformer. Sensors 21 (19), 6417.
Chen, R., Wang, L., Peng, C., Zhu, R., 2022. An effective sharding consensus algorithm for blockchain systems. Electronics 11 (16), http://dx.doi.org/10.3390/electronics11162597, URL https://www.mdpi.com/2079-9292/11/16/2597.
Chernyavskiy, A., Ilvovsky, D., Nakov, P., 2021. Transformers: "the end of history" for natural language processing? In: Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III 21. Springer, pp. 677–693.
Choi, Y., Bak, J., Na, C., Lee, J.-H., 2021. Learning sequential and structural information for source code summarization. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 2842–2851.
Chu, H., Zhang, P., Dong, H., Xiao, Y., Ji, S., Li, W., 2023. A survey on smart contract vulnerabilities: Data sources, detection and repair. Inf. Softw. Technol. 107221.
Cong, L.W., He, Z., 2019. Blockchain disruption and smart contracts. Rev. Financ. Stud. 32 (5), 1754–1797.
Conoscenti, M., Vetro, A., De Martin, J.C., 2016. Blockchain for the internet of things: A systematic literature review. In: 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications. AICCSA, IEEE, pp. 1–6.
Corbet, S., Lucey, B., Urquhart, A., Yarovaya, L., 2019. Cryptocurrencies as a financial asset: A systematic analysis. Int. Rev. Financ. Anal. 62, 182–199.
Davoudi, M., Ghavipour, M., Sargolzaei-Javan, M., Dinparast, S., 2023. Decentralized storage cryptocurrencies: An innovative network-based model for identifying effective entities and forecasting future price trends.
Delmolino, K., Arnett, M., Kosba, A., Miller, A., Shi, E., 2016. Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab. In: Financial Cryptography and Data Security: FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers 20. Springer, pp. 79–94.
Deng, W., Wei, H., Huang, T., Cao, C., Peng, Y., Hu, X., 2023. Smart contract vulnerability detection based on deep learning and multimodal decision fusion. Sensors 23 (16), 7246.
Derbentsev, V., Datsenko, N., Babenko, V., Pushko, O., Pursky, O., 2020. Forecasting cryptocurrency prices using ensembles-based machine learning approach. In: 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology. PIC S&T, IEEE, pp. 707–712.
Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805.
Dolgui, A., Ivanov, D., Potryasaev, S., Sokolov, B., Ivanova, M., Werner, F., 2020. Blockchain-oriented dynamic modelling of smart contract design and execution in the supply chain. Int. J. Prod. Res. 58, 2184–2199. http://dx.doi.org/10.1080/00207543.2019.1627439.
Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. ArXiv arXiv:2010.11929.
Du, X., Tang, Z., Wu, J., Chen, K., Cai, Y., 2022. A new hybrid cryptocurrency returns forecasting method based on multiscale decomposition and an optimized extreme learning machine using the sparrow search algorithm. IEEE Access 10, 60397–60411.
Ekblaw, A., Azaria, A., Halamka, J.D., Lippman, A., 2016. A case study for blockchain in healthcare:"MedRec" prototype for electronic health records and medical research data. In: Proceedings of IEEE Open & Big Data Conference, vol. 13. Vienna, Austria, p. 13.
Elman, J.L., 1990. Finding structure in time. Cogn. Sci. 14 (2), 179–211.
Eriguchi, A., Hashimoto, K., Tsuruoka, Y., 2016. Tree-to-sequence attentional neural machine translation. ArXiv preprint arXiv:1603.06075.
Feist, J., Grieco, G., Groce, A., 2019. Slither: a static analysis framework for smart contracts. In: 2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain. WETSEB, IEEE, pp. 8–15.
Fiore, M., Capodici, A., Rucci, P., Bianconi, A., Longo, G., Ricci, M., Sanmarchi, F., Golinelli, D., 2023. Blockchain for the healthcare supply chain: A systematic literature review. Appl. Sci. 13 (2), 686.
Gabriel, R.A., Park, B.H., Mehdipour, S., Bongbong, D.N., Simpson, S., Waterman, R.S., 2023. Leveraging a natural language processing model (transformers) on electronic medical record notes to classify persistent opioid use after surgery. Anesth. Analg. 137, 714–716. http://dx.doi.org/10.1213/ANE.0000000000006579.
