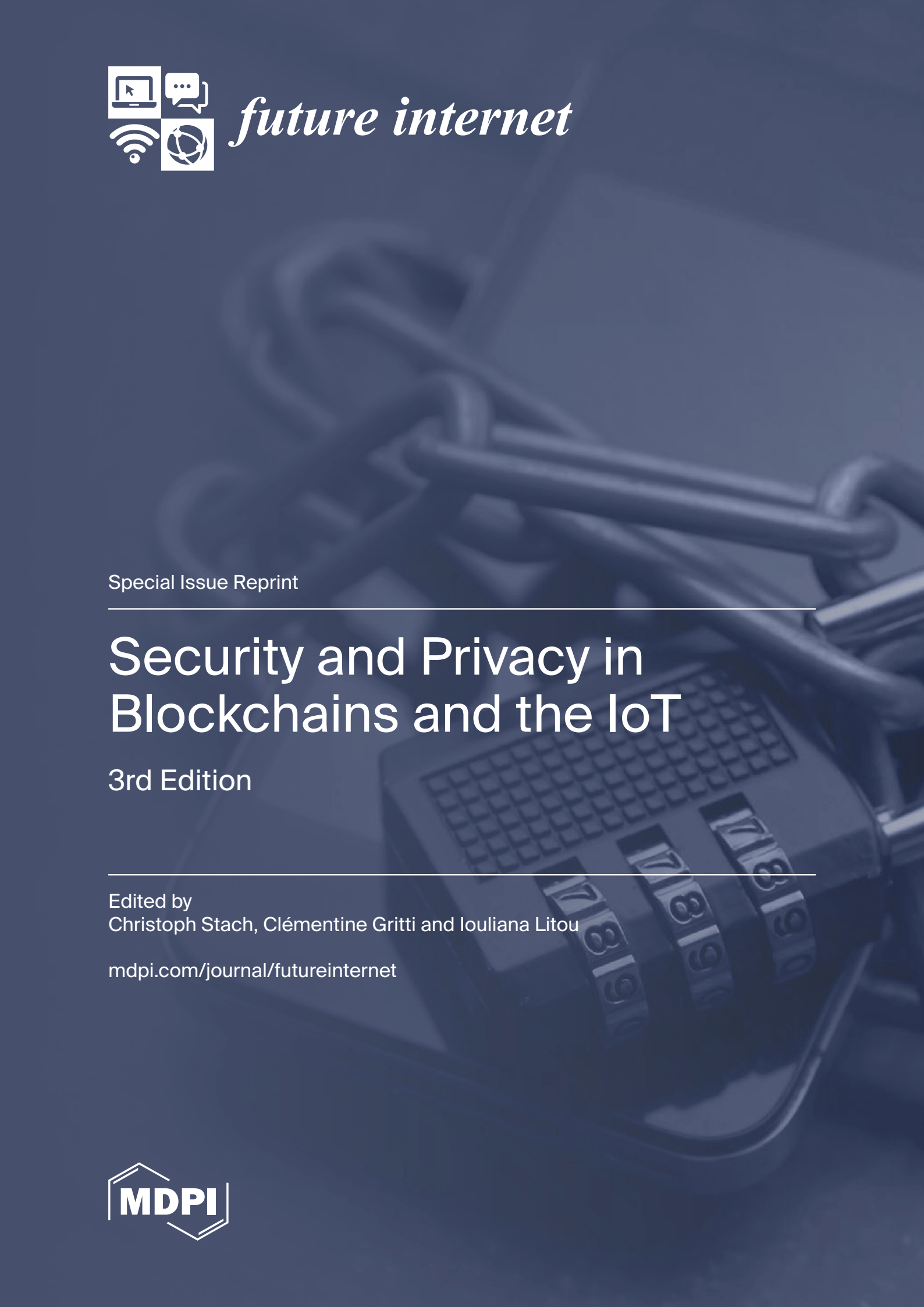
future internet



Special Issue Reprint

Security and Privacy in Blockchains and the IoT

3rd Edition

Edited by
Christoph Stach, Clémentine Gritti and Iouliana Litou

mdpi.com/journal/futureinternet

MDPI

Security and Privacy in Blockchains and the IoT—3rd Edition

Security and Privacy in Blockchains and the IoT—3rd Edition

Guest Editors

Christoph Stach
Clémentine Gritti
Iouliana Litou

MDPI

Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Guest Editors

Christoph Stach
University of Stuttgart
Stuttgart
Germany

Clémentine Gritti
CITI Lab
INSA Lyon
Villeurbanne
France

Iouliana Litou
Department of Informatics
Athens University of
Economics and Business
Athens
Greece

Editorial Office
MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of the Special Issue, published open access by the journal *Future Internet* (ISSN 1999-5903), freely accessible at: https://www.mdpi.com/journal/futureinternet/special_issues/I7D8TJCVP9.

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, A.A.; Lastname, B.B. Article Title. *Journal Name* **Year**, *Volume Number*, Page Range.

ISBN 978-3-7258-8975-4 (Hbk)
ISBN 978-3-7258-8976-1 (PDF)
https://doi.org/10.3390/books978-3-7258-8976-1

Cover image courtesy of Christoph Stach

© 2026 by the authors. Articles in this reprint are Open Access and distributed under the Creative Commons Attribution (CC BY) license. The reprint as a whole is distributed by MDPI under the terms and conditions of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) license (https://creativecommons.org/licenses/by-nc-nd/4.0/).

Contents

About the Editors . . . . . **vii**

Preface . . . . . **ix**

Christoph Stach, Clémentine Gritti and Iouliana Litou
Special Issue on Security and Privacy in Blockchains and the IoT—3rd Edition
Reprinted from: *Future Internet* **2026**, *18*, 90, https://doi.org/10.3390/fi18020090 . . . . . **1**

Md Kamrul Siam, Bilash Saha, Md Mehedi Hasan, Md Jobair Hossain Faruk, Nafisa Anjum, Sharaban Tahora, et al.
Securing Decentralized Ecosystems: A Comprehensive Systematic Review of Blockchain Vulnerabilities, Attacks, and Countermeasures and Mitigation Strategies
Reprinted from: *Future Internet* **2025**, *17*, 183, https://doi.org/10.3390/fi17040183 . . . . . **9**

Mohammad Jaminur Islam, Saminur Islam, Mahmud Hossain, Shahid Noor and S. M. Riazul Islam
Securing Blockchain Systems: A Layer-Oriented Survey of Threats, Vulnerability Taxonomy, and Detection Methods
Reprinted from: *Future Internet* **2025**, *17*, 205, https://doi.org/10.3390/fi17050205 . . . . . **50**

Yunus Kareem, Djamel Djenouri and Essam Ghadafi
A Survey on Emerging Blockchain Technology Platforms forSecuring the Internet of Things
Reprinted from: *Future Internet* **2024**, *16*, 285, https://doi.org/10.3390/fi16080285 . . . . . **86**

Talgar Bayan, Adnan Yazici and Richard Banach
Permissionless Blockchain Recent Trends, Privacy Concerns, Potential Solutions and Secure Development Lifecycle
Reprinted from: *Future Internet* **2025**, *17*, 547, https://doi.org/10.3390/fi17120547 . . . . . **105**

Víctor Pérez-Cano and Francisco Jurado
Fraud Detection in Cryptocurrency Networks—An Exploration Using Anomaly Detection and Heterogeneous Graph Transformers
Reprinted from: *Future Internet* **2025**, *17*, 44, https://doi.org/10.3390/fi17010044 . . . . . **144**

Dominic Lightbody, Duc-Minh Ngo, Andriy Temko, Colin C. Murphy and Emanuel Popovici
Dragon_Pi: IoT Side-Channel Power Data Intrusion Detection Dataset and Unsupervised Convolutional Autoencoder for Intrusion Detection
Reprinted from: *Future Internet* **2024**, *16*, 88, https://doi.org/10.3390/fi16030088 . . . . . **167**

Zhengyang Fan, Wanru Li, Kathryn Blackmond Laskey and Kuo-Chu Chang
Investigation of Phishing Susceptibility with Explainable Artificial Intelligence
Reprinted from: *Future Internet* **2024**, *16*, 31, https://doi.org/10.3390/fi16010031 . . . . . **205**

Saleh Alharbi
QuantumTrust-FedChain: A Blockchain-Aware Quantum-Tuned Federated Learning System for Cyber-Resilient Industrial IoT in 6G
Reprinted from: *Future Internet* **2025**, *17*, 493, https://doi.org/10.3390/fi17110493 . . . . . **223**

Yazan Aref and Abdelkader Ouda
HSM4SSL: Leveraging HSMs for Enhanced Intra-Domain Security
Reprinted from: *Future Internet* **2024**, *16*, 148, https://doi.org/10.3390/fi16050148 . . . . . **253**

Daniele Marletta, Alessandro Midolo and Emiliano Tramontana
A Blockchain-Based Strategy for Certifying Timestamps in a Distributed Healthcare Emergency Response Systems
Reprinted from: *Future Internet* **2025**, *17*, 210, https://doi.org/10.3390/fi17050210 . . . . . **275**

Han-Wen Huang, Chih-Hung Shih, Chen-Yu Li and Hao-Yung Teng
A Blockchain-Based Framework for OSINT Evidence Collection and Identification
Reprinted from: *Future Internet* **2025**, *17*, 551, https://doi.org/10.3390/fi17120551 . . . . . **298**

About the Editors

Christoph Stach

Christoph Stach is a substitute professor at the Institute for Parallel and Distributed Systems and the Substitute Head of the Chair of Data Engineering at the University of Stuttgart, Germany, where he also leads the Information Systems and Applications working area and various academic committees. Since earning his doctorate in 2017, specializing in data security and privacy, his research interests focus on trustworthy, demand-oriented data provisioning for data-intensive applications such as the Internet of Things, Industry 4.0, and eHealth. His expertise is demonstrated by his leadership in five large-scale industrial projects addressing automotive digitalization, autonomous IoT systems, and privacy protection in data streams. He is also dedicated to teaching, conveying the insights he has gained from both theoretical research and practical application to his students. Dr. Stach's academic record features over 70 peer-reviewed publications, four of which received a Best Paper Award. Additionally, he has served on more than 50 international conference program committees and acted as a Guest Editor for two books and three Special Issues.

Clémentine Gritti

Clémentine Gritti received an M.Sc. degree in computer science from Grenoble Alpes University, France, in 2012, and a Ph.D. degree in computer science from the University of Wollongong, Australia, in 2017. She was a Senior Lecturer at the University of Canterbury, New Zealand, between 2020 and 2023. Thereafter, she was a research fellow at Eurecom, France. In 2024, she joined INSA Lyon, France, as a Professor Junior. Her current research interests include the design and evaluation of public-key cryptographic protocols for applied security and privacy in various contexts, such as Federated Learning, Internet of Things, and blockchain technologies.

Iouliana Litou

Iouliana Litou is a computer scientist and researcher primarily focused on social networks, information dissemination, influence maximization, and misinformation analysis. She has pursued her academic career at the Athens University of Economics and Business (AUEB), Department of Informatics, where she completed her B.Sc. and M.Sc. degrees in Computer Science and later completed her Ph.D. in Computer Science under the supervision of Prof. Vana Kalogeraki. Her Ph.D. research centers on Participatory Sensing Systems and modeling information diffusion in online networks. Litou has been involved as a research assistant and software engineer in multiple research projects since 2011, contributing to areas such as real-time information propagation and emergency dissemination through online social platforms. She has co-authored various peer-reviewed publications in international conferences and journals.

Preface

This Reprint presents the third edition of the Special Issue on security and privacy in blockchains and the Internet of Things (IoT), bringing together a mature and thematically refined collection of scientific contributions that reflect the evolving challenges of modern data ecosystems. Over recent years, blockchain technologies and the IoT have moved beyond experimental deployments and are now increasingly embedded in real-world infrastructures that support data-driven and autonomous processes. As a result, questions of security, privacy, and trust have gained renewed urgency and complexity, motivating sustained and interdisciplinary research efforts.

The primary aim of this Reprint is to provide a coherent and in-depth perspective on how secure and privacy-preserving blockchain and IoT systems can be designed, analyzed, and deployed in practice. While early research in this area often focused on isolated mechanisms or narrow application scenarios, the contributions collected in this Reprint emphasize holistic approaches that span multiple system layers and that account for the interaction between decentralized infrastructures, data-intensive processing, and autonomous decision making. This perspective is essential in environments where failures or attacks may have cascading effects and where trust in both technology and data is a prerequisite for widespread adoption.

The third edition builds upon the foundations established by the two preceding editions of the Special Issue, while deliberately shifting its focus to address emerging research priorities. The first edition primarily concentrated on privacy-preserving mechanisms, lightweight security solutions, and legal considerations in early blockchain and IoT scenarios. The second edition broadened this view by addressing end-to-end data flows and fundamental security properties such as confidentiality, integrity, and authenticity across heterogeneous systems. In contrast, the third edition places explicit emphasis on trust as a central and integrative concept. It explores how trust can be established, assessed, and maintained in decentralized and intelligent systems, and how it interrelates with security mechanisms, privacy-preserving designs, and data-centric processes.

The scope of this Reprint reflects this refined focus. It includes foundational analyses of threats, vulnerabilities, and privacy risks in blockchain and IoT ecosystems, offering structured taxonomies and systematic perspectives on the evolving attack surface. The included contributions provide essential conceptual grounding and identify persistent research challenges that continue to shape the field. Complementing these analyses, several contributions propose concrete mechanisms and architectures for secure authentication, access control, key management, and data protection, with particular attention to heterogeneous and resource-constrained environments.

A distinctive feature of this third edition is its strong emphasis on intelligent and data-driven security mechanisms. Advances in machine learning, graph-based analytics, explainable artificial intelligence, and federated learning are increasingly leveraged to detect anomalies, assess trustworthiness, and enhance resilience against sophisticated attacks. The included contributions demonstrate how such techniques can be effectively integrated into blockchain and IoT systems, while also critically discussing their limitations, scalability implications, and the need for transparency and explainability. This focus reflects the growing importance of adaptive and learning-based approaches in addressing complex and dynamic threat models.

Trust is treated throughout this Reprint as a multidimensional property that extends beyond technical correctness. It encompasses trust in protocols and infrastructures, trust in the integrity and authenticity of data, and trust in the outcomes of autonomous or semi-autonomous decisions. Several contributions illustrate how trust can be supported through mechanisms such as auditability,

accountability, non-repudiation, and verifiable data lifecycle management, including secure storage and deletion. These aspects are particularly relevant in safety-critical and regulated domains, where technological assurances must align with organizational, legal, and societal expectations.

In addition to methodological contributions, this Reprint places strong emphasis on practical relevance and real-world applicability. System-oriented studies and applied frameworks demonstrate how blockchain-based security solutions can be deployed in domains such as healthcare, industrial systems, digital forensics, smart grids, and enterprise environments. Experimental evaluations, performance analyses, and realistic deployment scenarios enable readers to assess trade-offs and feasibility, bridging the gap between conceptual designs and operational systems. This practical orientation underscores the increasing maturity of the field and its readiness to address real-world challenges.

The motivation for assembling this Reprint is to offer readers both a snapshot of the current state-of-the-art and a forward-looking view on open research directions. Challenges related to scalability, cross-organizational collaboration, human-centric vulnerabilities, quantum-aware security, and the long-term sustainability of trust mechanisms remain active areas of investigation. By consolidating diverse perspectives and approaches, this Reprint seeks to stimulate further research and to encourage the development of resilient, privacy-preserving, and trustworthy distributed systems.

This Reprint is addressed to researchers, practitioners, and advanced students working in the areas of security, privacy, distributed systems, and data-intensive applications. It is intended to serve as both a reference for those entering the field and a source of insight for experts seeking to extend existing approaches or to apply them in new contexts.

As Guest Editors, we hope that this compilation not only documents the progress achieved so far, but also contributes to shaping future research and collaboration at the intersection of blockchain technologies and the IoT.

Christoph Stach, Clémentine Gritti, and Iouliana Litou
Guest Editors

future internet

MDPI

Editorial

Special Issue on Security and Privacy in Blockchains and the IoT—3rd Edition

Christoph Stach [1,*], Clémentine Gritti [2,*] and Iouliana Litou [3,*]

[1] Institute for Parallel and Distributed Systems, University of Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany
[2] CITI Lab, INSA Lyon, Building Hedy Lamarr, 6 Avenue des Art, 69100 Villeurbanne, France
[3] Department of Informatics, Athens University of Economics and Business, Patision 76, 104 34 Athens, Greece
* Correspondence: christoph.stach@ipvs.uni-stuttgart.de (C.S.); clementine.gritti@insa-lyon.fr (C.G.); litou@aueb.gr (I.L.); Tel.: +49-711-68588-433 (C.S.)

The rapid digital transformation of modern society is increasingly driven by the convergence of blockchain technologies and the Internet of Things (IoT), both of which have become foundational pillars of contemporary distributed systems. Blockchain introduces decentralized trust, immutability, and transparency, while IoT enables pervasive sensing, automation, and data-driven decision-making across domains such as healthcare, industry, energy, transportation, and smart cities. Collectively, these technologies promise to reshape digital infrastructures by eliminating single points of failure, improving operational efficiency, and enabling new forms of collaboration among heterogeneous and autonomous entities. However, this convergence also substantially enlarges the attack surface and introduces complex security and privacy challenges that cannot be adequately addressed by traditional centralized protection mechanisms.

IoT environments are inherently heterogeneous, resource-constrained, and often deployed in safety-critical contexts, making them particularly vulnerable to cyber attacks, data manipulation, and unauthorized access. At the same time, blockchain systems, despite their cryptographic foundations, face a wide range of threats including protocol-level attacks, smart contract vulnerabilities, privacy leakage, and emerging risks associated with scalability and quantum computing. The interaction between these two paradigms further amplifies such challenges, as weaknesses in one layer can cascade across the entire system. Ensuring confidentiality, integrity, availability, and accountability in these interconnected ecosystems therefore remains a fundamental research problem.

Beyond purely technical considerations, trust has emerged as a central concept in blockchain- and IoT-based systems. Trust is required not only in the underlying protocols and infrastructures, but also in the data they collect, process, and share, as well as in the autonomous decisions derived from that data. Privacy concerns are equally critical, particularly when sensitive personal, industrial, or societal data are involved and when regulatory requirements such as data protection and accountability must be satisfied. Consequently, security and privacy can no longer be treated as isolated design goals, but must be addressed holistically and throughout the entire system lifecycle.

Against this backdrop, sustained research efforts are required to develop robust security mechanisms, privacy-preserving architectures, and trust-building strategies that are both theoretically sound and practically viable. This Special Issue is motivated by the need to bring together advances from academia and industry that address these challenges from complementary perspectives. By fostering interdisciplinary approaches and bridging foundational research with real-world applications, this Special Issue aims to contribute to the development of resilient, trustworthy, and future-proof blockchain and IoT ecosystems.

Building upon the broad relevance of security, privacy, and trust in blockchain- and IoT-enabled systems, this Special Issue focuses on advancing the understanding and practical realization of these concepts in decentralized and data-intensive environments. The scope of this Special Issue reflects the increasing maturity of blockchain and IoT technologies, as well as their growing adoption in real-world, security-critical domains. Rather than treating both paradigms in isolation, this Special Issue emphasizes their combined deployment and the resulting security, privacy, and trust implications across different system layers.

The contributions collected in this volume address security and privacy challenges ranging from foundational analyses to concrete architectural solutions and applied systems. Several papers investigate vulnerabilities, attack surfaces, and mitigation strategies in blockchain and IoT ecosystems, providing structured taxonomies and systematic perspectives on current threats. Others focus on mechanisms that support secure authentication, access control, and key management as integral components of distributed architectures, particularly in heterogeneous and resource-constrained environments.

A central theme of this Special Issue is the protection of data throughout its lifecycle, including privacy-preserving data collection, secure storage, verifiable deletion, and trustworthy sharing across organizational boundaries. In this context, blockchain-based mechanisms for ensuring data integrity, auditability, and non-repudiation play a prominent role. The design and analysis of smart contracts, consensus mechanisms, and off-chain storage strategies are examined with respect to both security guarantees and operational efficiency.

Threat detection and resilience constitute another major focus of this Special Issue. Several contributions explore advanced detection mechanisms based on anomaly detection, graph-based learning, explainable artificial intelligence, and federated learning. These approaches address both technical attacks on infrastructures and human-centric vulnerabilities, highlighting the multifaceted nature of security and trust in decentralized systems. Scalability and performance considerations are explicitly taken into account, reflecting the requirements of large-scale IoT deployments, industrial systems, and next-generation network environments.

In addition to technical mechanisms, this Special Issue includes applied studies and system-oriented frameworks that demonstrate how blockchain-based security solutions can be integrated into domains such as healthcare, digital forensics, smart grids, and enterprise networks. These works implicitly address regulatory, legal, and compliance-related concerns by supporting transparency, traceability, and evidentiary integrity in sensitive application contexts. Through this carefully aligned scope and set of topics, this Special Issue presents a coherent collection of contributions that collectively advance the state of the art while remaining grounded in practical and deployable security and privacy solutions.

The inaugural two volumes of this Special Issue (see Stach [1] and Stach and Gritti [2]) established a solid foundation by addressing core security and privacy challenges in blockchain- and IoT-based systems, with a strong emphasis on data protection, secure data management, and compliance with regulatory requirements. While the first volume primarily focused on privacy-preserving mechanisms, lightweight authentication, and legal considerations in emerging IoT scenarios, the second volume broadened this perspective by offering a holistic view on end-to-end data flows and fundamental security properties such as authenticity, confidentiality, and integrity. Building on these contributions, this third volume shifts the focus toward the increasingly critical notion of trust and its interplay with security and privacy in decentralized and intelligent systems. In contrast to earlier volumes, this edition places greater emphasis on advanced threat models, trust-aware architectures, and data-driven detection mechanisms that reflect the growing complexity of modern blockchain and IoT deployments. Furthermore, emerging technologies such as artificial intelligence, federated learning, quantum-aware security concepts, and large-scale

https://doi.org/10.3390/fi18020090

industrial infrastructures receive increased attention. The current volume also strengthens the link between foundational research and real-world deployment by presenting system-oriented solutions and applied studies in domains such as healthcare, digital forensics, smart grids, and enterprise networks. In this way, this third edition complements and extends the previous volumes by addressing both the maturity of the technologies and the evolving security, privacy, and trust requirements of next-generation distributed systems.

This volume of the Special Issue comprises twelve high-quality contributions that address the overarching theme of security and privacy in blockchain and IoT systems from a wide range of methodological and application-oriented perspectives. Collectively, these papers reflect the diversity and maturity of current research, spanning foundational analyses, advanced security mechanisms, and domain-specific system implementations. For clarity and coherence, the contributions are organized into three thematic blocks, which are outlined in the following paragraphs.

Foundations, Threats, and Privacy Challenges in Blockchain and IoT Ecosystems.

The first thematic block lays the conceptual and analytical foundations of this Special Issue by examining fundamental security, privacy, and trust challenges in blockchain and IoT ecosystems. It brings together comprehensive reviews and surveys that systematically analyze vulnerabilities, attack surfaces, and emerging trends across decentralized systems. By structuring the problem space and identifying persistent research challenges, this block establishes a shared understanding that informs the more solution-oriented contributions that follow.

The paper by Siam et al. [3] provides a comprehensive systematic review of blockchain vulnerabilities, attacks, and mitigation strategies, offering a broad and structured entry point into the security landscape of decentralized ecosystems. Following PRISMA guidelines, the study synthesizes the recent literature to categorize attacks across multiple domains, including network, consensus, smart contracts, transactions, and governance. The authors not only identify dominant threat vectors such as smart contract vulnerabilities and consensus manipulation, but also analyze their underlying causes and real-world implications. By mapping attacks to corresponding countermeasures, the paper highlights gaps between theoretical defenses and practical deployment. Special attention is given to context-specific mitigation strategies for domains such as finance, healthcare, and IoT. The discussion of trade-offs between decentralization, scalability, and security further emphasizes the complexity of secure blockchain design. As such, this work provides a structured baseline for understanding the evolving threat landscape addressed by subsequent contributions.

Complementing this broad perspective, Islam et al. [4] present a layer-oriented survey that decomposes blockchain systems into data, network, consensus, contract, and application layers. This architectural viewpoint enables a more fine-grained analysis of how vulnerabilities manifest at different system levels and how attacks propagate across layers. The paper introduces a detailed taxonomy of detection and mitigation approaches, including formal verification, static and dynamic analysis, and machine learning-based techniques. By systematically linking specific threats to detection methods, the authors provide practical guidance for security tool selection and system hardening. The survey also highlights emerging challenges such as adversarial machine learning and the lack of labeled datasets for intelligent detection. The inclusion of post-quantum threats and explainability considerations extends the relevance of the analysis beyond current-generation systems. This layered perspective deepens the foundational understanding introduced in the preceding paper.

https://doi.org/10.3390/fi18020090

Shifting the focus toward the convergence of blockchain and IoT, Kareem et al. [5] survey emerging blockchain platforms that are explicitly adopted to enhance IoT security. The paper analyzes and compares platforms such as Ethereum, Hyperledger Fabric, IOTA, IoTeX, Algorand, and Multichain with respect to their suitability for resource-constrained environments. By examining consensus mechanisms, cryptographic primitives, and architectural design choices, the study clarifies how blockchain components contribute to IoT security requirements. The authors emphasize the role of permissioned architectures, interoperability, and lightweight cryptography in achieving practical deployments. Scalability limitations and performance bottlenecks are discussed alongside potential optimization strategies. The survey serves as a bridge between abstract security challenges and concrete platform-level design decisions. In doing so, it prepares the study area for the more mechanism-focused and applied contributions in later blocks.

Concluding the first thematic block, Bayan et al. [6] investigate recent trends and privacy concerns in permissionless blockchains within the rapidly evolving Web3 ecosystem. The paper identifies how developments such as high-throughput networks and real-world asset tokenization reshape the security and privacy threat landscape. A systematic analysis of privacy threats, including deanonymization, MEV exploitation, and key compromise, highlights the persistent tension between transparency and confidentiality. The authors critically assess privacy-enhancing technologies such as zero-knowledge proofs, ring signatures, and stealth addresses, discussing their trade-offs in scalability and usability. A notable contribution is the introduction of a secure development lifecycle tailored to decentralized applications, emphasizing security-by-design principles. By grounding the discussion in recent real-world incidents, the paper connects abstract privacy risks to concrete development practices. This forward-looking perspective provides a natural transition from foundational analysis to the design of effective countermeasures.

Threat Detection, Trust, and Intelligent Security Mechanisms.

The second thematic block focuses on concrete mechanisms for detecting threats, establishing trust, and enhancing resilience in blockchain and IoT systems. A unifying theme of this block is the use of data-driven and intelligent techniques to cope with increasingly sophisticated attack scenarios. Together, the contributions demonstrate how advanced analytics, machine learning, and trust modeling can complement traditional security mechanisms.

Pérez-Cano and Jurado [7] investigate fraud detection in cryptocurrency networks through a comparative analysis of unsupervised anomaly detection and supervised graph-based learning. Using large-scale real-world Bitcoin datasets, the study evaluates the effectiveness of classical anomaly detection algorithms and highlights their limitations when structural information is ignored. The introduction of heterogeneous graph transformers enables the exploitation of relational dependencies between transactions and wallets. The paper demonstrates that incorporating graph structure significantly improves fraud detection performance under severe class imbalance. By comparing supervised and unsupervised approaches within a unified framework, the study provides valuable insights into their respective strengths and limitations. The findings underscore the importance of combining intrinsic transaction features with network-level information. This work establishes a strong analytical foundation for intelligent detection mechanisms in decentralized financial systems.

Extending the detection perspective to IoT environments, Lightbody et al. [8] introduce Dragon_Pi, a novel dataset based on side-channel power consumption signals from IoT devices. The paper proposes an unsupervised convolutional autoencoder to detect anomalies without relying on labeled attack data. By focusing on physical power traces

https://doi.org/10.3390/fi18020090

rather than network traffic, the approach captures post-exploitation behavior that is often invisible to traditional monitoring tools. The cross-platform design demonstrates robustness across heterogeneous hardware architectures. The experimental results show that reconstruction error-based detection achieves a strong performance, particularly when combined with signal post-processing. The public release of the dataset addresses a critical gap in IoT security research. This contribution highlights the value of alternative data modalities for resilient intrusion detection.

Addressing the human dimension of security, Fan et al. [9] analyze phishing susceptibility using explainable artificial intelligence. The paper proposes a deep learning model augmented with SHAP-based explanations to identify individual risk factors influencing phishing behavior. By integrating behavioral habits, psychological traits, and demographic information, the study moves beyond population-level analysis toward personalized risk assessment. The use of explainable AI enables the transparent interpretation of model predictions, making the results actionable for security awareness and training. The findings reveal that behavioral factors often outweigh demographic characteristics in determining susceptibility. This human-centric perspective complements system-level detection approaches by addressing social engineering threats. The work demonstrates how trust and security depend not only on technical mechanisms but also on user behavior.

Concluding this block, Alharbi [10] presents a forward-looking framework that integrates blockchain, federated learning, and quantum-inspired trust modeling for industrial IoT in 6G environments. The proposed architecture introduces a quantum trust score to dynamically assess device reliability under Byzantine and poisoning attacks. Blockchain-based sharding ensures the scalable and immutable recording of trust updates. The experimental results demonstrate strong resilience and a high detection accuracy under adversarial conditions. The integration of explainability mechanisms further supports operational transparency. By addressing both current and future threat models, including quantum-era challenges, this work represents a culmination of the intelligent and trust-aware approaches discussed in this block. It also forms a conceptual bridge toward applied, large-scale system deployments.

Secure Blockchain-Based Architectures and Domain-Specific Applications.

The third thematic block emphasizes system-oriented solutions and real-world applications that demonstrate the practical maturity of blockchain-based security mechanisms. The common thread of this block is the translation of security, privacy, and trust concepts into deployable architectures across diverse domains. These contributions illustrate how theoretical principles can be operationalized under real-world constraints.

Aref and Ouda [11] introduce HSM4SSL, a hardware-backed architecture for securing intra-domain communications using hardware security modules. The proposed system unifies certificate management and cryptographic operations across multiple protocols through a modular SaaS design. Performance evaluations demonstrate improvements over conventional software-based solutions in terms of throughput and latency. The abstraction of vendor-specific HSM interfaces enhances flexibility and scalability. Automated certificate lifecycle management reduces operational complexity and human error. By focusing on enterprise-grade deployment, the paper addresses practical security challenges often overlooked in academic research. This contribution establishes a strong architectural foundation for secure system integration.

Focusing on healthcare emergency response systems, Marletta et al. [12] propose a blockchain-based framework for certifying timestamps and ensuring data integrity. By storing cryptographic hashes of critical events on an EVM-compatible blockchain, the system provides immutable and verifiable proof of data authenticity. Off-chain storage strategies

https://doi.org/10.3390/fi18020090

ensure scalability and privacy preservation, and experimental evaluations using real-world datasets demonstrate economic feasibility and reliable transaction confirmation times. The framework supports independent verification in dispute scenarios, strengthening trust among stakeholders. This work exemplifies how blockchain can replace centralized trusted third parties in high-stakes environments, highlighting the role of trust and non-repudiation in safety-critical applications.

In the context of digital forensics, Huang et al. [13] present a blockchain-based framework for OSINT evidence collection and identification. The proposed lifecycle model aligns technical processes with legal standards for evidence admissibility. On-acquisition notarization using a permissioned judicial blockchain ensures integrity and chain of custody. The integration of identity correlation mechanisms addresses attribution challenges in online investigations. AI-assisted tools further enhance evidence quality and reliability. By bridging forensic rigor with decentralized technologies, the framework strengthens trust in digital evidence. This contribution underscores the legal and societal relevance of blockchain-based security solutions.

Concluding this Special Issue, Zhang et al. [14] introduce MCS-VD, an alliance chain-driven multi-cloud storage and verifiable deletion scheme for smart grid data. The framework combines efficient consensus protocols with encrypted multi-cloud storage to address scalability and single-point-of-failure concerns. A dedicated deletion mechanism enables verifiable and permanent data removal, supporting regulatory compliance. The experimental results demonstrate significant performance gains over traditional Byzantine fault-tolerant protocols. The hierarchical design supports large-scale deployment in industrial environments. By addressing the full data lifecycle from collection to deletion, this work provides a comprehensive solution for critical infrastructure security. It offers a strong concluding perspective on the practical realization of secure, privacy-preserving, and trustworthy blockchain–IoT systems.

The twelve papers in this Special Issue offer a comprehensive and multifaceted view of current advances in security, privacy, and trust for blockchain and IoT ecosystems. By combining foundational analyses, intelligent detection mechanisms, and domain-specific system implementations, this Special Issue offers readers a coherent perspective on both the theoretical and practical dimensions of the field. The included survey and review articles establish a structured understanding of threat landscapes, architectural vulnerabilities, and emerging privacy challenges, serving as valuable reference points for researchers entering or extending work in this domain. At the same time, the methodological contributions demonstrate how advanced techniques such as machine learning, graph-based analytics, explainable artificial intelligence, and federated learning can be effectively leveraged to address real-world security threats.

A distinguishing feature of this Special Issue is its balanced integration of technical rigor and practical relevance. Several contributions move beyond conceptual proposals by providing experimentally validated systems, publicly available datasets, and performance evaluations under realistic conditions. This enables practitioners to assess the feasibility and trade-offs of proposed solutions, while offering researchers reproducible foundations for future studies. The strong emphasis on trust, accountability, and data integrity reflects the growing importance of these concepts in decentralized and autonomous systems, particularly in safety-critical and regulated environments.

Furthermore, the diversity of application domains, including healthcare, digital forensics, smart grids, industrial IoT, and enterprise networks, illustrates the broad applicability of blockchain-based security and privacy mechanisms. Readers benefit from insights into how common principles can be adapted to domain-specific constraints and requirements. The Special Issue also highlights emerging trends such as quantum-aware security, side-

https://doi.org/10.3390/fi18020090

channel-based intrusion detection, and privacy-aware development lifecycles, signaling important directions for future research and innovation.

By bringing together contributions from both academia and industry, this Special Issue fosters cross-disciplinary exchange and knowledge transfer. It provides a curated snapshot of the state of the art while simultaneously outlining practical pathways toward more resilient and trustworthy digital infrastructures. As such, this Special Issue is intended not only as a collection of individual research outcomes, but as a cohesive resource that informs, inspires, and supports the ongoing evolution of secure and privacy-preserving blockchain and IoT systems.

Despite the significant progress reflected in this Special Issue, securing blockchain and IoT ecosystems remains an ongoing challenge as these technologies continue to evolve and scale. Open research directions include achieving stronger security and privacy guarantees without compromising scalability, addressing emerging threats driven by artificial intelligence and quantum computing, and developing standardized benchmarks for evaluating trust-aware systems. Furthermore, the integration of human-centric factors, regulatory requirements, and explainable security mechanisms will become increasingly important for real-world adoption. Continued collaboration between academia, industry, and policymakers will therefore be essential to ensure that future decentralized systems are not only technically robust, but also trustworthy and socially acceptable.

As the Guest Editors, we would like to sincerely thank all authors for submitting their high-quality and insightful manuscripts to our Special Issue. Their contributions form the intellectual foundation of this volume and reflect the diversity, rigor, and relevance of current research in this rapidly evolving field. We are also deeply grateful to the reviewers for their careful, constructive, and timely evaluations, which were instrumental to enhancing the scientific quality and clarity of the accepted papers. Their expertise and dedication have been essential to maintaining the high standards of this Special Issue. Finally, we would like to express our sincere appreciation to the MDPI editorial team for their continuous support and efficient coordination throughout the entire editorial process, without which the successful completion of this Special Issue would not have been possible.

Author Contributions: All authors have read and agreed to the published version of the manuscript.

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

References

1. Stach, C. (Ed.) *Security and Privacy in Blockchains and the IoT*; MDPI: Basel, Switzerland, 2023. [CrossRef]
2. Stach, C.; Gritti, C. (Eds.) *Security and Privacy in Blockchains and the IoT II*; MDPI: Basel, Switzerland, 2023. [CrossRef]
3. Siam, M.K.; Saha, B.; Hasan, M.M.; Hossain Faruk, M.J.; Anjum, N.; Tahora, S.; Siddika, A.; Shahriar, H. Securing Decentralized Ecosystems: A Comprehensive Systematic Review of Blockchain Vulnerabilities, Attacks, and Countermeasures and Mitigation Strategies. *Future Internet* **2025**, *17*, 183. [CrossRef]
4. Islam, M.J.; Islam, S.; Hossain, M.; Noor, S.; Islam, S.M.R. Securing Blockchain Systems: A Layer-Oriented Survey of Threats, Vulnerability Taxonomy, and Detection Methods. *Future Internet* **2025**, *17*, 205. [CrossRef]
5. Kareem, Y.; Djenouri, D.; Ghadafi, E. A Survey on Emerging Blockchain Technology Platforms for Securing the Internet of Things. *Future Internet* **2024**, *16*, 285. [CrossRef]
6. Bayan, T.; Yazici, A.; Banach, R. Permissionless Blockchain Recent Trends, Privacy Concerns, Potential Solutions and Secure Development Lifecycle. *Future Internet* **2025**, *17*, 547. [CrossRef]
7. Pérez-Cano, V.; Jurado, F. Fraud Detection in Cryptocurrency Networks—An Exploration Using Anomaly Detection and Heterogeneous Graph Transformers. *Future Internet* **2025**, *17*, 44. [CrossRef]
8. Lightbody, D.; Ngo, D.M.; Temko, A.; Murphy, C.C.; Popovici, E. Dragon_Pi: IoT Side-Channel Power Data Intrusion Detection Dataset and Unsupervised Convolutional Autoencoder for Intrusion Detection. *Future Internet* **2024**, *16*, 88. [CrossRef]
9. Fan, Z.; Li, W.; Laskey, K.B.; Chang, K.C. Investigation of Phishing Susceptibility with Explainable Artificial Intelligence. *Future Internet* **2024**, *16*, 31. [CrossRef]

https://doi.org/10.3390/fi18020090

10. Alharbi, S. QuantumTrust-FedChain: A Blockchain-Aware Quantum-Tuned Federated Learning System for Cyber-Resilient Industrial IoT in 6G. *Future Internet* **2025**, *17*, 493. [CrossRef]
11. Aref, Y.; Ouda, A. HSM4SSL: Leveraging HSMs for Enhanced Intra-Domain Security. *Future Internet* **2024**, *16*, 148. [CrossRef]
12. Marletta, D.; Midolo, A.; Tramontana, E. A Blockchain-Based Strategy for Certifying Timestamps in a Distributed Healthcare Emergency Response Systems. *Future Internet* **2025**, *17*, 210. [CrossRef]
13. Huang, H.W.; Shih, C.H.; Li, C.Y.; Teng, H.Y. A Blockchain-Based Framework for OSINT Evidence Collection and Identification. *Future Internet* **2025**, *17*, 551. [CrossRef]
14. Zhang, L.; Luo, J.; Yang, Y.; Wang, W. MCS-VD: Alliance Chain-Driven Multi-Cloud Storage and Verifiable Deletion Scheme for Smart Grid Data. *Future Internet* **2026**, *18*, 56. [CrossRef]

Short Biography of Authors



Dr. Christoph Stach holds the Substitute Chair of Data Engineering and heads the Information Systems and Applications working area at the Institute for Parallel and Distributed Systems at the University of Stuttgart. He received his doctorate in 2017, specializing in data security and privacy for mobile applications. His current research focuses on demand-oriented data management at the intersection of big data, data refinement, and data protection. An accomplished researcher with an h-index of 20, Dr. Stach has authored over 70 peer-reviewed publications on this topic. The practical relevance and economic impact of his research are further demonstrated by his leadership in successfully executing five large-scale industrial projects. His extensive service to the academic community includes participating in more than 50 international conference program committees. He serves as Guest Editor for this Special Issue to explore and leverage the synergies between decentralized data management and the security challenges inherent in modern IoT environments.