Gao, S., Gao, C., He, Y., Zeng, J., Nie, L.Y., Xia, X., 2021. Code structure guided transformer for source code summarization. CoRR abs/2104.09340 arXiv preprint arXiv:2104.09340.
Gao, S., Gao, C., He, Y., Zeng, J., Nie, L., Xia, X., Lyu, M., 2023. Code structure–guided transformer for source code summarization. ACM Trans. Softw. Eng. Methodol. 32 (1), 1–32.
Gao, Y., Lyu, C., 2022. M2ts: Multi-scale multi-modal approach based on transformer for source code summarization. In: Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension. pp. 24–35.
Ghosh, A., Gupta, S., Dua, A., Kumar, N., 2020. Security of cryptocurrencies in blockchain technology: State-of-art, challenges and future prospects. J. Netw. Comput. Appl. 163, 102635. http://dx.doi.org/10.1016/j.jnca.2020.102635.
Golait, P., Tomar, D.S., Pateriya, R., Sharma, Y.K., 2023. Blockchain security and challenges: A review. In: 2023 IEEE 2nd International Conference on Industrial Electronics: Developments & Applications. ICIDeA, pp. 140–145. http://dx.doi.org/10.1109/ICIDeA59866.2023.10295211.
Gong, Z., Gao, C., Wang, Y., Gu, W., Peng, Y., Xu, Z., 2022. Source code summarization with structural relative position guided transformer. In: 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering. SANER, IEEE, pp. 13–24.
Gong, K., Song, X., Wang, N., Wang, C., Zhu, H., 2023a. SCGformer: Smart contract vulnerability detection based on control flow graph and transformer. IET Blockchain 3 (4), 213–221.
Gong, P., Yang, W., Wang, L., Wei, F., HaiLaTi, K., Liao, Y., 2023b. GRATDet: Smart contract vulnerability detector based on graph representation and transformer. Comput. Mater. Contin. 76 (2).
Grech, N., Kong, M., Jurisevic, A., Brent, L., Scholz, B., Smaragdakis, Y., 2018. Madmax: Surviving out-of-gas conditions in ethereum smart contracts. Proc. ACM Program. Lang. 2 (OOPSLA), 1–27.
Gu, T., Han, M., He, S., Chen, X., 2023. Trap contract detection in blockchain with improved transformer. In: GLOBECOM 2023-2023 IEEE Global Communications Conference. IEEE, pp. 5141–5146.
Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., et al., 2020. Conformer: Convolution-augmented transformer for speech recognition. ArXiv preprint arXiv:2005.08100.
Gunarto, D.M., Sa'adah, S., Utama, D.Q., 2023. Predicting cryptocurrency price using rnn and lstm method. J. Sisfokom (Sistem Inf. Dan Komputer) 12 (1), 1–8.
Guo, Y., Liang, C., 2016. Blockchain application and outlook in the banking industry. Financ. Innov. 2, 1–12.
Guo, J., Lu, L., Li, J., 2024. Smart contract vulnerability detection based on multi-scale encoders. Electronics 13 (3), 489.
Han, D., Li, Q., Zhang, L., Xu, T., 2022. A smart contract vulnerability detection model based on graph neural networks. In: 2022 4th International Conference on Frontiers Technology of Information and Computer. ICFTIC, IEEE, pp. 834–837.
He, F., Li, F., Liang, P., 2024. Enhancing smart contract security: Leveraging pre-trained language models for advanced vulnerability detection. IET Blockchain.
He, D., Wu, R., Li, X., Chan, S., Guizani, M., 2023. Detection of vulnerabilities of blockchain smart contracts. IEEE Internet Things J.
Hellendoorn, V.J., Sutton, C., Singh, R., Maniatis, P., Bieber, D., 2019. Global relational models of source code. In: International Conference on Learning Representations.
Hu, H., Bai, Q., Xu, Y., 2022. Scsguard: Deep scam detection for ethereum smart contracts. In: IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops. INFOCOM WKSHPS, IEEE, pp. 1–6.
Hu, X., Gao, Z., Xia, X., Lo, D., Yang, X., 2021. Automating user notice generation for smart contract functions. In: 2021 36th IEEE/ACM International Conference on Automated Software Engineering. ASE, IEEE, pp. 5–17.
Hu, W., Gu, Z., Xie, Y., Wang, L., Tang, K., 2019. Chinese text classification based on neural networks and word2vec. In: 2019 IEEE Fourth International Conference on Data Science in Cyberspace. DSC, pp. 284–291. http://dx.doi.org/10.1109/DSC.2019.00050.
Hu, X., Li, G., Xia, X., Lo, D., Jin, Z., 2018a. Deep code comment generation. In: Proceedings of the 26th Conference on Program Comprehension. pp. 200–210.