Clémentine Gritti received the M.Sc. degree in computer science from Grenoble Alpes University, France, in 2012, and the Ph.D. degree in computer science from the University of Wollongong, Australia, in 2017. She was a Senior Lecturer at the University of Canterbury, New Zealand, between 2020 and 2023. Thereafter, she was a research fellow at Eurecom, France. In 2024, she joined INSA Lyon, France, as a Professor Junior. Her current research interests include the design and evaluation of public-key cryptographic protocols for applied security and privacy in various contexts, such as Federated Learning, Internet of Things, and blockchain technologies.



Iouliana Litou is a computer scientist and researcher primarily focused on social networks, information dissemination, influence maximization, and misinformation analysis. She has pursued her academic career at the Athens University of Economics and Business (AUEB), Department of Informatics, where she completed her B.Sc. and M.Sc. degrees in Computer Science and later completed her Ph.D. in Computer Science under the supervision of Prof. Vana Kalogeraki. Her Ph.D. research centers on Participatory Sensing Systems and modeling information diffusion in online networks. Litou has been involved as a research assistant and software engineer in multiple research projects since 2011, contributing to areas such as real-time information propagation and emergency dissemination through online social platforms. She has co-authored various peer-reviewed publications in international conferences and journals.

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

https://doi.org/10.3390/fi18020090

future internet

MDPI

Systematic Review

Securing Decentralized Ecosystems: A Comprehensive Systematic Review of Blockchain Vulnerabilities, Attacks, and Countermeasures and Mitigation Strategies

Md Kamrul Siam [1], Bilash Saha [2], Md Mehedi Hasan [2], Md Jobair Hossain Faruk [1,*], Nafisa Anjum [2], Sharaban Tahora [2], Aiasha Siddika [2] and Hossain Shahriar [3,*]

[1] Department of Computer Science, New York Institute of Technology, New York, NY 10023, USA; ksiam01@nyit.edu
[2] Department of Information Technology, Kennesaw State University, Kennesaw, GA 30144, USA
[3] Center for Cybersecurity, University of West Florida, Pensacola, FL 32514, USA
* Correspondence: mhossa37@nyit.edu (M.J.H.F.); hshahria2012@gmail.com (H.S.)

Abstract: Blockchain technology has emerged as a transformative innovation, providing a transparent, immutable, and decentralized platform that underpins critical applications across industries such as cryptocurrencies, supply chain management, healthcare, and finance. Despite their promise of enhanced security and trust, the increasing sophistication of cyberattacks has exposed vulnerabilities within blockchain ecosystems, posing severe threats to their integrity, reliability, and adoption. This study presents a comprehensive and systematic review of blockchain vulnerabilities by categorizing and analyzing potential threats, including network-level attacks, consensus-based exploits, smart contract vulnerabilities, and user-centric risks. Furthermore, the research evaluates existing countermeasures and mitigation strategies by examining their effectiveness, scalability, and adaptability to diverse blockchain architectures and use cases. The study highlights the critical need for context-aware security solutions that address the unique requirements of various blockchain applications and proposes a framework for advancing proactive and resilient security designs. By bridging gaps in the existing literature, this research offers valuable insights for academics, industry practitioners, and policymakers, contributing to the ongoing development of robust and secure decentralized ecosystems.

Keywords: blockchain technology; blockchain security; blockchain attacks; cybercrime; decentralization; data privacy

1. Introduction

Blockchain technology can potentially revolutionize various industries, including finance, healthcare, and supply chain management [1,2]. Its decentralized nature offers several advantages over traditional centralized systems, such as increased transparency, immutability, and security. These features have fueled significant adoption of and investment in blockchain technologies. According to Deloitte's 2021 Global Blockchain Survey, 76% of respondents reported that their organizations had adopted blockchain as a digital asset, while 83% believed digital assets would replace fiat currency within the next decade [3]. The growth trajectory of blockchain technology is also evident from projections by Grand View Research Inc., which estimated that the global blockchain technology market would reach USD 1431.54 billion by 2030, with a compound annual growth rate (CAGR) of 87.7%

between 2023 and 2030 [4]. Similarly, another report predicted that the blockchain distributed ledger market would grow from USD 2.89 billion in 2020 to USD 137.29 billion by 2027, with a CAGR of 62.7% [5].

While the financial services sector currently dominates the blockchain market, the technology's applications have rapidly expanded into various domains. Beyond cryptocurrencies, blockchain enables cross-border money transfers [6], supports Ethereum Virtual Machine (EVM)-based smart contracts [7], integrates into the Internet of Things (IoT) ecosystem [8,9], enhances data and identity security [10,11], and facilitates electronic healthcare records (EHRs) [12–14]. Additionally, blockchain has been applied to automated logistics systems [15] and the burgeoning market of non-fungible tokens (NFTs) [16].

Despite its transformative potential, blockchain technology faces significant challenges, particularly in security. The decentralized nature of blockchain networks does not render them immune to attacks; on the contrary, it introduces novel vulnerabilities. Blockchain system hacks have increased, and NIST reported a notable rise in vulnerabilities in their published report in 2023 [17], while SonicWall highlighted a sharp rise in cybercrimes in its 2024 Cyber Threat Report [18]. Real-world incidents illustrate the severity of these threats. For example, Ethereum Classic suffered multiple 51% and double-spending attacks in 2019 and 2020, resulting in a loss of over USD 6.7 million [19,20]. On 10 August 2021, the Poly Network was exploited via a smart contract vulnerability, leading to the theft of USD 611 million—the largest crypto-related hack to date [21]. Similarly, Binance, the world's largest cryptocurrency exchange, faced a major breach in October 2022, with hackers stealing 2 million BNB tokens worth USD 570 million [22,23]. These incidents underscore the urgency of addressing security challenges within blockchain ecosystems.

As blockchain adoption accelerates, understanding its vulnerabilities and implementing robust countermeasures is critical. The current study aims to contribute to this effort by conducting a comprehensive analysis of blockchain attacks, vulnerabilities, and mitigation strategies. A thorough examination of existing security challenges and their impacts can help practitioners and researchers design more resilient blockchain systems. To guide this research, we formulated the following research questions:

- RQ1: What are the most common types of attacks on blockchain technology, and how do they affect the security and integrity of the system?
- RQ2: What security measures and technologies have been employed to detect and mitigate malicious blockchain attacks?
- RQ3: How can context-specific mitigation strategies be designed to address the unique requirements and constraints of different blockchain applications?

To address these questions, this study adopts a qualitative and quantitative approach to reviewing the state-of-the-art research on blockchain security and privacy. By presenting a holistic outline of vulnerabilities, attack vectors, and countermeasures, this research serves as a valuable resource for academics, industry professionals, and policymakers seeking to safeguard decentralized ecosystems from current and future threats.

2. Methodology

We employed a systematic literature search strategy, where we used three major online digital libraries: IEEE Xplore, ACM digital library, and Springer Nature Link digital library. An analytical process was undertaken to synthesize the information for the research questions. Out of the initial 1143 studies, 73 studies were from IEEE Xplore, 578 studies were from ACM digital library, and 492 studies were from SpringerLink library.

2.1. Scoping Criteria

To address the research questions concerning the common types of attacks on blockchain technology, their impact on security and integrity, and the security measures employed to detect and mitigate such attacks, we established comprehensive scoping criteria. These criteria focus on identifying, categorizing, and evaluating blockchain vulnerabilities, attack mechanisms, and mitigation strategies. In order to limit our scope, we focused on the top three cryptocurrencies (Bitcoin, Ethereum, and Tether) based on their market capital according to Forbes [24]. In order to guarantee that this systematic evaluation covers the most recent blockchain security risks, new attack vectors, and cutting-edge mitigation techniques, publications from 2020 to 2024 were chosen. With developments in decentralized finance (DeFi), smart contracts, Layer 2 scaling solutions, and cross-chain protocols, the blockchain environment has swiftly changed, bringing with it new vulnerabilities including bridge attacks, reentrancy problems, and flash loan exploits [25]. Furthermore, modern cryptographic countermeasures and security frameworks are required due to recent advancements in quantum computing and AI-driven cyberthreats [26]. Additionally, security compliance and governance models have been impacted by global regulatory changes, such as the EU's Markets in Crypto-Assets (MiCA) framework and the U.S. SEC's heightened scrutiny [27].

Our analysis explores key attack vectors, including 51% attacks, bribing attacks, double-spend attacks, smart contract vulnerability attacks, selfish mining attacks, and Sybil attacks [28,29]. For example, a 51% attack allows malicious actors to gain control of the majority of a network's computational power, enabling them to manipulate transactions and even reverse previously confirmed transactions. A notable instance occurred in the Ethereum Classic blockchain in 2019, resulting in significant financial losses. Similarly, the study examines vulnerabilities within smart contracts, such as the infamous Poly Network attack in 2021, where a critical exploit led to the theft of USD 611 million.

2.2. Structured Literature Search Procedure

2.2.1. Eligibility Criteria

- Inclusion: Peer-reviewed studies, conference papers, journal papers, and research articles published between 1 January 2020 and 31 December 2024.
- Exclusion: Studies lacking empirical data, opinion pieces, keynotes, short papers, magazines, books, non-English articles, retracted papers, and studies that do not discuss mitigation, detection, and prevention strategies.

2.2.2. Search Strategy

We conducted our search with specific keywords based on the given queries (last search date: 10 March 2025):

- IEEE Xplore library: ((("All Metadata": blockchain) AND ("All Metadata": security) AND ("All Metadata": privacy) AND ("All Metadata": attack) AND ("All Metadata": vulnerability*)) AND (("All Metadata": bitcoin) OR ("All Metadata": btc) OR ("All Metadata": tether) OR ("All Metadata": usdt) OR ("All Metadata": ethereum) OR ("All Metadata": eth)) AND (("All Metadata": smart contract) OR ("All Metadata": Cryptograph*) OR ("All Metadata": Cryptocurrenc*)));
- ACM digital library: [All: blockchain] AND [All: security] AND [All: privacy] AND [All: attack] AND [All: vulnerabilit*] AND [[All: bitcoin] OR [All: btc] OR [All: tether] OR [All: usdt] OR [All: ethereum] OR [All: eth]] AND [[All: "smart contract"] OR [All: cryptograph*] OR [All: cryptocurrenc*]] AND [E-Publication Date: 1 January 2020 TO 31 December 2024)];

- Nature SpringerLink library: (blockchain AND security AND privacy AND attack AND vulnerability*) AND (bitcoin OR btc OR tether OR usdt OR ethereum OR eth) AND (smart contract OR cryptograph* OR cryptocurrenc*).

2.2.3. Selection Process

We recruited six independent reviewers for the article screening process, and it was carried out using a manual quantitative analysis process achieved through open coding. All of the reviewers agreed to each of the codes that we used to answer our research questions. We also added a PRISMA flowchart to depict the study selection stages, which is shown in Figure 1.

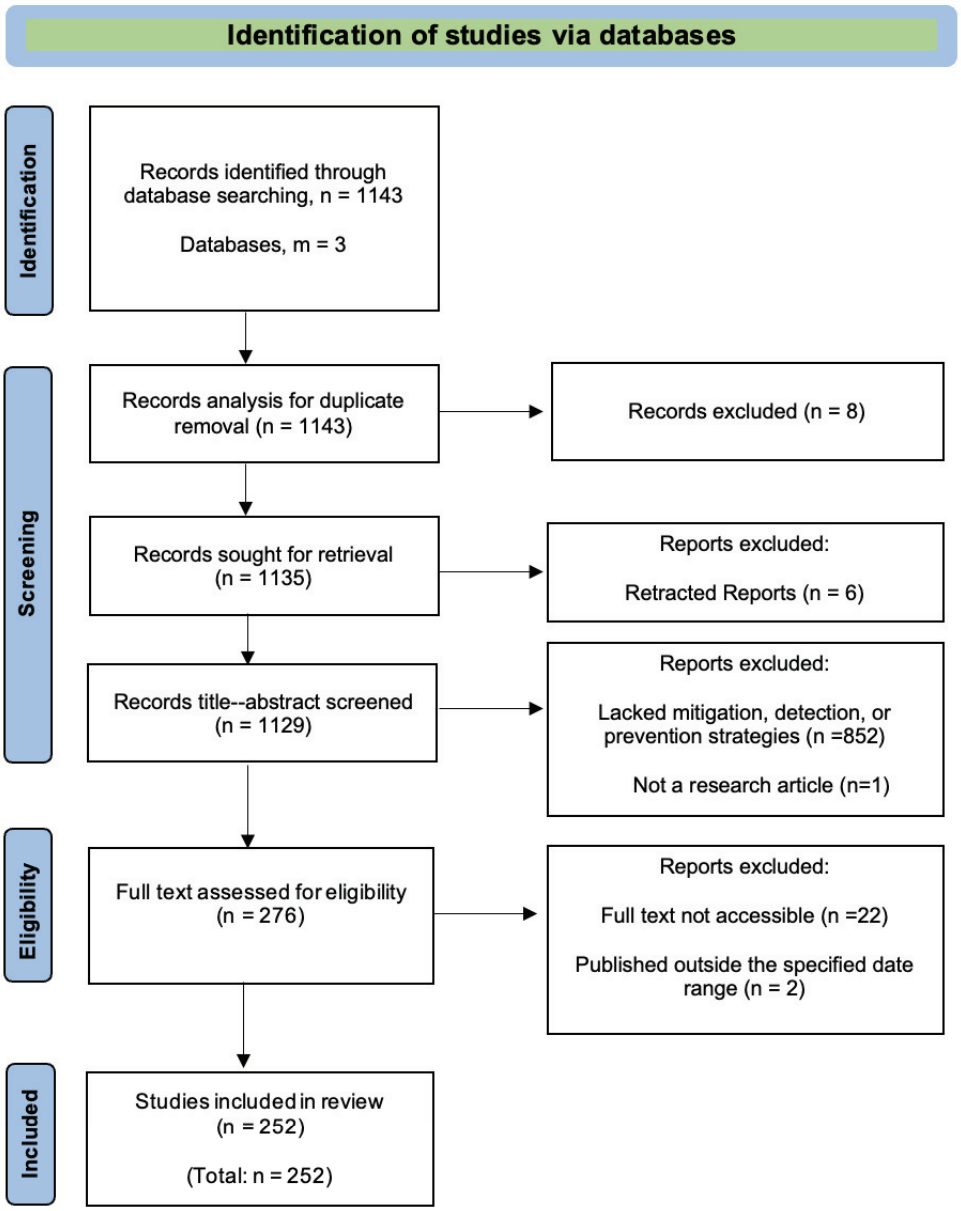


Figure 1. PRISMA flow diagram for study selection in the systematic review.

2.2.4. Data Extraction Process

The data extraction process was structured to ensure the consistency, accuracy, and relevance of the extracted information. Initially, we found 1143 studies from three distinct

databases. With the screening process that is shown in Figure 1, we ended up adding 252 studies to our research. Each selected study was reviewed to extract the key details, including attack type, impact assessment, mitigation strategies, study methodology, and blockchain application domain. The data were extracted using standardized forms and reviewed independently by six researchers to minimize bias, with any disagreements resolved through discussion or consultation with a seventh reviewer. The extracted information was cross-verified with original studies to maintain consistency, and any missing data were either supplemented using additional sources or excluded from the study. Finally, the extracted data were structured for comparison and the key findings were summarized to facilitate qualitative and quantitative analysis. This structured approach ensured that all relevant aspects of blockchain security threats and mitigation strategies were systematically captured and analyzed.

2.2.5. Study Quality Assessment

The study quality was assessed using the Critical Appraisal Skills Programme (CASP) checklist [30]. The CASP checklist ensures methodological rigor and helps to identify the potential biases in selected studies. Each study was evaluated based on relevance, validity, methodological soundness, bias and confounding factors, reproducibility, transparency, and applicability to blockchain security. In order to rank the studies based on their quality, a scoring system was applied based on their quality, and low-quality studies were excluded and studies without empirical evaluation were excluded during the final synthesis process.

2.3. Data Analysis Process

We started our investigation with 1143 studies from three databases, but we eliminated 8 of them since they were duplicates. Six pieces of research were also disqualified since they were retracted papers. We eliminated a total of 853 papers from our title–abstract screening procedure because they were either not research publications (n = 1) or did not address mitigation, detection, or prevention-related issues (n = 852). Lastly, we removed a total of 24 papers from our full-text analysis since 2 did not fall within the timeframe (2020–2024) that we were interested in, and 22 articles were not available even with institutional access.

The data analysis process involved categorizing attack types based on their mechanisms and assessing their impact on blockchain security and integrity. We gave specific attention to the evolution of attack techniques and motivations. Early attacks predominantly exploited technical flaws, while contemporary threats have evolved into more sophisticated strategies with financial gain as the primary objective. For instance, Sybil attacks leverage multiple fake identities to compromise network integrity, while advanced exploits target vulnerabilities in consensus algorithms such as Proof of Work (PoW) and Proof of Stake (PoS).

To evaluate countermeasures, we analyzed the effectiveness of network security protocols, including firewalls and intrusion detection systems, and cryptographic techniques such as digital signatures and hash functions. Security audits conducted by external experts and consensus mechanisms are critically assessed to determine their resilience against evolving attack methods. The analysis also examined how these measures have adapted over time in response to the growing sophistication of blockchain threats.

The research further explored the anticipated trajectory of blockchain attacks and corresponding security measures. As blockchain technology evolves, new attack vectors are expected to emerge, necessitating continuous advancements in defense mechanisms. Current challenges such as scalability, interoperability, and the trade-offs between decentralization and security are highlighted as critical areas for future investigation.

3. Attacks in Blockchain

3.1. Classification of Blockchain Attacks

Through our systematic literature search, we identified a diverse range of cyberattacks targeting blockchain-decentralized ecosystems, which we classified into distinct categories based on their mechanisms and impact, illustrated in Figure 2. These include quantum computing threats, which exploit advancements in quantum technology to undermine cryptographic security, and smart contract attacks, such as reentrancy vulnerabilities and gas limit manipulations, that exploit flaws in the execution of automated agreements. Additionally, network-based attacks including routing and eclipse attacks aim to disrupt the flow of information and isolate nodes, while transaction-based attacks, including double-spending and race attacks, compromise the integrity of financial operations.

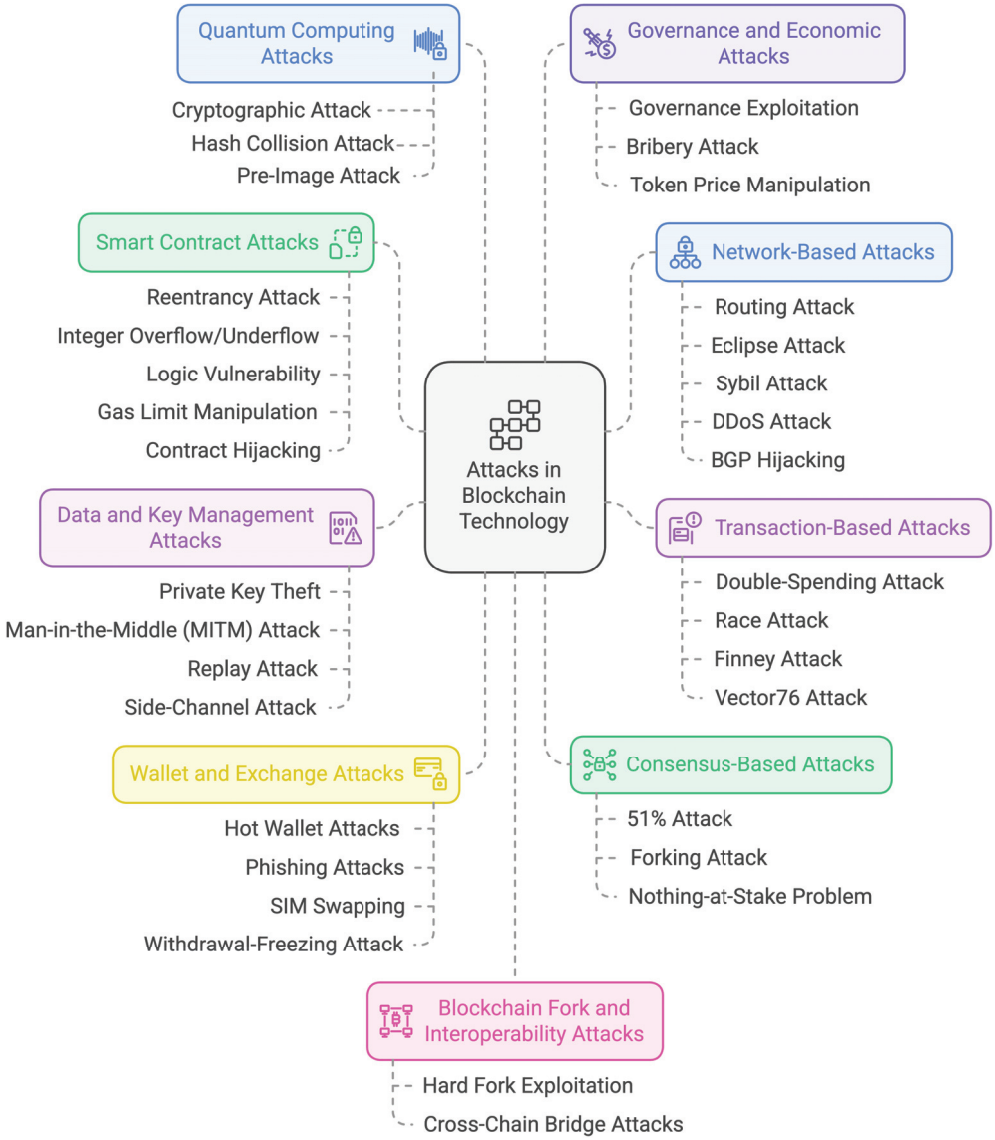


Figure 2. A comprehensive classification of attacks in blockchain technology.

Other identified categories include data and key management attacks, including private key theft and man-in-the-middle exploits, wallet and exchange attacks, including phishing and hot wallet compromises, and consensus-based attacks, such as the 51% attack, which undermines blockchain consensus protocols. Furthermore, blockchain fork and

interoperability attacks exploit chain splits or cross-chain bridges, while governance and economic attacks manipulate decision-making or token economies.

3.1.1. The 51% Attack

Also known as majority attacks, 51% attacks remain one of the most critical threats to blockchain networks [31]. They occur when an attacker or a group of miners gains control of more than 50% of a network's computational power, thereby compromising the consensus mechanism. This control enables malicious actors to manipulate blockchain data and execute attacks such as the following:

(a) Reversing transactions, enabling double spending (spending the same coins multiple times);
(b) Altering the order of transactions;
(c) Disrupting the activities of other miners;
(d) Preventing confirmation of legitimate transactions.

For instance, Ethereum Classic suffered a series of 51% attacks in 2019, causing financial losses and undermining the network's integrity. Indicators of a 51% attack often include abrupt changes in network behavior or transaction anomalies.

3.1.2. Smart Contract Vulnerabilities

Smart contracts, which are self-executing programs running on blockchain networks, play a crucial role in automating and enforcing agreements without intermediaries. Despite their advantages, smart contracts are vulnerable to coding errors and design flaws that can result in unauthorized access or loss of funds [32–34]. These vulnerabilities are particularly critical due to the immutability of blockchain, which makes it challenging to correct errors once deployed [35]. Notable cases, such as the Poly Network hack in 2021, illustrate how attackers exploit smart contract vulnerabilities to siphon funds, demonstrating the need for rigorous security audits and robust coding practices.

3.1.3. Double-Spending Attack

A double-spending attack occurs when an attacker spends the same digital currency or asset more than once, violating the principle of data consistency. This attack is facilitated by exploiting weaknesses in the consensus algorithm or the delay in transaction confirmation. For example, in January 2021, a double-spending attack on Monero's blockchain resulted in the theft of USD 3.3 million [36,37]. Factors contributing to this vulnerability include the following:

- Insufficiently secure or slow consensus algorithms;
- Delayed block confirmation times;
- Acceptance of unverified transactions;
- Direct connections of incoming nodes to the main chain [38,39].

These factors provide adversaries with an opportunity to manipulate transaction records and execute double-spending schemes.

3.1.4. Man-in-the-Middle (MITM) Attack

In a blockchain network, a man-in-the-middle (MITM) attack occurs when an attacker intercepts and manipulates communication between two nodes. This allows the attacker to alter transmitted data, steal private keys, or tamper with transaction records, resulting in significant security breaches, data loss, and financial damage [40–42]. MITM attacks undermine the trust and integrity of blockchain systems, emphasizing the need for secure communication protocols and encryption mechanisms.

3.1.5. Routing Attack

Blockchain networks rely on routing protocols for node communication, transaction propagation, and consensus building. In a routing attack, adversaries manipulate the routing protocol to redirect traffic or compromise nodes [43]. By injecting false routing information, attackers can force nodes to choose malicious paths for transactions and block propagation. This facilitates further attacks, such as the following:

(a) Double-spending attacks;
(b) Denial-of-Service (DoS) attacks;
(c) 51% attacks.

Routing attacks disrupt network functionality and compromise data security, highlighting the necessity of robust routing protocols and real-time network monitoring.

3.1.6. Sybil Attack

A Sybil attack targets peer-to-peer (P2P) networks by allowing a malicious node to create multiple fake identities, disrupting network functionality. This attack is particularly effective in consensus mechanisms like Proof of Work (PoW), where a node's computational power determines its ability to validate transactions and earn rewards. For example, in May 2022, the Ethereum-based DeFi protocol, Saddle Finance, suffered a Sybil attack that exploited vulnerabilities in its on-chain governance system, leading to a loss of over USD 10 million [44]. Similarly, in August 2021, a Sybil attack compromised the Liquid Global cryptocurrency exchange, resulting in losses exceeding USD 90 million [45]. Another study highlights that the node authentication process during the blockchain network's joining phase is particularly vulnerable to Sybil attacks, as attackers can insert malicious nodes that disrupt the network and cause delays in block propagation. This vulnerability underscores the need for robust node authentication mechanisms to prevent the insertion of fake identities.

3.1.7. Race Attack

A race attack, a prevalent form of double-spending attack, occurs when a malicious actor broadcasts two conflicting transactions simultaneously to the network [46,47]. In this scenario, the attacker sends one transaction to the victim, who accepts the payment and delivers the product without waiting for confirmation. Simultaneously, the attacker broadcasts a contradictory transaction to the network, retracting the same amount of cryptocurrency, thereby invalidating the initial transaction [48]. If recipients fail to authenticate the initial transaction properly, the attacker can effectively acquire products or services without making a legitimate payment. This exploit highlights the importance of requiring sufficient confirmations before finalizing transactions to prevent race attacks [49].

3.1.8. Eclipse Attack

An eclipse attack occurs when a malicious actor gains control over all peers connected to a targeted node, isolating it from the rest of the blockchain network. This isolation allows the attacker to manipulate the targeted node, compromise its data, and block legitimate network communication, restricting the node's interactions solely to malicious peers [43,50]. For example, attackers may send a falsified transaction to the isolated node, presenting it as proof of payment, while broadcasting a conflicting transaction to the broader network to double spend the same tokens [50]. By monopolizing the connections of a node, eclipse attacks disrupt network consensus and can have severe implications for blockchain integrity and reliability. Addressing this vulnerability requires implementing diverse connection strategies, random peer selection, and robust routing protocols.

3.1.9. Replay Attack

A replay attack exploits the reuse of valid transaction data to disrupt blockchain networks or steal funds. This attack typically occurs when blockchain ledgers experience hard forks or chain splits, creating separate chains that share a transaction history. In such cases, transactions broadcast on one chain may be replayed on another due to shared transaction data and the absence of replay protection mechanisms [51]. Vulnerabilities arise from incompatible transaction formats, shared historical data, and weak replay protection, particularly in cryptocurrency exchanges, decentralized finance (DeFi) platforms, and blockchain-based supply chain systems [52]. Replay attacks can have significant financial and operational consequences, emphasizing the need for robust replay protection mechanisms, such as unique chain identifiers and transaction tagging, to prevent unauthorized transaction duplication.

3.2. Key Insights and Implications

These attacks highlight the evolving threat landscape within blockchain ecosystems. While blockchain technology offers significant advantages, its adoption across diverse applications necessitates a continuous focus on identifying and mitigating security vulnerabilities. Understanding the root causes and mechanisms of these attacks provides critical insights for developing resilient security measures. As blockchain systems continue to evolve, addressing these challenges remains a vital area for research and innovation. The causes and affected sectors of certain typical blockchain security attack types are summarized in Table 1.

Table 1. A comprehensive overview of blockchain security attacks.

Attack Name	Cause of Vulnerability	Impact on Sectors	References
51% Attack	• Low network hash rate; • Centralized mining power; • Renting computational resources; • Lack of honest economic incentives; • Low attack execution cost.	• Cryptocurrency exchanges; • DeFi platforms; • Blockchain-based supply chains.	[53–55]
Smart Contract Vulnerabilities	• Programming errors; • Complex business logic; • Inadequate testing and auditing; • Lack of standards and best practices.	• DeFi platforms; • Cryptocurrencies; • DAOs; • Supply chain systems.	[56,57]
Man-in-the-Middle Attack	• Intercepted communication between nodes; • Stolen private keys; • Tampered transaction data.	• Blockchain networks; • Cryptocurrency exchanges; • Healthcare.	[40–42]
Replay Attack	• Shared transaction history; • Incompatible message formats; • Lack of replay protection.	• Cryptocurrency exchanges; • DeFi platforms; • Blockchain-based supply chains.	[51,58]
Double-Spending Attack	• Delayed transaction confirmation; • Network latency and forks; • Acceptance of unverified transactions.	• Cryptocurrencies; • Crypto-payment systems; • PoW-based solutions.	[36,38,59]
Routing Attack	• Manipulated routing protocols; • False routing information.	• P2P networks; • PoW/PoS-based systems; • Healthcare.	[43,60]
Sybil Attack	• Fake node insertion in routing tables; • Ignoring node authentication; • Centralized computational authority.	• Cryptocurrencies; • Blockchain-based supply chains; • P2P networks.	[61–63]
Race Attack	• Simultaneous contradictory transactions; • Lack of transaction validation.	• P2P networks; • Cryptocurrency exchanges; • Crypto-payment systems.	[64,65]
Eclipse Attack	• Isolation of nodes by malicious peers; • Exploitation of untrusted peers; • Weak peer-to-peer protocols.	• PoW-based solutions; • Healthcare; • Blockchain networks.	[37,66,67]

3.3. Detecting and Mitigating Malicious Blockchain Attacks

As blockchain technology adoption accelerates, the potential for malicious attacks grows, necessitating robust detection and mitigation strategies. A variety of methods and technologies has been developed to enhance the security and resilience of blockchain networks. These include machine-learning-based activity detection, consensus algorithms to prevent attacks such as double-spending attacks, and cryptographic tools for data integrity. Each subsection focuses on specific types of malicious blockchain attacks, providing a comprehensive understanding of their detection and mitigation techniques.

3.3.1. The 51% Attack

A 51% attack occurs when a single entity or group gains control of more than 50% of a network's mining hash rate or staking power, enabling them to manipulate the blockchain. This attack is particularly relevant to Proof-of-Work (PoW) blockchains like Bitcoin and Litecoin, as well as smaller chains with lower hash rates. The attacker can double spend coins, prevent transactions from being confirmed, and disrupt the network's integrity.

Detection Techniques

Detection techniques for 51% attacks focus on monitoring network hash rate distribution and identifying unusual patterns in block creation. Key methods include the following:

- Hash rate monitoring: Continuous tracking of hash rate distribution among mining pools to detect centralization risks [68].
- Block propagation analysis: Observing block propagation times and orphan rates to identify potential manipulation [69].
- Network consensus monitoring: Analyzing deviations from normal consensus behavior, such as sudden changes in block confirmation times [70].

Mitigation Strategies

To mitigate the risk of a 51% attack, the following strategies are essential:

- Decentralization of mining power: Encouraging a diverse and distributed mining pool ecosystem to reduce the risk of hash rate concentration [69].
- Network size and security: Increasing the overall hash rate and network size to make it economically infeasible for an attacker to gain majority control [68].
- Consensus algorithm enhancements: Transitioning to more secure consensus mechanisms, such as Proof of Stake (PoS), which are less susceptible to hash-rate-based attacks [70].
- Real-time alerts and response systems: Implementing systems to detect and respond to unusual network activity, such as sudden hash rate spikes or block reorganizations [69].

By adopting these measures, blockchain networks can reduce the likelihood of a 51% attack and maintain the integrity and trustworthiness of their systems.

3.3.2. Smart Contract Vulnerabilities

Smart contracts, being self-executing and immutable, are susceptible to vulnerabilities that can lead to significant security breaches. Detection techniques for these vulnerabilities are broadly categorized into static and dynamic analysis methods.

Detection Techniques

Static analysis methods include the following:

- Symbolic execution: Code is analyzed with symbolic values, enabling the generation of algebraic terms and propositional formulas to uncover logical errors [71,72].
- Control flow graph (CFG) construction: Represents execution flow as a directed graph, aiding understanding of the program's structure [71,72].
- Pattern recognition and rule-based analysis: Matches code against predefined secure patterns or known vulnerabilities [57,72].
- Decompilation analysis: Translates lower-level bytecode into higher-level representations for easier parsing and vulnerability identification [71].
- Formal verification: Uses mathematical proofs to ensure the program adheres to its specified properties [73].

Dynamic analysis methods include the following:

- Runtime Execution Trace: Captures the sequence of instructions executed during runtime for anomaly detection [71].
- Fuzz Testing: Provides structured data as inputs to trigger unexpected behavior, such as crashes or abnormal execution paths [72].

Mitigation Strategies

To enhance smart contract security, the following measures are crucial:

- Adopting established design patterns and best practices to minimize coding errors.
- Implementing robust access control mechanisms to prevent unauthorized interactions.
- Conducting thorough security audits and regular code reviews to identify potential vulnerabilities.
- Utilizing comprehensive testing methods, including fuzz testing and formal verification [73].
- Deploying upgradeable contracts to allow future improvements without disrupting the system.
- Monitoring deployed contracts and addressing detected vulnerabilities proactively [71,72].

3.3.3. Man-in-the-Middle (MITM) Attack

Man-in-the-middle (MITM) attacks involve intercepting and manipulating communication between nodes in a blockchain network. These attacks can compromise data integrity, steal sensitive information, and undermine trust within the network.

Detection Techniques

- Network Monitoring: Detects unusual activity, such as rogue nodes or unexpected data transmissions [74].
- Consensus Checks: Identifies discrepancies in transaction data and halts suspicious processes [75].
- Digital Signatures: Ensures data integrity and authenticates transaction senders, preventing tampering [76].
- Reputation Systems: Tracks node behavior to identify and flag potentially malicious nodes [77].

Mitigation Strategies

- Encryption: Ensures that data transmitted between nodes remain secure and incomprehensible to attackers [78].

- Multi-Factor Authentication (MFA): Protects private keys using techniques such as biometric authentication and one-time passwords [79].
- Consensus Mechanisms: Requires multiple nodes to verify and approve transactions, making it difficult for attackers to manipulate data [80].
- Identity Management: Authenticates and authorizes nodes before allowing them to join the network using tools like a public key infrastructure and digital certificates [81].
- Real-Time Monitoring: Combines network and transaction monitoring to detect and mitigate potential MITM attacks [82].

Effective detection and mitigation of MITM attacks safeguard blockchain networks by ensuring data confidentiality, integrity, and authenticity. Table 2 summarizes the findings.

Table 2. A summarized overview of the detection and mitigation techniques for blockchain attacks.

Attack	Detection Techniques	Security Measures (Mitigation)	References
51% Attack	• Monitor unusual network behavior; • Analyze mined blocks and block reorganization; • Detect double spending and suspicious transactions; • Observe mining pool activities; • Implement alerting mechanisms.	• Use robust consensus mechanisms; • Increase network size and decentralization; • Apply checkpoints and diverse mining communities.	[83–85]
Smart Contract Vulnerabilities	• Static analysis (e.g., symbolic execution, control flow graphs); • Dynamic analysis (e.g., fuzz testing, runtime monitoring); • Formal verification and code auditing.	• Adopt best design patterns and access control mechanisms; • Conduct regular security audits; • Use upgradeable contracts and comprehensive testing.	[57,71,72]
Double-Spending Attack	• Monitor transaction verification times; • Analyze block confirmation and ledger consistency; • Detect suspicious fork creation.	• Increase confirmation times and block consistency; • Use key-evolving cryptography; • Disable unconfirmed connections.	[36,59,86]
Race Attack	• Monitor transaction broadcasts; • Insert observer nodes for suspicious activity.	• Periodic transaction monitoring; • Deploy observer nodes for continuous surveillance.	[46,87]
Replay Attack	• Detect duplicate transactions; • Analyze transaction history in forked chains.	• Implement strong replay protection; • Use coin-splitting techniques; • Opt-in replay protection.	[51,58]
Sybil Attack	• Monitor node reputation and activity; • Identify suspicious node behavior during PoW initiation.	• Restrict high-computing-power nodes; • Implement non-refundable deposits (e.g., currency burning); • Use fidelity bonds and coin-age mechanisms.	[61,62]
Eclipse Attack	• Use anomaly detection tools; • Run seeding and eliminate exploitation windows.	• Implement timestamp-based protocols; • Use threat database models for detection; • Enhance network load monitoring.	[37,66]
Man-in-the-Middle Attack	• Detect unusual network activity; • Check transaction discrepancies; • Monitor potentially malicious nodes.	• Encrypt communication between nodes; • Use multi-factor authentication for private key access; • Employ public key infrastructure and biometric authentication.	[74,76]
Routing Attack	• Monitor network traffic patterns; • Use reputation systems to identify malicious nodes.	• Secure routing protocols using digital certificates; • Apply distributed consensus mechanisms.	[88,89]

3.3.4. Routing Attack

Detecting routing attacks in blockchain networks is challenging due to the difficulty of distinguishing between benign and malicious network behavior. However, several techniques have been proposed to identify such attacks effectively. Network monitoring is a key technique, allowing the tracking of transaction routing paths and the detection of anomalies in network traffic [88]. For instance, a transaction propagating through an unusual path may indicate a routing attack [89]. Data analysis can uncover patterns and trends in network traffic that suggest malicious activities. For example, a sudden increase in the number of transactions routed through a specific node may signal a routing attack [90,91]. Reputation systems are also critical, assigning scores to nodes based on their behavior. Nodes with low reputation scores can be flagged as potential sources of routing attacks [92].