Hu, X., Li, G., Xia, X., Lo, D., Lu, S., Jin, Z., 2018b. Summarizing source code with transferred api knowledge.

Idé, T., 2018. Collaborative anomaly detection on blockchain from noisy sensor data. In: 2018 IEEE International Conference on Data Mining Workshops. ICDMW, IEEE, pp. 120–127.

Inamdar, A., Bhagtani, A., Bhatt, S., Shetty, P.M., 2019. Predicting cryptocurrency value using sentiment analysis. In: 2019 International Conference on Intelligent Computing and Control Systems. ICCS, IEEE, pp. 932–934.

Iyer, S., Konstas, I., Cheung, A., Zettlemoyer, L., 2016. Summarizing source code using a neural attention model. In: 54th Annual Meeting of the Association for Computational Linguistics 2016. Association for Computational Linguistics, pp. 2073–2083.

Jain, V.K., Tripathi, M., 2024. An integrated deep learning model for ethereum smart contract vulnerability detection. Int. J. Inf. Secur. 23 (1), 557–575.

Jay, P., Kalariya, V., Parmar, P., Tanwar, S., Kumar, N., Alazab, M., 2020. Stochastic neural networks for cryptocurrency price prediction. IEEE Access 8, 82804–82818.

Jayabalan, J., N., J., 2021. A study on distributed consensus protocols and algorithms: The backbone of blockchain networks. In: 2021 International Conference on Computer Communication and Informatics. ICCCI, pp. 1–10. http://dx.doi.org/10.1109/ICCCI50826.2021.9402318.

Jeon, S., Lee, G., Kim, H., Woo, S.S., 2021. Smartcondetect: Highly accurate smart contract code vulnerability detection mechanism using bert. In: KDD Workshop on Programming Language Processing.

Jiang, F., Cao, Y., Xiao, J., Yi, H., Lei, G., Liu, M., Deng, S., Wang, H., 2022. VDDL: A deep learning-based vulnerability detection model for smart contracts. In: International Conference on Machine Learning for Cyber Security. Springer, pp. 72–86.

Jie, W., Chen, Q., Wang, J., Koe, A.S.V., Li, J., Huang, P., Wu, Y., Wang, Y., 2023. A novel extended multimodal AI framework towards vulnerability detection in smart contracts. Inform. Sci. 636, 118907.

Ju, C., Gang, L., Sun, D., 2020. The application of blockchain in intelligent power data exchange. In: Proceedings of the 2020 4th International Conference on Electronic Information Technology and Computer Engineering. http://dx.doi.org/10.1145/3443467.3443725.

Khan, D., Jung, L.T., Hashmani, M., 2021. Systematic literature review of challenges in blockchain scalability. Appl. Sci. http://dx.doi.org/10.3390/app11209372.

Khaniki, M.A.L., Manthouri, M., 2024. Enhancing price prediction in cryptocurrency using transformer neural network and technical indicators. ArXiv preprint arXiv:2403.03606.

Khezr, S., Moniruzzaman, M., Yassine, A., Benlamri, R., 2019. Blockchain technology in healthcare: A comprehensive review and directions for future research. Appl. Sci. 9 (9), 1736.

Kim, J., Kim, S., Wimmer, H., Liu, H., 2021a. A cryptocurrency prediction model using LSTM and GRU algorithms. In: 2021 IEEE/ACIS 6th International Conference on Big Data, Cloud Computing, and Data Science. BCD, IEEE, pp. 37–44.

Kim, J., Nakashima, M., Fan, W., Wuthier, S., Zhou, X., Kim, I., Chang, S.-Y., 2021b. Anomaly detection based on traffic monitoring for secure blockchain networking. In: 2021 IEEE International Conference on Blockchain and Cryptocurrency. ICBC, IEEE, pp. 1–9.

Kim, S., Ryu, S., 2020. Analysis of blockchain smart contracts: Techniques and insights. In: 2020 IEEE Secure Development (SecDev). pp. 65–73. http://dx.doi.org/10.1109/SecDev45635.2020.00026.

Kim, G., Shin, D.-H., Choi, J.G., Lim, S., 2022. A deep learning-based cryptocurrency price prediction model that uses on-chain data. IEEE Access 10, 56232–56248.

Kneissler, A., Oelbracht, S., 2023. Addressing the practical challenges of implementing blockchain in engineering and manufacturing. AHFE Int. http://dx.doi.org/10.54941/ahfe1004313.

Koltun, V., Yamshchikov, I.P., 2023. Pump it: Twitter sentiment analysis for cryptocurrency price prediction. Risks 11 (9), 159.

Kshetri, N., 2017. Can blockchain strengthen the internet of things? IT Prof. 19 (4), 68–72.

Kushwaha, S.S., Joshi, S., Singh, D., Kaur, M., Lee, H.-N., 2022. Ethereum smart contract analysis tools: A systematic review. IEEE Access PP, 1. http://dx.doi.org/10.1109/ACCESS.2022.3169902.

Lahmiri, S., Bekiros, S., 2019. Cryptocurrency forecasting with deep learning chaotic neural networks. Chaos Solitons Fractals 118, 35–40.

Laroiya, C., Saxena, D., Komalavalli, C., 2020. Applications of blockchain technology. Handb. Res. Blockchain Technol. http://dx.doi.org/10.1016/b978-0-12-819816-2.00009-5.

Lê Hông, B., Lê Đú'c, T., Đoàn Minh, T., Trân Tuân, D., Phan Thê, D., Pham Văn, H., 2023. Contextual language model and transfer learning for reentrancy vulnerability detection in smart contracts. In: Proceedings of the 12th International Symposium on Information and Communication Technology. pp. 739–745.

LeClair, A., Jiang, S., McMillan, C., 2019. A neural model for generating natural language summaries of program subroutines. In: 2019 IEEE/ACM 41st International Conference on Software Engineering. ICSE, IEEE, pp. 795–806.

Li, T., Fang, Y., Lu, Y., Yang, J., Jian, Z., Wan, Z., Li, Y., 2022. SmartVM: A smart contract virtual machine for fast on-chain DNN computations. IEEE Trans. Parallel Distrib. Syst. PP, 1. http://dx.doi.org/10.1109/TPDS.2022.3177405.

Li, K., Li, H., Hou, H., Li, K., Chen, Y., 2017. Proof of vote: A high-performance consensus protocol based on vote mechanism & consortium blockchain. In: 2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS). pp. 466–473. http://dx.doi.org/10.1109/HPCC-SmartCity-DSS.2017.61.

Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., Hoi, S.C.H., 2021a. Align before fuse: Vision and language representation learning with momentum distillation. Adv. Neural Inf. Process. Syst. 34, 9694–9705.

Li, H., Wang, T., Qiao, Z., Yang, B., Gong, Y., Wang, J., Qiu, G., 2021b. Blockchain-based searchable encryption with efficient result verification and fair payment. J. Inf. Secur. Appl. 58, 102791. http://dx.doi.org/10.1016/J.JISA.2021.102791.

Liang, Y., Feng, K., Ren, Z., 2023. Human activity recognition based on transformer via smart-phone sensors. In: 2023 IEEE 3rd International Conference on Computer Communication and Artificial Intelligence. CCAI, pp. 267–271. http://dx.doi.org/10.1109/CCAI57533.2023.10201297.

Liang, W., Xiao, L., Zhang, K., Tang, M., He, D., Li, K.-C., 2021. Data fusion approach for collaborative anomaly intrusion detection in blockchain-based systems. IEEE Internet Things J. 9 (16), 14741–14751.

Liu, K., Kim, D., Bissyandé, T.F., Yoo, S., Le Traon, Y., 2018. Mining fix patterns for findbugs violations. IEEE Trans. Softw. Eng. 47 (1), 165–188.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019. RoBERTa: A robustly optimized BERT pretraining approach. arXiv:1907.11692.

Liu, Z., Qian, P., Wang, X., Zhuang, Y., Qiu, L., Wang, X., 2021. Combining graph neural networks with expert knowledge for smart contract vulnerability detection. IEEE Trans. Knowl. Data Eng. 35 (2), 1296–1310.

Liu, L., Tsai, W.-T., Bhuiyan, M.Z.A., Peng, H., Liu, M., 2022. Blockchain-enabled fraud discovery through abnormal smart contract detection on Ethereum. Future Gener. Comput. Syst. 128, 158–166.

Liu, Y., Xiao, G., Chen, W., Zheng, Z., 2023. A LSTM and GRU-based hybrid model in the cryptocurrency price prediction. In: International Conference on Blockchain and Trustworthy Systems. Springer, pp. 32–43.

Luo, F., Luo, R., Chen, T., Qiao, A., He, Z., Song, S., Jiang, Y., Li, S., 2024. Scvhunter: Smart contract vulnerability detection based on heterogeneous graph attention network. In: Proceedings of the IEEE/ACM 46th International Conference on Software Engineering. pp. 1–13.

Luu, L., Chu, D.-H., Olickel, H., Saxena, P., Hobor, A., 2016. Making smart contracts smarter. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 254–269.

Ma, R., Wang, X., Ding, J., 2013. Multilevel core-sets based aggregation clustering algorithm. J. Softw. 24 (3), 490–506.

McWaters, R.J., Galaski, R., Chatterjee, S., 2016. The future of financial infrastructure: An ambitious look at how blockchain can reshape financial services. In: World Economic Forum. vol. 49, pp. 368–376.

Mendoza, A., Gu, G., 2018. Mobile application web api reconnaissance: Web-to-mobile inconsistencies & vulnerabilities. In: 2018 IEEE Symposium on Security and Privacy. SP, IEEE, pp. 756–769.

Merity, S., Keskar, N.S., Socher, R., 2017. Regularizing and optimizing LSTM language models. ArXiv preprint arXiv:1708.02182.

Mettler, M., 2016. Blockchain technology in healthcare: The revolution starts here. In: 2016 IEEE 18th International Conference on E-Health Networking, Applications and Services (Healthcom). IEEE, pp. 1–3.

Mittal, R., Arora, S., Bhatia, M., 2018. Automated cryptocurrencies prices prediction using machine learning. ICTACT J. Soft Comput. 8 (04), 4.