To mitigate routing attacks, several strategies have been proposed. Secure routing protocols, such as BGPsec and RPKI, use digital certificates to authenticate routing information and prevent spoofing attacks [93]. Distributed consensus mechanisms, including Proof of Work (PoW) and Proof of Stake (PoS), increase the difficulty for attackers to control the majority of the network's computational power or stake, thereby preventing routing attacks [94]. Peer-to-peer networking enhances network resilience by allowing nodes to communicate directly, reducing reliance on centralized routing authorities [95]. Furthermore, node reputation systems can identify malicious or compromised nodes, preventing them from propagating transactions and blocks in the network [96]. Nodes with low reputation scores can be excluded or subjected to additional scrutiny, thereby limiting the risk of routing attacks.

3.3.5. Race Attack

Race attacks have been extensively studied, with several detection techniques proposed. A key detection method is the Listening Period, during which vendors associate each transaction with a listening time frame. During this period, transactional records are scrutinized for any signs of a race attack. If no malicious activity is detected within this time frame, the vendor proceeds with the transaction [97]. Another effective technique is the Insertion of Observers, where vendors deploy monitoring nodes within the network. These observers transmit all transactions to the vendor, enabling swift detection of race or double-spending attacks [97].

To prevent race attacks, several mitigation strategies can be employed. Timestamping all transactions ensures that the processing time is recorded, allowing the first recorded transaction to be considered valid. Cryptographic signatures authenticate transactions, verifying that they were submitted by the intended user. Multi-party signatures enhance security by requiring multiple parties to approve a transaction, reducing the risk of unauthorized activities. Increasing the block processing time also mitigates race attacks, as attackers face greater difficulty in executing multiple fraudulent transactions within a restricted timeframe [64].

3.3.6. Eclipse Attack

Eclipse attacks have been addressed in various studies, with several detection approaches proposed. One method involves analyzing the timestamps of questionable blocks. If the time interval between newly created blocks is abnormally high, it may indicate an eclipse attack, with detection typically requiring 2–3 h [66]. Another technique is the ubiquitous gossip protocol, which allows users to connect with protocol-dominated servers through gossip messages without requiring changes to the Bitcoin protocol or the peer-to-

peer network. This approach minimizes dependency on web servers for connecting with Bitcoin block headers [98]. Additionally, redundancy can be introduced by replicating keys across multiple nodes and conducting routing failure tests to identify compromised nodes.

To mitigate eclipse attacks, several countermeasures are available. Elimination of Far Successors is a technique where nodes autonomously calculate the distances between their successors, removing any malicious entries immediately [99]. Bitcoin clients also use a classification system to segregate new peers and previously established connections, preventing attackers from exploiting these distinctions [43]. Expanding the network size further complicates an attacker's ability to control a significant portion of the network. Peer-discovery mechanisms allow nodes to connect with a diverse set of peers, thereby reducing the likelihood of isolation and increasing resilience against eclipse attacks.

3.3.7. Double-Spending Attack

Double-spending attacks pose a significant threat to blockchain systems by allowing malicious actors to spend the same digital asset multiple times, undermining transaction integrity. Several detection and prevention mechanisms have been developed to mitigate this risk effectively.

Key detection techniques include transaction and block verification processes, where the transaction time and block confirmation are monitored to identify inconsistencies or unusual behavior [36]. Updating the existing block rather than removing it and implementing timestamping during chain-building processes further enhance detection accuracy [86]. Approaches such as the Longest Chain Rule in Proof-of-Stake (PoS) protocols, key-evolving cryptography, and increasing the number of confirmed blocks have also been shown to effectively identify double-spending attempts [100]. Continuous monitoring of node connections, disabling unconfirmed connections, and employing transaction forwarding mechanisms ensure additional layers of security [37,59].

Mitigation measures for double-spending attacks include adopting recipient-oriented transactions and increasing block confirmation thresholds, which make fraudulent transactions more challenging to execute. Cryptographic methods such as key-evolving cryptography provide secure mechanisms to verify transaction authenticity. Additionally, ensuring a consistent ledger and disabling unconfirmed connections prevent the propagation of invalid transactions [39]. These strategies collectively enhance blockchain reliability and maintain transactional trustworthiness.

3.3.8. Sybil Attack

Sybil attacks exploit vulnerabilities in peer-to-peer (P2P) networks by introducing multiple malicious nodes that mimic legitimate nodes, thereby disrupting network operations. Addressing these attacks requires a combination of detection and mitigation strategies.

TrustChain, a blockchain solution incorporating the NetFlow algorithm, calculates node reputation to detect and mitigate Sybil attacks by identifying malicious nodes [61]. Decentralized registration mechanisms prevent Sybil node injection into routing tables, addressing vulnerabilities such as Routing Table Insertion (RTI) attacks [62]. Identity verification during the initial node authentication phase, as proposed by Tayebeh et al., can constrain nodes exhibiting abnormally high computational power, mitigating risks to the network [63]. Furthermore, imposing non-refundable deposits, applying time locks to fund usage, and leveraging coin-age mechanisms can reduce the likelihood of Sybil nodes affecting block propagation and consensus [101].

Effective countermeasures also include implementing rigorous node authentication processes, which may involve network joining fees, source validation of node connections, and monitoring nodes' forwarding behavior over time [102]. By integrating these techniques, blockchain networks can significantly reduce the impact of Sybil attacks and ensure network stability.

3.3.9. Replay Attack

Replay attacks exploit vulnerabilities in blockchain systems, particularly during hard forks or chain splits, by replaying valid transactions from one chain onto another. Detection techniques include monitoring for duplicate transactions across chains and analyzing transaction histories to identify repeated or suspicious activities [52]. Incorporating cryptographic measures, such as one-time private–public key pairs and elliptic-curve-based encryption, further strengthens detection capabilities [87].

Mitigation strategies for replay attacks include implementing strong replay protection, which ensures transactions made on a new blockchain post-fork cannot be replayed on the original chain. Splitting coins into distinct transactions on each chain prevents cross-chain replay vulnerabilities [51]. Opt-in replay protection allows users to mark transactions manually, granting additional control over transaction security [52]. These measures collectively enhance the resilience of blockchain systems against replay attacks.

4. Our Findings

In our study, we identified several critical findings that highlight the significance of addressing blockchain security vulnerabilities. Common attacks, including double-spending attacks, 51% attacks, and smart contract vulnerabilities, pose substantial threats to blockchain networks by undermining their security and integrity, leading to financial losses and eroding public trust.

4.1. Probability of Blockchain Attacks

We attempted to identify the probability of various blockchain attack types, which varies significantly based on the underlying vulnerabilities and the operational design of blockchain systems. Smart contract vulnerabilities exhibit the highest probability compared to other types of vulnerabilities in almost every case, and there are different key factors, including code vulnerabilities, questionable audit quality, and lack of formal verifications. These vulnerabilities often lead to attacks like reentrancy exploits, which are particularly devastating in decentralized finance (DeFi) platforms.

Table 3 illustrates the analysis of the likelihood of various blockchain attacks, and reveals significant variations in attack probabilities. Man-in-the-middle (MITM) attacks and double-spending attacks also demonstrate relatively high probabilities, at around 35–45%. The prevalence of MITM attacks arises from weaknesses in communication protocols and insufficient encryption mechanisms in blockchain networks. Double-spending attacks, meanwhile, are facilitated by delayed transaction confirmations or inadequate consensus mechanisms in Proof-of-Work (PoW) or hybrid systems.

Other attack types, such as Sybil attacks, leverage weaknesses in node authentication processes, allowing attackers to disproportionately influence network consensus or disrupt data propagation. Similarly, routing attacks and eclipse attacks exhibit moderate likelihoods, targeting network-layer vulnerabilities to isolate or delay nodes. Lower-probability attacks, like replay attacks, occur in specific contexts such as double spending or hard forks but remain critical threats in those scenarios.

These probabilities highlight the higher risks associated with certain attack types, such as double-spending attacks and smart contract vulnerabilities. Prioritizing security measures for these high-risk areas can significantly enhance blockchain resilience. These findings provide critical insights for researchers and practitioners, offering a roadmap for addressing blockchain security challenges effectively.

Table 3. Blockchain attack targets and key factors influencing likelihood for different types of attacks.

Attack Type	Target	Key Factors Influencing Likelihood
51% Attack	PoW (Bitcoin, Litecoin, Small Chains)	Hash rate concentration, mining cost, network difficulty
Sybil Attack	PoS, Permissionless Chains	Node diversity, validator stake, network entry barriers
Eclipse Attack	P2P Nodes	Network size, node connectivity, topology
Smart Contract Vulnerabilities	Ethereum, BSC, DeFi Contracts	Code vulnerabilities, audit quality, formal verification
Double-Spending Attack	PoW/PoS Networks	Transaction finality, block confirmation time
Routing Attack	All Blockchain Networks	ISP dependency, network propagation speed
Replay Attack	Transaction Authentication	Weak authentication, transaction replay capability
Man-in-the-Middle Attack	Blockchain Communication Layer	Unencrypted communication, weak key exchange
Race Attack	PoW/PoS Networks (Fast Transaction Confirmations)	Transaction propagation speed, network latency, low block confirmation requirements

4.2. Detection and Mitigation Strategies for Blockchain Attacks

Our study comprehensively analyzed the diverse range of technologies and security measures available for detecting and mitigating blockchain attacks. These strategies encompass advanced consensus algorithms, such as Byzantine Fault Tolerance (BFT), Proof of Stake (PoS), and Delegated Proof of Stake (DPoS), which enhance the network's resilience against consensus-based attacks like 51% attacks and Sybil attacks. Cryptographic techniques, including zero-knowledge proofs (ZKPs), multi-signature schemes, and lattice-based post-quantum cryptography, serve as robust safeguards against key management vulnerabilities, ensuring the integrity of transactions and the confidentiality of user identities.

Additionally, smart contract auditing tools and methodologies, such as static and dynamic analysis, formal verification, and fuzz testing, play a pivotal role in identifying vulnerabilities like reentrancy attacks or logic flaws within smart contracts. Secure development practices, such as adhering to well-defined design patterns, conducting regular code reviews, and implementing access control mechanisms, further mitigate the risks associated with poorly designed systems. Industry-specific measures, such as secure oracles for blockchain-based supply chains or privacy-preserving encryption techniques for healthcare applications, highlight the necessity of tailoring security solutions to specific use cases.

Given the dynamic nature of blockchain security, continuous research and cross-disciplinary collaboration are imperative. Emerging threats, such as quantum computing risks and cross-chain interoperability vulnerabilities, demand proactive development of quantum-resistant cryptography and secure protocols for decentralized applications. By staying informed about evolving attack vectors and integrating state-of-the-art technologies, the blockchain community can build robust security frameworks that ensure the resilience and trustworthiness of blockchain systems across diverse applications. This approach not only safeguards existing implementations but also fosters trust in and broad adoption of blockchain technology.

4.3. Significant Keyword Frequencies

In our analysis, we found that the keyword frequencies provide significant insights into the research trends and priorities within the field. The term "Blockchain" emerged as the most frequently mentioned keyword, appearing 16,400 times, which underscores its central role in the studies. This was closely followed by "Security" (11,518) and "Smart" (13,760), indicating a strong emphasis on security aspects and smart technologies, particularly in the context of blockchain systems. We also observed that keywords such as "Vulnerability/Vulnerabilities" (8422), "Attack" (5450), and "Detection" (7043) were highly prevalent, reflecting a substantial focus on identifying and addressing security threats and weaknesses within blockchain and related technologies. Additionally, terms like "Privacy" (4454) and "Encryption" (1633) highlight the importance of data protection and cryptographic methods in ensuring secure transactions and network integrity. The prominence of "Ethereum" (4672) and "Bitcoin" (1971) further emphasize the dominance of these platforms in blockchain research. Overall, our findings illustrate a research landscape deeply engaged with blockchain technology, its security challenges, and the development of strategies to mitigate vulnerabilities and enhance system robustness. Table 4 illustrates the significant keywords from our included studies.

Table 4. Keyword frequencies of the included studies.

Keyword	Count
Technology/Technologies	4378
Security	11,518
Systems	3697
Blockchain	16,400
Distributed	2112
Privacy	4454
Information	5438
Control	2881
Encryption	1633
Detection	7043
Prevention	381
Attack	5450
Vulnerability/Vulnerabilities	8422
Protocol	1901
Transaction	5249
Network	6848
Bitcoin	1971
Ethereum	4672
Smart	13,760
Internet	2375
Cryptocurrency/Cryptocurrencies	1221
Strategies	554
Strategy	590
Mitigation	385
Cryptography	642
IoT	633
Tether	20

4.4. Publication Frequencies by Year

As shown in Figure 3, our results revealed a significant trend in the included studies' yearly publication frequency. According to the data, just 22 pieces of research were published in 2020, the year with the fewest publications, but, in the years that followed,

there was a noticeable rise, with 39 publications in 2021 and 37 in 2022. After reaching 58 publications in 2023, the rising trend continued, culminating in a noteworthy peak of 118 publications in 2024. This dramatic increase in publications, especially in 2024, suggests that interest in the topic is expanding and that research is moving more quickly. The rise might be explained by the growing importance of blockchain technology, its uses, and the security issues that surround it, which have drawn more attention from both researchers and industry professionals. This pattern emphasizes how dynamic and quickly changing the area is, which reflects how crucial it is to solve new problems and advance our understanding of blockchain and related fields.

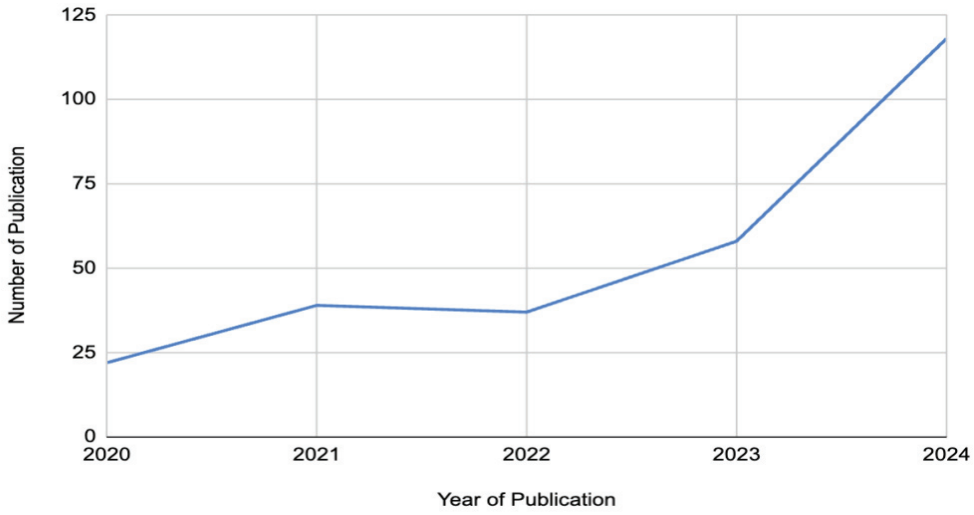


Figure 3. Publication frequency based on included studies.

4.5. Distribution of Attack Categories

In our study, we determined the distribution of attack types and their frequencies across the included studies. Also, we categorized the attacks into 42 distinct categories based on their characteristics and impact areas. The category "Smart Contract Vulnerability" received the greatest attention, with 48 mentions, emphasizing its importance as a significant area of concern in blockchain and related technologies. This was followed by "Financial Fraud" (18) and "Privacy Violation" (19), both of which received significant attention, emphasizing the need for tackling financial and privacy threats in digital ecosystems. Other important categories are "Network Attack" (10), "Cryptographic Attack" (9), and "Denial of Service" (8), emphasizing the importance of network security, cryptographic vulnerabilities, and service disruption threats. Additionally, the term "IoT Vulnerability" occurred 22 times, showing a rising level of worry about the security of Internet of Things (IoT) systems. Categories such as "Authentication/Authorization" (17) and "Data Security" (4) highlight the significance of safe access control and data protection techniques. The inclusion of less common but significant categories, such as "Supply Chain Attack" (2), "Threat Intelligence" (1), and "Trusted Execution Environment" (1), implies that a wide variety of security concerns is being investigated. Overall, these findings point to a research environment that emphasizes smart contract vulnerabilities, financial fraud, and privacy concerns while also addressing a wide range of additional security threats and weaknesses in blockchain and associated technologies. Figure 4 demonstrates the distribution of attack categories.

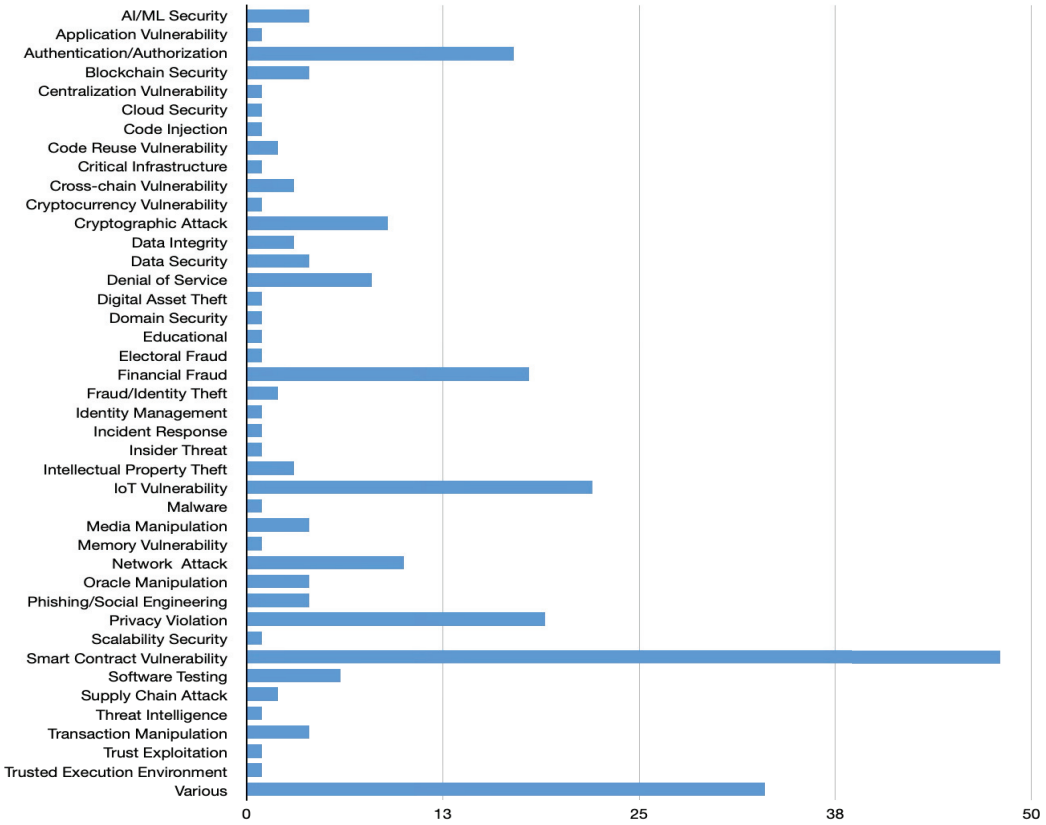


Figure 4. Distribution of attack categories and their frequencies in the included studies.