Morishima, S., 2021. Scalable anomaly detection in blockchain using graphics processing unit. Comput. Electr. Eng. 92, 107087.

Mou, L., Li, G., Zhang, L., Wang, T., Jin, Z., 2016. Convolutional neural networks over tree structures for programming language processing. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, no. 1.

Mueller, B., 2017. Mythril: security analysis tool for evm bytecode.

Murray, K., Rossi, A., Carraro, D., Visentin, A., 2023. On forecasting cryptocurrency prices: A comparison of machine learning, deep learning, and ensembles. Forecasting 5 (1), 196–209.

Nakamoto, S., 2008. Bitcoin: A peer-to-peer electronic cash system.

Naseer, S., Saleem, Y., Khalid, S., Bashir, M.K., Han, J., Iqbal, M.M., Han, K., 2018. Enhanced network anomaly detection based on deep neural networks. IEEE Access 6, 48231–48246.

Nasrulin, B., Muzammal, M., Qu, Q., 2018. A robust spatio-temporal verification protocol for blockchain. pp. 52–67. http://dx.doi.org/10.1007/978-3-030-02922-7-4.

Nguyen, H.H., Nguyen, N.-M., Xie, C., Ahmadi, Z., Kudendo, D., Doan, T.-N., Jiang, L., 2023. MANDO-HGT: Heterogeneous graph transformers for smart contract vulnerability detection. In: 2023 IEEE/ACM 20th International Conference on Mining Software Repositories. MSR, IEEE, pp. 334–346.

Nikolić, I., Kolluri, A., Sergey, I., Saxena, P., Hobor, A., 2018. Finding the greedy, prodigal, and suicidal contracts at scale. In: Proceedings of the 34th Annual Computer Security Applications Conference. pp. 653–663.

Ofori-Boateng, D., Dominguez, I.S., Akcora, C., Kantarcioglu, M., Gel, Y.R., 2021. Topological anomaly detection in dynamic multilayer blockchain networks. In: Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part I 21. Springer, pp. 788–804.

Oikonomopoulos, S., Tzafilkou, K., Karapiperis, D., Verykios, V., 2022. Cryptocurrency price prediction using social media sentiment analysis. In: 2022 13th International Conference on Information, Intelligence, Systems & Applications. IISA, IEEE, pp. 1–8.

Patel, M.M., Tanwar, S., Gupta, R., Kumar, N., 2020. A deep learning-based cryptocurrency price prediction scheme for financial institutions. J. Inf. Secur. Appl. 55, 102583.

Pathak, S., Kakkar, A., 2020. Cryptocurrency price prediction based on historical data and social media sentiment analysis. In: Innovations in Computer Science and Engineering: Proceedings of 7th ICICSE. Springer, pp. 47–55.

Penmetsa, S., Vemula, M., 2023. Cryptocurrency price prediction with LSTM and transformer models leveraging momentum and volatility technical indicators. In: 2023 IEEE 3rd International Conference on Data Science and Computer Application. ICDSCA, IEEE, pp. 411–416.

Pilipchenko, A., Kuzminsky, V., Chumachenko, O., 2021. Using methods of technical analysis to forecast the cryptocurrency market. Scientific Notes the University KROK http://dx.doi.org/10.31732/2663-2209-2021-64-28-35.

Prajapati, P., 2020. Predictive analysis of Bitcoin price considering social sentiments. ArXiv preprint arXiv:2001.10343.

Prause, G., 2019. Smart contracts for smart supply chains. IFAC-PapersOnLine http://dx.doi.org/10.1016/j.ifacol.2019.11.582.

Pronchakov, Y., Bugaienko, O., 2019. Methods of forecasting the prices of cryptocurrency on the financial markets. Technol. Transf.: Innov. Solutions Soc. Sci. Humanit. 13–16.

Qian, P., Liu, Z., He, Q., Zimmermann, R., Wang, X., 2020. Towards automated reentrancy detection for smart contracts based on sequential models. IEEE Access 8, 19685–19695.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al., 2018. Improving Language Understanding by Generative Pre-Training. OpenAI.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J., 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv:1910.10683.

Rana, N., Dwivedi, Y.K., Hughes, D.L., 2021. Analysis of challenges for blockchain adoption within the Indian public sector: an interpretive structural modelling approach. Inf. Technol. People 35, 548–576. http://dx.doi.org/10.1108/ITP-07-2020-0460.

Regnath, E., Steinhorst, S., 2018. LeapChain: Efficient blockchain verification for embedded IoT. In: 2018 IEEE/ACM International Conference on Computer-Aided Design. ICCAD, pp. 1–8. http://dx.doi.org/10.1145/3240765.3240820.

Ressi, D., Romanello, R., Piazza, C., Rossi, S., 2024. AI-enhanced blockchain technology: A review of advancements and opportunities. J. Netw. Comput. Appl. 103858.

Rizzo, M., Ressi, D., Gasparetto, A., Rossi, S., 2024. A comparison of machine learning techniques for ethereum smart contract vulnerability detection.

Ruj, S., 2024. Zero-knowledge proofs for blockchains. In: 2024 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S). pp. 67–68. http://dx.doi.org/10.1109/DSN-S60304.2024.00028.

Sanjay Rai, G., Goyal, S., Chatterjee, P., 2023. Anomaly detection in blockchain using machine learning. In: Computational Intelligence for Engineering and Management Applications: Select Proceedings of CIEMA 2022. Springer, pp. 487–499.

Sanju, T., Liyanage, H., Bandara, K., Kandakkulama, D., Silva, D.D., Perera, J., 2023. Stock-crypto-app 2013 recommendation system for stock and cryptocurrency market using cutting edge machine learning technology. Int. Res. J. Innov. Eng. Technol. http://dx.doi.org/10.47001/irjiet/2023.709012.

Saravanan, S.S., Luo, T., Van Ngo, M., 2023. TSI-gan: Unsupervised time series anomaly detection using convolutional cycle-consistent generative adversarial networks. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, pp. 39–54.

Sayadi, S., Rejeb, S.B., Choukair, Z., 2019. Anomaly detection model over blockchain electronic transactions. In: 2019 15th International Wireless Communications & Mobile Computing Conference. IWCMC, IEEE, pp. 895–900.

Shafay, M., Ahmad, R.W., Salah, K., Yaqoob, I., Jayaraman, R., Omar, M., 2023. Blockchain for deep learning: review and open challenges. Clust. Comput. 26 (1), 197–221.

Shayegan, M.J., Sabor, H.R., Uddin, M., Chen, C.-L., 2022. A collective anomaly detection technique to detect crypto wallet frauds on bitcoin network. Symmetry 14 (2), 328.

Sheng, Z., Song, L., Wang, Y., 2025. Dynamic feature fusion: Combining global graph structures and local semantics for blockchain fraud detection. arXiv:2501.02032. URL https://arxiv.org/abs/2501.02032.

Shi, C., Cai, B., Zhao, Y., Gao, L., Sood, K., Xiang, Y., 2023. CoSS: leveraging statement semantics for code summarization. IEEE Trans. Softw. Eng.

Signorini, M., Pontecorvi, M., Kanoun, W., Di Pietro, R., 2018a. Advise: anomaly detection tool for blockchain systems. In: 2018 IEEE World Congress on Services. SERVICES, IEEE, pp. 65–66.

Signorini, M., Pontecorvi, M., Kanoun, W., Di Pietro, R., 2018b. Bad: blockchain anomaly detection. ArXiv preprint arXiv:1807.03833.

Singh, S., Bhat, M., 2024. Transformer-based approach for ethereum price prediction using crosscurrency correlation and sentiment analysis. ArXiv preprint arXiv:2401.08077.

Son, Y., Vohra, S., Vakkalagadda, R., Zhu, M., Hirde, A., Kumar, S., Rajaram, A., 2022. Using transformers and deep learning with stance detection to forecast cryptocurrency price movement. In: 2022 13th International Conference on Information and Communication Technology Convergence. ICTC, IEEE, pp. 1–6.

Song, A., Seo, E., Kim, H., 2023a. Anomaly VAE-transformer: A deep learning approach for anomaly detection in decentralized finance. IEEE Access PP, 1. http://dx.doi.org/10.1109/ACCESS.2023.3313448.

Song, A., Seo, E., Kim, H., 2023b. Anomaly vae-transformer: a deep learning approach for anomaly detection in decentralized finance. IEEE Access.

Sridhar, S., Sanagavarapu, S., 2021. Multi-head self-attention transformer for dogecoin price prediction. In: 2021 14th International Conference on Human System Interaction. HSI, IEEE, pp. 1–6.

Steinert, L., Herff, C., 2018. Predicting altcoin returns using social media. PLoS One 13 (12), e0208119.

Sui, J., Chu, L., Bao, H., 2023. An opcode-based vulnerability detection of smart contracts. Appl. Sci. 13 (13), 7721.

Sun, J., Jia, Y., Wang, Y., Tian, Y., Zhang, S., 2025. Ethereum fraud detection via joint transaction language model and graph representation learning. Inf. Fusion 120, 103074.

Sun, G., Jiang, C., Shen, J., Zhang, Y., 2023a. SCOBERT: A pre-trained BERT for smart contract vulnerability detection. In: 2023 8th International Conference on Data Science in Cyberspace. DSC, pp. 24–30. http://dx.doi.org/10.1109/DSC59305.2023.00014.

Sun, X., Liu, M., Sima, Z., 2020. A novel cryptocurrency price trend forecasting model based on LightGBM. Financ. Res. Lett. 32, 101084.