4.6. Analysis of Core Reasons for Vulnerabilities

In our research, we discovered the primary causes of vulnerabilities throughout the included studies, as shown in Figure 5. The most common explanation was "Software Vulnerability", which had 35 instances, showing the ubiquity of faults in software systems as a key source of security hazards. This was followed by "Access Control Weakness" (23) and "AI Model Bias" (21), which emphasizes the difficulties in protecting access mechanisms and resolving biases in artificial intelligence models. Furthermore, "IoT Security Weakness" occurred 15 times, indicating rising worries about the security of Internet of Things (IoT) devices. With 10 instances apiece, "Blockchain Security Issue" and "Cryptographic Flaw" are two more noteworthy explanations that highlight how critical it is to fix flaws in blockchain systems and cryptographic implementations. Although they are less common, problems like "Human Manipulation" (3), "Third-Party Risk" (3), and "Data Privacy Issues" (8) also add to the overall vulnerability illustration.

The category "Other" had the highest count (117), which represents a broad range of less frequently cited but equally critical reasons for vulnerabilities. These include emerging threats, such as novel attack techniques not yet widely studied; regulatory and compliance gaps, where security weaknesses arise from a lack of standardized regulations, particularly in emerging technologies like blockchain and AI; economic and incentive misalignment, where cost-cutting or misaligned incentives in decentralized finance (DeFi) lead to security flaws; legacy system weaknesses, where outdated or deprecated technologies remain susceptible to known exploits; interoperability and integration flaws, caused by poor integration between platforms or systems; lack of security culture and awareness, where organizational or user behavior leads to vulnerabilities, such as developers neglecting best practices or users falling for scams; and Protocol Design Flaws, where inherent weaknesses in system architectures, such as blockchain consensus mechanisms or smart contract

frameworks, create exploitable vulnerabilities. These findings emphasize the diverse and multifaceted nature of security weaknesses, ranging from technical flaws to human, organizational, and systemic factors, and highlight the need for comprehensive strategies to mitigate these risks.

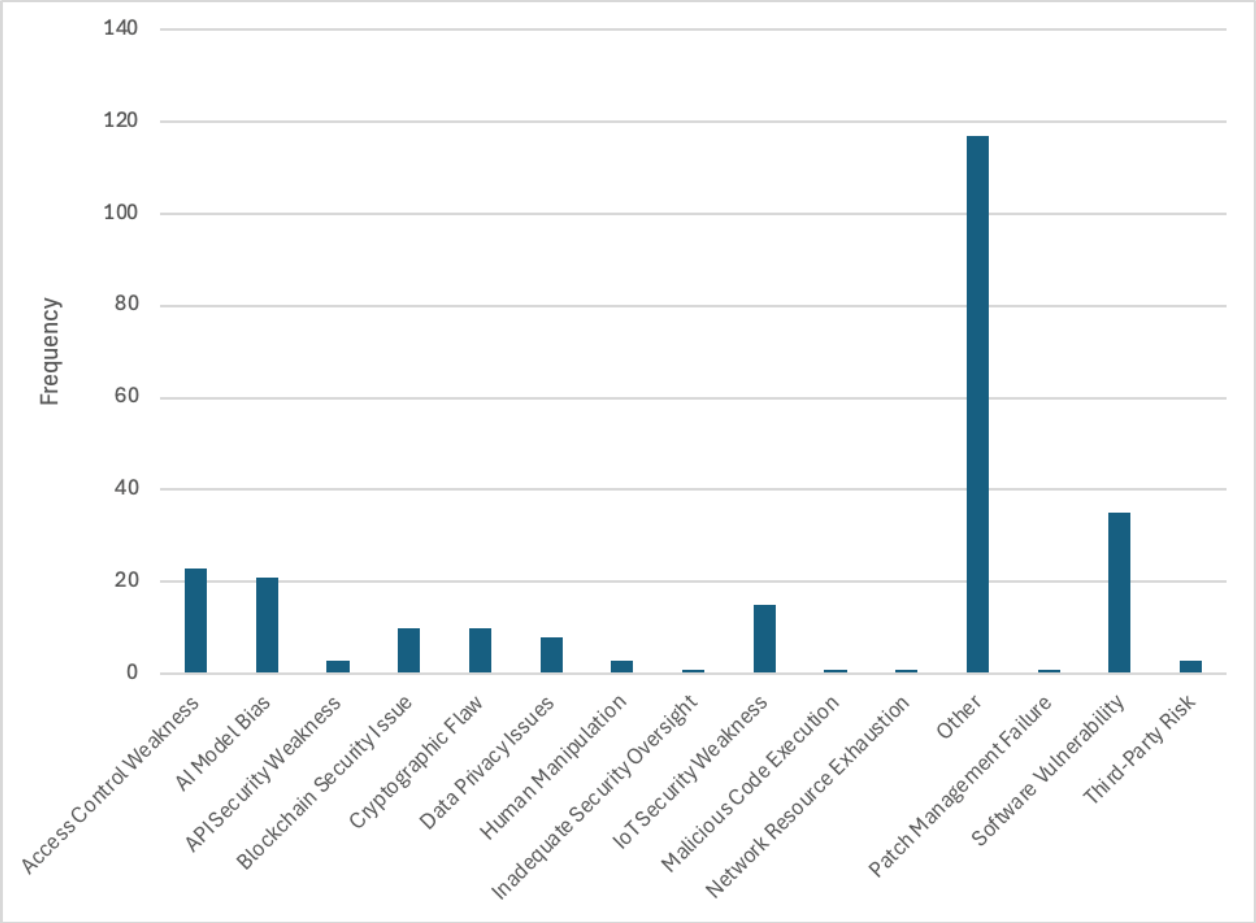


Figure 5. Distribution of core reasons for vulnerabilities across the included studies.

4.7. Attack Categories and Their Impact Results Across the Included Studies

In order to systematically analyze the effect outcomes of assaults throughout the available research, we divided them into 42 different categories. This classification made it possible for us to recognize the wide variety of vulnerabilities and the effects they have on different fields. For example, studies have emphasized the security concerns of AI/ML, including financial exploitation threats and poor vulnerability identification [103–106]. Likewise, Web3 apps' application vulnerability resulted in execution errors that went unnoticed and possible security compromises [107]. Studies [69,70,108–116] that examined the authentication/authorization category found a variety of effects, such as private key breaches, illegal access, and legal liabilities. Performance deterioration and election result manipulation were two major blockchain security issues [117–120]. This thorough taxonomy highlights the complexity of security issues and their wide-ranging effects while offering an organized summary of the threat landscape. We want to enable focused mitigation efforts and more investigation into certain areas of susceptibility by classifying these findings into discrete categories. Table 5 summarizes our findings in different attack categories and their impact results across the included studies.

Table 5. Cybersecurity attacks and their impacts based on their attack categories.

Reference(s)	Attack Category	Impact Result
[103–106]	AI/ML Security	Unreliable vulnerability detection, financial exploitation risks, missed security flaws in contract audits, and unpatched vulnerabilities in deployed contracts.
[107]	Application Vulnerability	Undetected execution faults in Web3 applications, leading to potential security breaches.
[69,70,108–116]	Authentication or Authorization	User errors, private key leaks, unauthorized access attempts, patient data exposure, healthcare service disruptions, legal liabilities, data breaches, loss of intellectual property, regulatory penalties, compromised data integrity, financial loss, reputation damage, operational disruption, unauthorized vehicle use, financial fraud, identity theft, data theft, and unauthorized smart contract execution.
[117–120]	Blockchain Security	Performance degradation, increased attack surface, data tampering risks, election result manipulation, transaction bottlenecks, and reduced performance.
[121]	Centralization Vulnerability	Anonymity loss and privileged operations due to centralized control.
[122]	Cloud Security	State continuity violations in cloud-based systems.
[123]	Code Injection	Database manipulation through malicious code injection.
[124,125]	Code Reuse Vulnerability	Propagation of known security flaws and widespread deployment of vulnerable contracts.
[126]	Critical Infrastructure	Grid data manipulation in critical infrastructure systems.
[127–129]	Cross-Chain Vulnerability	Unintended contract behavior exploitation, cross-chain transaction fraud, and privacy leaks.
[130]	Cryptocurrency Vulnerability	Bitcoin loss due to script flaws in cryptocurrency systems.
[68,131–137]	Cryptographic Attack	Unauthorized data access in smart grids, data breaches, loss of user trust, legal liabilities, data theft, espionage, compromised encryption, and privacy leakage.
[138–140]	Data Integrity	Loss of critical digital evidence, unauthorized modifications, and compromised tenant data.
[141–144]	Data Security	Data loss, unauthorized access, medical data breaches, identity theft, and loss of control over information.
[145–152]	Denial of Service	Service disruption, network security compromise, financial losses, blockchain slowdown, transaction delays, and smart contract failure.
[153]	Digital Asset Theft	Player data compromises in digital asset systems.

Table 5. *Cont.*

Reference(s)	Attack Category	Impact Result
[154]	Domain Security	Phishing attacks and lost domain access due to security vulnerabilities.
[155]	Educational	Knowledge gaps in blockchain security education and training.
[156]	Electoral Fraud	Vote tampering and lack of transparency in electoral systems.
[157–173]	Financial Fraud	Financial fraud, illicit contract use, fund misuse, investor losses, scam tokens, unfair governance token distributions, and cyber threats to financial stability.
[174,175]	Fraud/Identity Theft	Fake degrees, credential fraud, compromised health data, and fraudulent registrations.
[176]	Identity Management	Higher threat exposure in self-sovereign identity systems.
[177]	Incident Response	Delayed security responses and uncoordinated mitigation efforts.
[178]	Insider Threat	Compromised smart grid infrastructure due to insider threats.
[179–181]	Intellectual Property Theft	Revenue loss for copyright holders, unauthorized data usage, and copyright infringement.
[119,182–201]	IoT Vulnerability	Data interception, unauthorized access, widespread IoT network vulnerabilities, data leaks, privacy issues, sensitive data exposure, financial loss, and compromised industrial control systems.
[202]	Malware	Data loss and ransom demands due to malware attacks.
[203–206]	Media Manipulation	Public misinformation, identity theft, political and financial misinformation risks, fake news, and lack of content authenticity.
[207]	Memory Vulnerability	Memory corruption and code execution vulnerabilities.
[89,196,208–215]	Network Attack	Operational disruptions, compromised data integrity, financial losses, service disruption, data tampering, cybercriminal activities, and denial-of-service (DoS) attacks.
[216–219]	Oracle Manipulation	Flash-loan-based financial exploits, price manipulations, inaccurate data inputs, and market manipulation.
[152,220–222]	Phishing/Social Engineering	User credential theft, financial fraud, and unauthorized fund theft.
[223–240]	Privacy Violation	Privacy violations in AI models, user identity theft, misinformation, data exposure, identity leaks, compromised vehicle safety, legal and financial risks, and unauthorized access to stored blockchain data.

Table 5. *Cont.*

Reference(s)	Attack Category	Impact Result
[241]	Scalability Security	High fees and slow transactions due to scalability issues.
[163,242–286]	Smart Contract Vulnerability	Financial exploits, blockchain instability, unauthorized fund transfers, contract hijacking, financial fraud, money laundering, and irreversible financial losses.
[287–292]	Software Testing	Security vulnerabilities overlooked, undetected software vulnerabilities, and unreliable security testing results.
[293,294]	Supply Chain Attack	Malfunctioning electronic components and compromised software due to supply chain attacks.
[295]	Threat Intelligence	Slow response to cyber threats due to inadequate threat intelligence.
[296–299]	Transaction Manipulation	Financial loss, disruption of services, unauthorized transactions, and financial manipulation.
[300]	Trust Exploitation	False service discovery and compromised interactions due to trust exploitation.
[301]	Trusted Execution Environment	Vulnerabilities in trusted execution environments.
[109,123,278,302–329]	Various (studies that discuss attacks from multiple categories)	Online fraud, identity theft, unauthorized access to ECUs, denial of service (DoS), financial losses, privacy violations, data breaches, network collapse, theft of NFTs, and unauthorized code execution.

5. Discussion

Blockchain technology has revolutionized various industries by offering a decentralized and secure platform for transactions and data management. However, its potential is often undermined by persistent security challenges that threaten the integrity, confidentiality, and availability of blockchain networks. This research explored the security vulnerabilities in blockchain systems, emphasizing the detection and mitigation strategies for various attacks. The findings address three core research questions (RQs) and highlight their practical and theoretical implications, paving the way for more secure and resilient blockchain networks.

5.1. RQ1: Common Types of Attacks on Blockchain Technology and Their Impact

Blockchain technology faces several common attack types that compromise security and integrity:

- Smart contract vulnerabilities: Reentrancy attacks, integer overflow, and weak access control mechanisms lead to financial fraud and unauthorized transactions.
- Denial-of-service (DoS) attacks: DDoS blockchain state storage attacks cause network congestion and transaction delays, impacting system reliability.
- Consensus attacks: 51% attacks, selfish mining, and long-range attacks exploit mining power to reverse transactions and double spend assets.
- Oracle manipulation: Flash loan exploits and price manipulation enable attackers to control asset prices and execute fraudulent trades in DeFi applications.
- Cryptographic attacks: Quantum cryptographic threats, identity theft, and weak key management lead to unauthorized access and data breaches.

- Privacy violations: Data leaks in EHRs and IoT privacy breaches due to weak encryption mechanisms expose user-sensitive data.
- IoT security weaknesses: IoT device hijacking, unauthorized access, and industrial IoT intrusions result in compromised network security.

These attacks cause financial losses, data breaches, network disruption, and erosion of trust in blockchain systems.

5.2. RQ2: Security Measures and Technologies for Detecting and Mitigating Blockchain Attacks

To counter malicious blockchain attacks, various security measures are employed:

- Detection Techniques:
  - AI and ML-based anomaly detection for fraud and transaction manipulation.
  - Static and dynamic smart contract analysis for vulnerability detection.
  - Cryptographic verification ensuring transaction integrity.
  - Consensus monitoring for anomaly detection in mining behavior.
- Mitigation Measures:
  - Secure smart contract coding practices and formal verification.
  - Blockchain-based authentication for secure access control.
  - Privacy-preserving mechanisms such as zero-knowledge proofs (ZKP) and homomorphic encryption.
  - Hybrid blockchain models integrating public and private blockchains for enhanced security.
- Prevention Techniques:
  - Decentralized identity management with multi-factor authentication.
  - Intrusion prevention systems (IPS) for blocking unauthorized transactions.
  - Tokenization and encrypted storage for securing sensitive data.

These security mechanisms help ensure blockchain resilience against various threats.

5.3. RQ3: Context-Specific Mitigation Strategies for Different Blockchain Applications

Different blockchain applications require tailored security approaches:

- Smart Contracts (DeFi and Financial Transactions):
  - Pre-deployment formal verification to identify coding errors.
  - Multi-signature authentication to prevent unauthorized withdrawals.
  - Real-time transaction monitoring for Ponzi scheme detection.
- Enterprise and Government Blockchains:
  - Hybrid blockchain integration balancing transparency and confidentiality.
  - Regulatory compliance mechanisms such as AML and KYC frameworks.
  - Decentralized governance models using DAOs for decision-making.
- IoT and Industrial Blockchain Applications:
  - Lightweight blockchain security protocols for resource-constrained devices.
  - Edge and fog computing integration to secure IoT data.
  - Zero-knowledge proofs for securing IoT-generated data.
- Privacy-Critical Applications (Healthcare, Identity Management):
  - Decentralized identity management to prevent unauthorized access.
  - Confidential transactions with advanced encryption mechanisms.

- Blockchain-Based Voting and Governance:
  - Cryptographic vote verification for secure elections.
  - Resilient consensus mechanisms to prevent rollback attacks.

By implementing application-specific security strategies, blockchain networks can achieve better security, compliance, and efficiency.

5.4. Broader Implications

This research highlights the evolving nature of blockchain security, where the increasing sophistication of attack vectors necessitates continual advancements in detection and mitigation strategies. The integration of machine learning and artificial intelligence into anomaly detection, as well as the exploration of post-quantum cryptographic techniques, represent promising directions for future research.

Additionally, the environmental impact of resource-intensive consensus mechanisms, such as Proof of Work (PoW), underscores the importance of exploring energy-efficient alternatives like Proof-of-Stake (PoS) and hybrid models. The scalability and adoption of these solutions will play a critical role in ensuring blockchain technology's long-term sustainability.

5.5. Practical and Theoretical Contributions

From a theoretical perspective, this study provides a structured taxonomy of blockchain attacks, detailing their mechanisms, causes, and countermeasures. Practically, it offers actionable recommendations for blockchain practitioners and industry stakeholders, emphasizing the importance of tailoring security measures to meet application-specific needs. These contributions bridge the gap between academic research and practical implementation, facilitating the development of secure blockchain ecosystems.

6. Conclusions

This research provides a comprehensive analysis of mitigating blockchain attacks and enhancing security measures, aligning with the transformative potential of blockchain technology across industries. Blockchain's decentralized nature, combined with its benefits in transparency, immutability, and security, makes it a revolutionary tool for applications ranging from finance to supply chains and healthcare. However, our analysis has also highlighted significant challenges, particularly concerning security vulnerabilities and attacks that threaten the integrity and trustworthiness of blockchain networks.

The severity and recurrence of these security threats are evident in real-world examples, such as 51% attacks, double-spending attacks, and smart contract vulnerabilities. These attacks have led to substantial financial losses and have eroded public confidence in blockchain systems. The increasing prevalence of reported vulnerabilities and the rise in cybercrime further emphasize the urgency of addressing these challenges through robust and adaptive security measures.

Through our research questions, we identified and categorized common blockchain attacks, evaluated their impact on security and system integrity, and assessed the efficacy of existing detection and mitigation strategies. Our findings revealed that smart contract vulnerabilities and double-spending attacks represent the most significant threats due to their prevalence and the potential for large-scale financial damage. Moreover, the need for tailored solutions is evident as blockchain applications across industries exhibit varying security requirements and threat landscapes.

This research underscores the importance of employing customized security measures tailored to the unique operational needs of blockchain applications. For instance, financial

systems require enhanced consensus mechanisms and cryptographic techniques to prevent double spending, while supply chain networks benefit from robust node reputation systems and secure routing protocols to maintain data integrity. By carefully analyzing the unique characteristics and threats faced by blockchain systems, developers and practitioners can implement effective, context-specific security measures.

Our study contributes to the body of knowledge on blockchain security by offering a detailed examination of attack mitigation techniques and their practical applications. It equips researchers and practitioners with actionable insights to help them make informed decisions and adopt robust security strategies. Furthermore, it highlights the limitations of current security mechanisms, encouraging the exploration of cutting-edge technologies to address emerging threats. A notable limitation of this study is the lack of an in-depth discussion about the mitigation or prevention techniques for the attacks, although some studies have in-depth discussions on mitigating or preventing those attacks, which we want to address in future research.

Looking ahead, we recognize that the dynamic nature of blockchain technology necessitates ongoing research and collaboration to address new vulnerabilities and attack vectors. Emerging technologies, such as quantum-resistant cryptography and artificial-intelligence-based anomaly detection, hold promise for further enhancing the security and resilience of blockchain systems. By prioritizing security and adopting proactive measures, we can ensure the trust in and integrity of blockchain technology, paving the way for its widespread adoption and transformative impact.