Sun, X., Tu, L., Zhang, J., Cai, J., Li, B., Wang, Y., 2023b. ASSBert: Active and semi-supervised bert for smart contract vulnerability detection. J. Inf. Secur. Appl. 73, 103423.

Swan, M., 2015. Blockchain: Blueprint For a New Economy. O'Reilly Media, Inc.

Tang, X., Du, Y., Lai, A., Zhang, Z., Shi, L., 2023. Deep learning-based solution for smart contract vulnerabilities detection. Sci. Rep. 13 (1), 20106.

Tang, X., Zhou, K., Cheng, J., Li, H., Yuan, Y., 2021. The vulnerabilities in smart contracts: A survey. In: Advances in Artificial Intelligence and Security: 7th International Conference, ICAIS 2021, Dublin, Ireland, July 19–23, 2021, Proceedings, Part III 7. Springer, pp. 177–190.

Tapscott, D., Tapscott, A., 2016. Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World. Penguin.

Tikhomirov, S., Voskresenskaya, E., Ivanitskiy, I., Takhaviev, R., Marchenko, E., Alexandrov, Y., 2018. Smartcheck: Static analysis of ethereum smart contracts. In: Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain. pp. 9–16.

Trautmann, L., Lasch, R., 2020. Smart contracts in the context of procure-to-pay. Smart Sustain. Supply Chain Logist. – Trends, Challenges, Methods Best Pr. http://dx.doi.org/10.1007/978-3-030-61947-3-1.

Tsankov, P., Dan, A., Drachsler-Cohen, D., Gervais, A., Buenzli, F., Vechev, M., 2018. Securify: Practical security analysis of smart contracts. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. pp. 67–82.

Vani, S., Doshi, M., Nanavati, A., Kundu, A., 2022. Vulnerability analysis of smart contracts. ArXiv preprint arXiv:2212.07387.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.

Voronov, T., Raz, D., Rottenstreich, O., 2021. Scalable blockchain anomaly detection with sketches. In: 2021 IEEE International Conference on Blockchain (Blockchain). IEEE, pp. 1–10.

Wan, Y., Zhao, Z., Yang, M., Xu, G., Ying, H., Wu, J., Yu, P.S., 2018. Improving automatic source code summarization via deep reinforcement learning. In: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering. pp. 397–407.

Wang, Y., Chen, X., Huang, Y., Zhu, H.-N., Bian, J., 2022. An empirical study on real bug fixes in smart contracts projects. http://dx.doi.org/10.2139/ssrn.4250240, ArXiv arXiv:2210.11990.

Wang, Z., Ni, A., Tian, Z., Wang, Z., Gong, Y., 2024. Research on blockchain abnormal transaction detection technology combining CNN and transformer structure. Comput. Electr. Eng. 116, 109194.

Wang, S., Pathania, A., Mitra, T., 2020. Neural network inference on mobile socs. IEEE Des. Test 37 (5), 50–57. http://dx.doi.org/10.1109/MDAT.2020.2968258.

Wang, Z., Wu, W., Zeng, C., Yao, J., Yang, Y., Xu, H., 2023. Graph neural networks enhanced smart contract vulnerability detection of educational blockchain. arXiv:2303.04477, URL https://arxiv.org/abs/2303.04477.

Wang, X., Xu, F., 2023. The value of smart contract in trade finance. Manuf. Serv. Oper. Manag. 25 (6), 2056–2073.

Wei, B., Li, Y., Li, G., Xia, X., Jin, Z., 2020. Retrieve and refine: exemplar-based neural comment generation. In: Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering. pp. 349–360.

Wei, B., Li, G., Xia, X., Fu, Z., Jin, Z., 2019. Code generation as a dual task of code summarization. Adv. Neural Inf. Process. Syst. 32.

Weng, J., Weng, J., Zhang, J., Li, M., Zhang, Y., Luo, W., 2019. Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive. IEEE Trans. Dependable Secur. Comput. 18 (5), 2438–2455.

Wołk, K., 2020. Advanced social media sentiment analysis for short-term cryptocurrency price prediction. Expert Syst. 37 (2), e12493.

Wood, G., et al., 2014. Ethereum: A secure decentralised generalised transaction ledger. Ethereum Proj. Yellow Pap. 151 (2014), 1–32.

Wu, H., Dong, H., He, Y., Duan, Q., 2023. Smart contract vulnerability detection based on hybrid attention mechanism model. Appl. Sci. 13 (2), 770.

Wu, H., Zhang, Z., Wang, S., Lei, Y., Lin, B., Qin, Y., Zhang, H., Mao, X., 2021. Peculiar: Smart contract vulnerability detection based on crucial data flow graph and pre-training techniques. In: 2021 IEEE 32nd International Symposium on Software Reliability Engineering. ISSRE, IEEE, pp. 378–389.