Author Contributions: M.K.S. designed the systematic search strategy, performed keyword analysis, conducted the literature review, contributed to the results and discussion, created the visualizations, and participated in manuscript refinement; B.S. conceived the initial idea for the paper and contributed to the development of the introduction and methodology; M.M.H. assisted in the preparation of the results and contributed to the discussion; M.J.H.F. coordinated the overall research effort and contributed to the abstract, title, introduction, and conclusion; N.A., S.T. and A.S. participated in the analysis of the included literature; H.S. provided academic supervision, critical feedback, and guidance, led scholarly discussions, and finalized the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Chidambaranathan, S.; Geetha, R. Deep learning enabled blockchain based electronic heathcare data attack detection for smart health systems. *Meas. Sens.* **2024**, *31*, 100959. [CrossRef]
2. Srivastava, A.; Gupta, J. Attack resistant blockchain-based healthcare record system using modified RSA Algorithm. *Int. J. Inf. Technol.* **2024**, *16*, 417–424. [CrossRef]
3. Deloitte's 2021 Global Blockchain Survey: Financial Leaders See Digital Assets as the Future. 2021. Available online: https://www2.deloitte.com/us/en/insights/topics/understanding-blockchain-potential/global-blockchain-survey.html (accessed on 25 March 2025).
4. FinancialNewsMedia.com. Global Blockchain Technology Market Expected to Reach $1.4 Trillion by 2030. Available online: https://www.prnewswire.com/news-releases/global-blockchain-technology-market-expected-to-reach-1-4-trillion-by-2030-301535849.html (accessed on 20 April 2024).
5. Pramod, B.; Vaibhav, M. *Blockchain Distributed Ledger Market Expected to Reach $ 137.29 Billion by 2027*; Allied Market Research: Wilmington, DE, USA, 2021.
6. Sood, A.; Simon, R. Implementation of Blockchain in Cross Border Money Transfer. In Proceedings of the 2019 4th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 21–22 November 2019; pp. 104–107. [CrossRef]

7. Wang, S.; Ouyang, L.; Yuan, Y.; Ni, X.; Han, X.; Wang, F.Y. Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends. *IEEE Trans. Syst. Man. Cybern. Syst.* **2019**, *49*, 2266–2277. [CrossRef]
8. Panarello, A.; Tapas, N.; Merlino, G.; Longo, F.; Puliafito, A. Blockchain and IoT Integration: A Systematic Survey. *Sensors* **2018**, *18*, 2575. [CrossRef] [PubMed]
9. Saha, B.; Islam, M.S.; Riad, A.K.; Tahora, S.; Shahriar, H.; Sneha, S. BlockTheFall: Wearable Device-based Fall Detection Framework Powered by Machine Learning and Blockchain for Elderly Care. *arXiv* **2023**, arXiv:2306.06452.
10. Tahora, S.; Saha, B.; Sakib, N.; Shahriar, H.; Haddad, H. Blockchain Technology in Higher Education Ecosystem: Unraveling the Good, Bad, and Ugly. *arXiv* **2023**, arXiv:2306.04071.
11. Shrier, D.; Wu, W.; Pentland, A. Blockchain & infrastructure (identity, data security). *Mass. Inst. Technol.-Connect. Sci.* **2016**, *1*, 1–19.
12. Hölbl, M.; Kompara, M.; Kamišalić, A.; Nemec Zlatolas, L. A systematic review of the use of blockchain in healthcare. *Symmetry* **2018**, *10*, 470. [CrossRef]
13. Randolph, J.; Faruk, M.J.H.; Saha, B.; Shahriar, H.; Valero, M.; Zhao, L.; Sakib, N. Blockchain-based Medical Image Sharing and Automated Critical-results Notification: A Novel Framework. In Proceedings of the 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), Los Alamitos, CA, USA, June 27–1 July 2022; pp. 1756–1761.
14. Faruk, M.J.H.; Shahriar, H.; Saha, B.; Barek, A. Security in Electronic Health Records System: Blockchain-Based Framework to Protect Data Integrity. In *Blockchain for Cybersecurity in Cyber-Physical Systems*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 125–137.
15. Perboli, G.; Musso, S.; Rosano, M. Blockchain in Logistics and Supply Chain: A Lean Approach for Designing Real-World Use Cases. *IEEE Access* **2018**, *6*, 62018–62028. [CrossRef]
16. Madine, M.; Salah, K.; Jayaraman, R.; Battah, A.; Hasan, H.; Yaqoob, I. Blockchain and NFTs for Time-Bound Access and Monetization of Private Data. *IEEE Access* **2022**, *10*, 94186–94202. [CrossRef]
17. O'Reilly, P.; Rigopoulos, K. *Fiscal Year 2023 Cybersecurity and Privacy Annual Report*; Number NIST SP 800-229; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2024; p. NIST SP 800-229. [CrossRef]
18. Kramer, S. Unpacking SonicWall 2024 Mid-Year Cyber Threat Report. 2024. Available online: https://thecuberesearch.com/unpacking-sonicwall-2024-midyear-cyber-threat-report/ (accessed on 20 April 2024).
19. Badertscher, C.; Lu, Y.; Zikas, V. A rational protocol treatment of 51% attacks. In Proceedings of the Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, 16–20 August 2021; Proceedings, Part III 41; Springer: Berlin/Heidelberg, Germany,2021; pp. 3–32.
20. Sinclair, S. Ethereum Classic's MESS Solution Won't Provide 'Robust' Security Against 51% Attacks. Available online: https://www.coindesk.com/markets/2020/08/20/ethereum-classics-mess-solution-wont-provide-robust-security-against-51-attacks/ (accessed on 20 April 2024).
21. Kustov, V.; Aleksey, G.; Nikolay, B.; Ekaterina, S.; Ravi, R.V. Three Sources of Blockchain Technology Vulnerabilities—How to Deal with them? In Proceedings of the 2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India, 8 September 2022; pp. 1–8. [CrossRef]
22. Livni, E. Binance Blockchain Hit by $570 Million Hack, Exposing Crypto Vulnerabilities. *The New York Times*, 7 October 2022.
23. Oxford Analytica. Binance breach underlines risks for crypto ecosystem. *Emerald Expert Briefings* **2022**. [CrossRef]
24. Beck, R.H. Top 10 Cryptocurrencies of March 4, 2025. Available online: https://www.investopedia.com/top-10-cryptocurrencies-5189931 (accessed on 20 April 2024).
25. Li, X.; Cheng, J.; Shi, Z.; Liu, J.; Zhang, B.; Xu, X.; Tang, X.; Sheng, V. Blockchain Security Threats and Collaborative Defense: A Literature Review. *Comput. Mater. Contin.* **2023**, *76*, 2597–2629. [CrossRef]
26. Hussain, A.H.; Hasan, M.N.; Prince, N.U.; Islam, M.M.; Islam, S.; Hasan, S.K. Enhancing cyber security using quantum computing and Artificial Intelligence: A review. *World J. Adv. Res. Rev.* **2021**, *10*, 448–456. [CrossRef]
27. Mollajafari, S.; Bechkoum, K. Blockchain Technology and Related Security Risks: Towards a Seven-Layer Perspective and Taxonomy. *Sustainability* **2023**, *15*, 13401. [CrossRef]
28. Karakostas, D.; Kiayias, A.; Zacharias, T. Blockchain bribing attacks and the efficacy of counterincentives. In Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, Salt Lake City, UT, USA, 14–18 October 2024; pp. 1031–1045.
29. Madhushanie, N.; Vidanagamachchi, S.; Arachchilage, N. Selfish mining attack in blockchain: A systematic literature review. *Int. J. Inf. Secur.* **2024**, *23*, 2333–2351. [CrossRef]
30. CASP Checklist for Systematic Reviews and Meta-Analysis. Available online: https://casp-uk.net/casp-tools-checklists/systematic-reviews-meta-analysis-observational-studies/ (accessed on 25 March 2025).

31. Careem, M.A.A.; Dutta, A. Reputation based Routing in MANET using Blockchain. In Proceedings of the 2020 International Conference on Communication Systems & Networks (COMSNETS), Bengaluru, India, 7–11 January 2020; pp. 1–6.
32. Luu, L.; Chu, D.H.; Olickel, H.; Saxena, P.; Hobor, A. Making Smart Contracts Smarter. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS), Vienna, Austria, 24–28 October 2016; pp. 254–269.
33. Peng, K.; Li, M.; Huang, H.; Wang, C.; Wan, S.; Choo, K.K.R. Security challenges and opportunities for smart contracts in Internet of Things: A survey. *IEEE Internet Things J.* **2021**, *8*, 12004–12020. [CrossRef]
34. Sayeed, S.; Marco-Gisbert, H.; Caira, T. Smart contract: Attacks and protections. *IEEE Access* **2020**, *8*, 24416–24427. [CrossRef]
35. Mense, A.; Flatscher, M. Security Vulnerabilities in Ethereum Smart Contracts. In Proceedings of the 20th International Conference on Information Integration and Web-Based Applications & Services (iiWAS), Yogyakarta, Indonesia, 19–21 November 2018; pp. 375–380.
36. Begum, A.; Tareq, A.H.; Sultana, M.; Sohel, M.K.; Rahman, T.; Sarwar, A.H. Blockchain Attacks, Analysis and a Model to Solve Double Spending Attack. *Int. J. Mach. Learn. Comput.* **2020**, *10*, 352–357.
37. Marcus, Y.; Heilman, E.; Goldberg, S. Low-Resource Eclipse Attacks on Ethereum's Peer-to-Peer Network. *Cryptology ePrint Archive* **2018**, *2018*, 857. Available online: https://eprint.iacr.org/2018/857 (accessed on 20 April 2024).
38. Deirmentzoglou, E.; Papakyriakopoulos, G.; Patsakis, C. A survey on long-range attacks for proof of stake protocols. *IEEE Access* **2019**, *7*, 28712–28725. [CrossRef]
39. Grundmann, M.; Neudecker, T.; Hartenstein, H. Exploiting transaction accumulation and double spends for topology inference in bitcoin. In Proceedings of the Financial Cryptography and Data Security: FC 2018 International Workshops, BITCOIN, VOTING, and WTSC, Nieuwpoort, Curaçao, 2 March 2018, Revised Selected Papers 22; Springer: Berlin/Heidelberg, Germany, 2019; pp. 113–126.
40. Riadi, I.; Umar, R.; Busthomi, I.; Muhammad, A.W. Block-hash of blockchain framework against man-in-the-middle attacks. *Regist. J. Ilm. Teknol. Sist. Inf.* **2022**, *8*, 1–9. [CrossRef]
41. Razmjouei, P.; Kavousi-Fard, A.; Dabbaghjamanesh, M.; Jin, T.; Su, W. Ultra-lightweight mutual authentication in the vehicle based on smart contract blockchain: Case of MITM attack. *IEEE Sens. J.* **2020**, *21*, 15839–15848. [CrossRef]
42. Akter, S.; Chellappan, S.; Chakraborty, T.; Khan, T.A.; Rahman, A.; Al Islam, A.A. Man-in-the-middle attack on contactless payment over NFC communications: Design, implementation, experiments and detection. *IEEE Trans. Dependable Secur. Comput.* **2020**, *18*, 3012–3023. [CrossRef]
43. Aggarwal, S.; Kumar, N. Attacks on blockchain. In *Advances in Computers*; Elsevier: Amsterdam, The Netherlands, 2021; Volume 121, pp. 399–410.
44. The Defiant Weekly. Saddle Finance Loses More than $10 Million in a Recent DeFi Exploit. Available online: https://thedefiant.io/saddle-finance-loses-10-million (accessed on 20 April 2024).
45. CoinDesk. Japan's Liquid Global Exchange Hacked; $90M in Crypto Siphoned Off. 2021. Available online: https://www.coindesk.com/markets/2021/08/19/japans-liquid-global-exchange-hacked-90m-in-crypto-siphoned-off/ (accessed on 6 April 2023).
46. Shemov, G.; Garcia de Soto, B.; Alkhzaimi, H. Blockchain applied to the construction supply chain: A case study with threat model. *Front. Eng. Manag.* **2020**, *7*, 564–577. [CrossRef]
47. Averin, A.; Averina, O. Review of blockchain technology vulnerabilities and blockchain-system attacks. In Proceedings of the 2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), Vladivostok, Russia, 1–4 October 2019; pp. 1–6.
48. König, L.; Unger, S.; Kieseberg, P.; Tjoa, S.; Blockchains, J.R.C. The Risks of the Blockchain A Review on Current Vulnerabilities and Attacks. *J. Internet Serv. Inf. Secur.* **2020**, *10*, 110–127.
49. Rathod, N.; Motwani, D. Security threats on blockchain and its countermeasures. *Int. Res. J. Eng. Technol.* **2018**, *5*, 1636–1642.
50. Castro, M.; Druschel, P.; Ganesh, A.; Rowstron, A.; Wallach, D.S. Secure routing for structured peer-to-peer overlay networks. *ACM SIGOPS Oper. Syst. Rev.* **2002**, *36*, 299–314. [CrossRef]
51. Duan, L.; Sun, Y.; Zhang, K.; Ding, Y. Multiple-Layer Security Threats on the Ethereum Blockchain and Their Countermeasures. *Secur. Commun. Networks* **2022**, *2022*, 5307697. [CrossRef]
52. NFTing. What is a Replay Attack? Available online: https://nfting.store/blogs/news/what-is-a-replay-attack (accessed on 20 April 2024).
53. Aponte-Novoa, F.A.; Orozco, A.L.S.; Villanueva-Polanco, R.; Wightman, P. The 51% attack on blockchains: A mining behavior study. *IEEE Access* **2021**, *9*, 140549–140564. [CrossRef]
54. Shanaev, S.; Shuraeva, A.; Vasenin, M.; Kuznetsov, M. Cryptocurrency value and 51% attacks: Evidence from event studies. *J. Altern. Investments* **2019**, *22*, 65–77. [CrossRef]
55. Saad, M.; Spaulding, J.; Njilla, L.; Kamhoua, C.A.; Nyang, D.; Mohaisen, A. Overview of attack surfaces in blockchain. In *Blockchain for Distributed Systems Security*; John Wiley & Sons: Hoboken, NJ, USA, 2019; pp. 51–66.

56. Praitheeshan, P.; Pan, L.; Yu, J.; Liu, J.; Doss, R. Security analysis methods on ethereum smart contract vulnerabilities: A survey. *arXiv* **2019**, arXiv:1908.08605.
57. Vivar, A.L.; Castedo, A.T.; Orozco, A.L.S.; Villalba, L.J.G. An Analysis of Smart Contracts Security Threats Alongside Existing Solutions. *Entropy* **2020**, *22*, 203. [CrossRef] [PubMed]
58. Sonnino, A.; Bano, S.; Al-Bassam, M.; Danezis, G. Replay Attacks and Defenses Against Cross-Shard Consensus in Sharded Distributed Ledgers. In Proceedings of the 2020 IEEE European Symposium on Security and Privacy (EuroS&P), Genoa, Italy, 7–11 September 2020; pp. 294–308.
59. Zhang, S.; Lee, J.H. Double-Spending with a Sybil Attack in the Bitcoin Decentralized Network. *IEEE Trans. Ind. Inform.* **2019**, *15*, 5715–5722. [CrossRef]
60. Apostolaki, M.; Zohar, A.; Vanbever, L. Hijacking Bitcoin: Routing Attacks on Cryptocurrencies. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–24 May 2017; pp. 375–392. [CrossRef]
61. Otte, P.; de Vos, M.; Pouwelse, J. TrustChain: A Sybil-Resistant Scalable Blockchain. *Future Gener. Comput. Syst.* **2020**, *107*, 770–780. [CrossRef]
62. Pradhan, S.; Tripathy, S.; Nandi, S. Blockchain-Based Security Framework for P2P File Sharing System. In Proceedings of the 2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Indore, India, 16–19 December 2018; pp. 1–6. [CrossRef]
63. Rajab, T.; Manshaei, M.H.; Dakhilalian, M.; Jadliwala, M.; Rahman, M.A. On the Feasibility of Sybil Attacks in Shard-Based Permissionless Blockchains. *arXiv* **2020**, arXiv:2002.06531. Available online: https://arxiv.org/abs/2002.06531 (accessed on 20 April 2024). [CrossRef]
64. Morganti, G.; Schiavone, E.; Bondavalli, A. Risk Assessment of Blockchain Technology. In Proceedings of the 2018 Eighth Latin-American Symposium on Dependable Computing (LADC), Foz do Iguaçu, Brazil, 24–26 October 2018; pp. 87–96.
65. Conti, M.; Kumar, E.S.; Lal, C.; Ruj, S. A survey on security and privacy issues of bitcoin. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3416–3452. [CrossRef]
66. Alangot, B.; Reijsbergen, D.; Venugopalan, S.; Szalachowski, P.; Yeo, K.S. Decentralized and lightweight approach to detect eclipse attacks on proof of work blockchains. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 1659–1672. [CrossRef]
67. Xu, G.; Guo, B.; Su, C.; Zheng, X.; Liang, K.; Wong, D.S.; Wang, H. Am I eclipsed? A smart detector of eclipse attacks for Ethereum. *Comput. Secur.* **2020**, *88*, 101604. [CrossRef]
68. Ghaffari, A.; Jelodari, N.; Pouralish, S.; Derakhshanfard, N.; Arasteh, B. Securing internet of things using machine and deep learning methods: A survey. *Clust. Comput.* **2024**, *27*, 9065–9089. [CrossRef]
69. Yan, K.; Zhang, X.; Diao, W. Stealing Trust: Unraveling Blind Message Attacks in Web3 Authentication. In Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS), Salt Lake City, UT, USA, 9–13 December 2024; pp. 555–569. [CrossRef]
70. Papaspirou, V.; Kantzavelou, I.; Yigit, Y.; Maglaras, L.; Katsikas, S. A Blockchain-Based Multi-Factor Honeytoken Dynamic Authentication Mechanism. In Proceedings of the 19th International Conference on Availability, Reliability and Security (ARES), Vienna, Austria, 29–31 July 2024; pp. 1–9. [CrossRef]
71. Samreen, N.F.; Alalfi, M.H. A survey of security vulnerabilities in ethereum smart contracts. *arXiv* **2021**, arXiv:2105.06974.
72. Xu, J.; Dang, F.; Ding, X.; Zhou, M. A Survey on Vulnerability Detection Tools of Smart Contract Bytecode. In Proceedings of the 2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE), Dalian, China, 27–29 September 2020; pp. 94–98. [CrossRef]
73. He, D.; Wu, R.; Li, X.; Chan, S.; Guizani, M. Detection of Vulnerabilities of Blockchain Smart Contracts. *IEEE Internet Things J.* **2023**, *10*, 12178–12185. [CrossRef]
74. Choi, J.; Ahn, B.; Bere, G.; Ahmad, S.; Mantooth, H.A.; Kim, T. Blockchain-Based Man-in-the-Middle (MITM) Attack Detection for Photovoltaic Systems. In Proceedings of the 2021 IEEE Design Methodologies Conference (DMC), Virtual Event, 16–17 August 2021; pp. 1–6.
75. Wazid, M.; Bera, B.; Mitra, A.; Das, A.K.; Ali, R. Private Blockchain-Envisioned Security Framework for AI-Enabled IoT-Based Drone-Aided Healthcare Services. In Proceedings of the 2nd ACM MobiCom Workshop on Drone Assisted Wireless Communications for 5G and Beyond (DroneCom), London, UK, 25 September 2020; pp. 37–42.
76. Kulkarni, O. Preventing the Man-in-the-Middle Attack on Internet Communication Using Blockchain Technology. Ph.D. Thesis, National College of Ireland, Dublin, Ireland, 2019.
77. Momeni, H.; Sadoogi, N.; Farrokhifar, M.; Gharibeh, H.F. Fault diagnosis in photovoltaic arrays using GBSSL method and proposing a fault correction system. *IEEE Trans. Ind. Inform.* **2019**, *16*, 5300–5308. [CrossRef]
78. Homoliak, I.; Venugopalan, S.; Hum, Q.; Szalachowski, P. A Security Reference Architecture for Blockchains. In Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 14–17 July 2019; pp. 390–397.