Xu, P., Lee, J., Barth, J.R., Richey, R.G., 2021. Blockchain as supply chain technology: considering transparency and security. Int. J. Phys. Distrib. Logist. Manage. 51 (3), 305–324.

Xu, G., Liu, L., Dong, J., 2023a. Vulnerability detection of ethereum smart contract based on SolBERT-BiGRU-attention hybrid neural model. CMES Comput. Model. Eng. Sci. 137 (1).

Xu, J., Yin, K., Liu, L., 2020. State-continuity approximation of markov decision processes via finite element methods for autonomous system planning. IEEE Robot. Autom. Lett. 5 (4), 5589–5596.

Xu, C., Zhang, S., Zhu, L., Shen, X., Zhang, X., 2023b. Illegal accounts detection on ethereum using heterogeneous graph transformer networks. In: International Conference on Information and Communications Security. Springer, pp. 665–680.

Xu, P., Zhu, X., Clifton, D.A., 2023c. Multimodal learning with transformers: A survey. IEEE Trans. Pattern Anal. Mach. Intell.

Yan, X., Wang, S., Gai, K., 2022. A semantic analysis-based method for smart contract vulnerability. In: 2022 IEEE 8th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security. IDS, IEEE, pp. 23–28.

Yang, Z., Dai, M., Guo, J., 2022a. Formal modeling and verification of smart contracts with spin. Electronics 11 (19), 3091.

Yang, Z., Keung, J., Yu, X., Gu, X., Wei, Z., Ma, X., Zhang, M., 2021. A multi-modal transformer-based code summarization approach for smart contracts. In: 2021 IEEE/ACM 29th International Conference on Program Comprehension. ICPC, IEEE, pp. 1–12.

Yang, L., Liu, X.-Y., Li, X., Li, Y., 2019. Price prediction of cryptocurrency: an empirical study. In: Smart Blockchain: Second International Conference, SmartBlock 2019, Birmingham, UK, October 11–13, 2019, Proceedings 2. Springer, pp. 130–139.

Yang, H., Zhang, J., Gu, X., Cui, Z., 2022b. Smart contract vulnerability detection based on abstract syntax tree. In: 2022 8th International Symposium on System Security, Safety, and Reliability. ISSSR, IEEE, pp. 169–170.

Yang, Z., Zhu, W., 2023. Improvement and optimization of vulnerability detection methods for ethernet smart contracts. IEEE Access.

Yu, C., Yang, W., Xie, F., He, J., 2022. Technology and security analysis of cryptocurrency based on blockchain. Complex. 2022, 5835457:1–5835457:15. http://dx.doi.org/10.1155/2022/5835457.

Yu, J., Zhang, Q., Wang, L., 2019. Design of optimal hybrid controller for multi-phase batch processes with interval time varying delay. IEEE Access 7, 164029–164043.

Zhang, H., Shafiq, M.O., 2024. Survey of transformers and towards ensemble learning using transformers for natural language processing. J. Big Data 11 (1), 25.

Zhang, L., Wang, J., Wang, W., Jin, Z., Su, Y., Chen, H., 2022a. Smart contract vulnerability detection combined with multi-objective detection. Comput. Netw. 217, 109289.

Zhang, L., Wang, J., Wang, W., Jin, Z., Zhao, C., Cai, Z., Chen, H., 2022b. A novel smart contract vulnerability detection method based on information graph and ensemble learning. Sensors 22 (9), 3581.

Zhang, R., Xue, R., Liu, L., 2019. Security and privacy on blockchain. ACM Comput. Surv. 52 (3), 1–34.

Zhang, J., Ye, A., Chen, J., Zhang, Y., Yang, W., 2023. CSFL: Cooperative security aware federated learning model using the blockchain. Comput. J. 67 (4), 1298–1308. http://dx.doi.org/10.1093/comjnl/bxad060, arXiv:https://academic.oup.com/comjnl/article-pdf/67/4/1298/57295832/bxad060.pdf.

Zhao, H., Crane, M., Bezbradica, M., 2022. Attention! Transformer with Sentiment on Cryptocurrencies Price Prerediction. SciTePress.

Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H., 2017. An overview of blockchain technology: Architecture, consensus, and future trends. In: 2017 IEEE International Congress on Big Data (BigData Congress). IEEE, pp. 557–564.

Zhou, X., Chen, Y., Guo, H., Chen, X., Huang, Y., 2023. Security code recommendations for smart contract. In: 2023 IEEE International Conference on Software Analysis, Evolution and Reengineering. SANER, pp. 190–200. http://dx.doi.org/10.1109/SANER56733.2023.00027.

Zhou, Q., Dang, X., Huo, D., Ruan, Q., Li, C., Wang, Y., Xu, Z., 2022. LogBlock: An anomaly detection method on permissioned blockchain based on log-block sequence. In: 2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta). IEEE, pp. 1008–1015.