79. Kebande, V.R.; Awaysheh, F.M.; Ikuesan, R.A.; Alawadi, S.A.; Alshehri, M.D. A blockchain-based multi-factor authentication model for a cloud-enabled internet of vehicles. *Sensors* **2021**, *21*, 6018. [CrossRef] [PubMed]
80. Alkaeed, M.; Soliman, M.M.; Khan, K.M.; Elfouly, T.M. Distributed Framework via Blockchain Smart Contracts for Smart Grid Systems Against Cyber-Attacks. In Proceedings of the 2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC), Johor Bahru, Malaysia, 8 August 2020; pp. 100–105.
81. Jurcut, A.; Niculcea, T.; Ranaweera, P.; Le-Khac, N.A. Security considerations for Internet of Things: A survey. *SN Comput. Sci.* **2020**, *1*, 193. [CrossRef]
82. Abdallah, R.; Besancenot, J.; Bertelle, C.; Duvallet, C.; Gilletta, F. An Extensive Preliminary Blockchain Survey from a Maritime Perspective. *Smart Cities* **2023**, *6*, 846–877. [CrossRef]
83. Frankenfield, J. *51% Attack*; Investopedia: New York, NY, USA, 2019; Volume 7.
84. Ye, C.; Li, G.; Cai, H.; Gu, Y.; Fukuda, A. Analysis of Security in Blockchain: Case Study in 51% Attack Detecting. In Proceedings of the 2018 5th International Conference on Dependable Systems and Their Applications (DSA), Guangzhou, China, 18–21 December 2018; pp. 15–24.
85. Bastiaan, M. Preventing the 51% Attack: A Stochastic Analysis of Two Phase Proof of Work in Bitcoin. Bachelor's Thesis, University of Twente, Enschede, The Netherlands, 2015. Available online: https://essay.utwente.nl/74730/ (accessed on 20 April 2024).
86. Lee, H.; Shin, M.; Kim, K.S.; Kang, Y.; Kim, J. *Recipient-Oriented Transaction for Preventing Double Spending Attacks in Private Blockchain*; IEEE: Hong Kong, China, 2018.
87. Dasgupta, D.; Shrein, J.M.; Gupta, K.D. A survey of blockchain from security perspective. *J. Bank. Financ. Technol.* **2019**, *3*, 1–17. [CrossRef]
88. Sahay, R.; Geethakumari, G.; Mitra, B. A novel blockchain based framework to secure IoT-LLNs against routing attacks. *Computing* **2020**, *102*, 2445–2470. [CrossRef]
89. Tran, M.; Choi, I.; Moon, G.J.; Vu, A.V.; Kang, M.S. A Stealthier Partitioning Attack Against Bitcoin Peer-to-Peer Network. In Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 18–21 May 2020; pp. 894–909.
90. Tekiner, E.; Acar, A.; Uluagac, A.S.; Kirda, E.; Selcuk, A.A. SoK: Cryptojacking Malware. In Proceedings of the 2021 IEEE European Symposium on Security and Privacy (EuroS&P), Vienna, Austria, 6–10 September 2021; pp. 120–139.
91. Arisdakessian, S.; Wahab, O.A.; Mourad, A.; Otrok, H.; Guizani, M. A survey on iot intrusion detection: Federated learning, game theory, social psychology and explainable ai as future directions. *IEEE Internet Things J.* **2022**, *10*, 4059–4092. [CrossRef]
92. ABBASSI, Y.; Benlahmer, H. BCSDN-IoT: Towards an IoT security architecture based on SDN and Blockchain. *Int. J. Electr. Comput. Eng. Syst.* **2022**, *13*, 155–163. [CrossRef]
93. Mastilak, L.; Helebrandt, P.; Galinski, M.; Kotuliak, I. Secure Inter-Domain Routing Based on Blockchain: A Comprehensive Survey. *Sensors* **2022**, *22*, 1437. [CrossRef]
94. Sayeed, S.; Marco-Gisbert, H. Assessing Blockchain Consensus and Security Mechanisms against the 51% Attack. *Appl. Sci.* **2019**, *9*, 1788. [CrossRef]
95. Ihle, C.; Trautwein, D.; Schubotz, M.; Meuschke, N.; Gipp, B. Incentive Mechanisms in Peer-to-Peer Networks—A Systematic Literature Review. *ACM Comput. Surv.* **2023**, *55*, 308. [CrossRef]
96. de Oliveira, M.T.; Reis, L.H.; Medeiros, D.S.; Carrano, R.C.; Olabarriaga, S.D.; Mattos, D.M. Blockchain reputation-based consensus: A scalable and resilient mechanism for distributed mistrusting applications. *Comput. Netw.* **2020**, *179*, 107367. [CrossRef]
97. Alsunbul, A.; Elmedany, W.; Al-Ammal, H. Blockchain Application in Healthcare Industry: Attacks and Countermeasures. In Proceedings of the 2021 International Conference on Data Analytics for Business and Industry (ICDABI), Sakheer, Bahrain, 6–7 December 2021; pp. 621–629.
98. Anita, N.; Vijayalakshmi, M. Blockchain Security Attack: A Brief Survey. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; pp. 1–6.
99. Rottondi, C.; Panzeri, A.; Yagne, C.T.; Verticale, G. Detection and mitigation of the eclipse attack in chord overlays. *Int. J. Comput. Sci. Eng.* **2016**, *13*, 111–121. [CrossRef]
100. Ekparinya, P.; Gramoli, V.; Jourjon, G. Double-Spending Risk Quantification in Private, Consortium and Public Ethereum Blockchains. *arXiv* **2018**, arXiv:1805.05004.
101. Quintyne-Collins, M. Short Paper: Towards Characterizing Sybil Attacks in Cryptocurrency Mixers. Cryptology ePrint Archive, Paper 2019/1111. 2019. Available online: https://eprint.iacr.org/2019/1111 (accessed on 25 March 2025).
102. Swathi, P.; Modi, C.; Patel, D. Preventing Sybil Attack in Blockchain Using Distributed Behavior Monitoring of Miners. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; pp. 1–6. [CrossRef]

103. Boi, B.; Esposito, C.; Lee, S. Smart Contract Vulnerability Detection: The Role of Large Language Model (LLM). *ACM SIGAPP Appl. Comput. Rev.* **2024**, *24*, 19–29. [CrossRef]
104. Chen, C.; Su, J.; Chen, J.; Wang, Y.; Bi, T.; Yu, J.; Wang, Y.; Lin, X.; Chen, T.; Zheng, Z. When ChatGPT Meets Smart Contract Vulnerability Detection: How Far Are We? *ACM Trans. Softw. Eng. Methodol.* **2024**, *33*, 17. [CrossRef]
105. Guo, H.; Chen, Y.; Chen, X.; Huang, Y.; Zheng, Z. Smart Contract Code Repair Recommendation based on Reinforcement Learning and Multi-metric Optimization. *ACM Trans. Softw. Eng. Methodol.* **2024**, *33*, 106. [CrossRef]
106. Rabieinejad, E.; Yazdinejad, A.; Parizi, R.M. A Deep Learning Model for Threat Hunting in Ethereum Blockchain. In Proceedings of the 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Shenyang, China, 22–25 October 2021; pp. 1185–1190. [CrossRef]
107. Wu, Z.; Wu, J.; Zhang, H.; Li, Z.; Chen, J.; Zheng, Z.; Xia, Q.; Fan, G.; Zhen, Y. DAppFL: Just-in-Time Fault Localization for Decentralized Applications in Web3. In Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA), Vienna, Austria, 15–19 July 2024; pp. 137–148. [CrossRef]
108. Atiewi, S.; Al-Rahayfeh, A.; Almiani, M.; Abuhussein, A.; Yussof, S. Ethereum blockchain-based three factor authentication and multi-contract access control for secure smart home environment in 5G networks. *Clust. Comput.* **2024**, *27*, 4551–4568. [CrossRef]
109. Brahmam, M.G.; R, V.A. ICSMPC: Design of an Iterative-Learning Contextual Side Chaining Model for Improving Security of Priority-Aware Cloud Resources. *J. Grid Comput.* **2023**, *21*, 55. [CrossRef]
110. Das, D.; Banerjee, S.; Biswas, U. A secure vehicle theft detection framework using Blockchain and smart contract. *Peer- Netw. Appl.* **2021**, *14*, 672–686. [CrossRef]
111. Heikamp, F.; Pan, L.; Doss, R.; Trujillo-Rasua, R.; Ruj, S. ForTrac: A Secure NFT-Based Forward Traceability System for Providing Data Accuracy and Completeness. In Proceedings of the 5th ACM International Symposium on Blockchain and Secure Critical Infrastructure (BSCI), Melbourne, VIC, Australia, 10–14 July 2023; pp. 1–10. [CrossRef]
112. Huang, Q.; Liao, D.; Xing, Z.; Zuo, Z.; Wang, C.; Xia, X. Semantic-Enriched Code Knowledge Graph to Reveal Unknowns in Smart Contract Code Reuse. *ACM Trans. Softw. Eng. Methodol.* **2023**, *32*, 147. [CrossRef]
113. Ilyas, B.; Kumar, A.; Ali, S.M.; Lei, H. Blockchain-enabled IoT access control model for sharing electronic healthcare data. *Multimed. Tools Appl.* **2024**, *84*, 8127–8148. [CrossRef]
114. Liu, B.; Sun, S.; Szalachowski, P. SMACS: Smart Contract Access Control Service. In Proceedings of the 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Valencia, Spain, 29 June–2 July 2020; pp. 221–232. [CrossRef]
115. Si, J.J.; Sharma, T.; Wang, K.Y. Understanding User-Perceived Security Risks and Mitigation Strategies in the Web3 Ecosystem. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), Honolulu, HI, USA, 11–16 May 2024; pp. 1–22. [CrossRef]
116. Wang, K.; Ling, Y.; Zhang, Y.; Yu, Z.; Wang, H.; Bai, G.; Ooi, B.C.; Dong, J.S. Characterizing Cryptocurrency-themed Malicious Browser Extensions. *Proc. ACM Meas. Anal. Comput. Syst.* **2022**, *6*, 43. [CrossRef]
117. Al-Kafi, G.M.A.; Ali, G.; Faiza, J.T.; Pal, K.R.; Reno, S. Correction to: SHBF: A Secure and Scalable Hybrid Blockchain Framework for Resolving Trilemma Challenges. *Int. J. Inf. Technol.* **2024**, *16*, 269–270. [CrossRef]
118. Vivekanandan, G.; Divya, B.; Madhav, V.T.; Naveen, K.; Karthick, T.K.; Yuvanesh, P. VoteChain: Promising a Secure and Transparent Election Using Blockchain and Biometrics. In Proceedings of the 2024 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), Chennai, India, 17–18 October 2024; pp. 1–6. [CrossRef]
119. Govindaram, A.; A, J. FLBC-IDS: A Federated Learning and Blockchain-Based Intrusion Detection System for Secure IoT Environments. *Multimed. Tools Appl.* **2024**, *83*, Article 19777. [CrossRef]
120. Vishwakarma, L.; Das, D. BlockTree: A nonlinear structured, scalable and distributed ledger scheme for processing digital transactions. *Clust. Comput.* **2021**, *24*, 3751–3765. [CrossRef]
121. Yan, K.; Zhang, J.; Liu, X.; Diao, W.; Guo, S. Bad Apples: Understanding the Centralized Security Risks in Decentralized Ecosystems. In Proceedings of the ACM Web Conference 2023 (WWW '23), Austin, TX, USA, 30 April–4 May 2023; pp. 2274–2283. [CrossRef]
122. Niu, J.; Peng, W.; Zhang, X.; Zhang, Y. NARRATOR: Secure and Practical State Continuity for Trusted Execution in the Cloud. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS), Los Angeles, CA, USA, 7–11 November 2022; pp. 2385–2399. [CrossRef]
123. Vidal, F.R.; Ivaki, N.; Laranjeiro, N. OpenSCV: An open hierarchical taxonomy for smart contract vulnerabilities. *Empir. Softw. Eng.* **2024**, *29*, 101. [CrossRef]
124. Chen, J.; Chen, C.; Hu, J.; Grundy, J.; Wang, Y.; Chen, T.; Zheng, Z. Identifying Smart Contract Security Issues in Code Snippets from Stack Overflow. In Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA), Vienna, Austria, 15–19 July 2024; pp. 1198–1210. [CrossRef]

125. Weiss, K.; Ferreira Torres, C.; Wendland, F. Analyzing the Impact of Copying-and-Pasting Vulnerable Solidity Code Snippets from Question-and-Answer Websites. In Proceedings of the 2024 ACM on Internet Measurement Conference (IMC), Madrid, Spain, 21–23 October 2024; pp. 713–730. [CrossRef]
126. Akhras, R.; Hajj, W.E.; Hajj, H.; Shaban, K.; Jaber, R. ECC: Enhancing Smart Grid Communication with Ethereum Blockchain, Asymmetric Cryptography, and Cloud Services. In Proceedings of the 2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA), Thessaloniki, Greece, 9–11 October 2023; pp. 1–10. [CrossRef]
127. Eshghie, M.; Artho, C.; Stammler, H.; Ahrendt, W.; Hildebrandt, T.; Schneider, G. HighGuard: Cross-Chain Business Logic Monitoring of Smart Contracts. In Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering (ASE), Sacramento, CA, USA, 16–20 September 2024; pp. 2378–2381. [CrossRef]
128. Haugum, T.; Hoff, B.; Alsadi, M.; Li, J. Security and Privacy Challenges in Blockchain Interoperability—A Multivocal Literature Review. In Proceedings of the International Conference on Evaluation and Assessment in Software Engineering (EASE), Gothenburg, Sweden, 13–15 June 2022; pp. 347–356. [CrossRef]
129. Tran, T.D.; Vo, K.A.; Phan, D.T.; Tan, C.N.; Pham, V.H. ChainSniper: A Machine Learning Approach for Auditing Cross-Chain Smart Contracts. In Proceedings of the 2024 9th International Conference on Intelligent Information Technology (ICIIT), Ho Chi Minh City, Vietnam, 26–28 January 2024; pp. 223–230. [CrossRef]
130. Zheng, P.; Luo, X.; Zheng, Z. BSHUNTER: Detecting and Tracing Defects of Bitcoin Scripts. In Proceedings of the 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), Melbourne, Australia, 14–20 May 2023; pp. 307–318. [CrossRef]
131. Awasthi, C.; Mishra, P.K.; Pal, P.K.; Khan, S.B.; Agarwal, A.K.; Gadekallu, T.R.; Malibari, A.A. Preservation of Sensitive Data Using Multi-Level Blockchain-Based Secured Framework for Edge Network Devices. *J. Grid Comput.* **2023**, *21*, 69. [CrossRef]
132. Fischlin, M.; Günther, F. Verifiable Verification in Cryptographic Protocols. In Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS), Copenhagen, Denmark, 26–30 November 2023; pp. 3239–3253. [CrossRef]
133. Liu, A.; Chen, X.b.; Xu, G.; Wang, Z.; Sun, Y.; Wang, Y.; Feng, H. QBIoV: A secure data sharing scheme for the Internet of vehicles based on quantum-enabled blockchain. *Quantum Inf. Process.* **2024**, *23*, 225. [CrossRef]
134. Pedrosa, M.; Zúquete, A.; Costa, C. RAIAP: Renewable authentication on isolated anonymous profiles: A GDPR compliant self-sovereign architecture for distributed systems. *Peer-to-Peer Netw. Appl.* **2020**, *13*, 1577–1599. [CrossRef]
135. Ramya, R.R.; Banumathi, J. An optimized approach with 128-bit key management for IoT-enabled smart grid: Enhancing efficiency, security, and sustainability. *Electr. Eng.* **2025**, *107*, 2207–2225. [CrossRef]
136. Reddy, N.M.; Budati, A.K.; Islam, S.; Ramesh, G. Enhanced Elliptic Curve-Diffie Hellman Technique with Bigdata Analytics for Satellite Image Security Enhancement in Internet of Things Systems. *Earth Sci. Inform.* **2024**, *17*, 711–723. [CrossRef]
137. Wu, G.; Wang, H.; Yang, Z.; He, D.; Chan, S. Electronic Health Records Sharing Based on Consortium Blockchain. *J. Med. Syst.* **2024**, *48*, 106. [CrossRef]
138. Antonopoulos, P.; Kaushik, R.; Kodavalla, H.; Rosales Aceves, S.; Wong, R.; Anderson, J.; Szymaszek, J. SQL Ledger: Cryptographically Verifiable Data in Azure SQL Database. In Proceedings of the 2021 International Conference on Management of Data (SIGMOD), Xi'an, China, 20–25 June 2021; pp. 2437–2449. [CrossRef]
139. Shahaab, A.; Hewage, C.; Khan, I. Preventing Spoliation of Evidence with Blockchain: A Perspective from South Asia. In Proceedings of the 2021 3rd International Conference on Blockchain Technology (ICBCT), Shanghai, China, 26–28 March 2021; pp. 45–52. [CrossRef]
140. Sharma, A.; Kaur, P. Tamper-proof multitenant data storage using blockchain. *Peer-to-Peer Netw. Appl.* **2023**, *16*, 431–449. [CrossRef]
141. Aldaej, A.; Ahanger, T.A.; Ullah, I. Deep neural network-based secure healthcare framework. *Neural Comput. Appl.* **2024**, *36*, 17467–17482. [CrossRef]
142. Li, X.; Wang, Z.; Leung, V.C.M.; Ji, H.; Liu, Y.; Zhang, H. Blockchain-empowered Data-driven Networks: A Survey and Outlook. *ACM Comput. Surv.* **2022**, *54*, 58. [CrossRef]
143. Liu, T.; Liu, J.; Wang, J.; Zhai, D.; Liu, Y.; He, X. Anonymous Storage and Verification Model of IIoT Based on Blockchain: Anonymous storage and verification model of IIoT production status based on blockchain. In Proceedings of the 2021 4th International Conference on Blockchain Technology and Applications (ICBTA), Xi'an, China, 17–19 December 2021; pp. 144–150. [CrossRef]
144. Pathak, M.; Mishra, K.N.; Singh, S.P. Securing data and preserving privacy in cloud IoT-based technologies: An analysis of assessing threats and developing effective safeguards. *Artif. Intell. Rev.* **2024**, *57*, 269. [CrossRef]
145. Aguru, A.; Erukala, S. OTI-IoT: A Blockchain-based Operational Threat Intelligence Framework for Multi-vector DDoS Attacks. *ACM Trans. Internet Technol.* **2024**, *24*, 15. [CrossRef]
146. Hadian, M.; Erfani, S.H.; Deypir, M.; Mirabi, M. CD-TMS: A combinatorial design-based token management system to enhance security and performance in blockchain. *Clust. Comput.* **2024**, *27*, 4515–4536. [CrossRef]

147. He, Z.; Li, Z.; Qiao, A.; Luo, X.; Zhang, X.; Chen, T.; Song, S.; Liu, D.; Niu, W. Nurgle: Exacerbating Resource Consumption in Blockchain State Storage via MPT Manipulation. In Proceedings of the 2024 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 20–23 May 2024; pp. 2180–2197. [CrossRef]
148. Janjua, H.A.; Yue, L.; Hayat, S. Smart Scan: An Approach to Detect Denial of Service Vulnerability in Ethereum Smart Contracts. In Proceedings of the 2023 7th International Conference on Electronic Information Technology and Computer Engineering (EITCE), Xiamen, China, 13–15 October 2023; pp. 1572–1578. [CrossRef]
149. Li, K.; Wang, Y.; Tang, Y. DETER: Denial of Ethereum Txpool sERvices. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS), Coex, Republic of Korea, 15–19 November 2021; pp. 1645–1667. [CrossRef]
150. Mo, Y.; Chen, J.; Wang, Y.; Zheng, Z. Toward Automated Detecting Unanticipated Price Feed in Smart Contract. In Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA), Seattle, WA, USA, 17–21 July 2023; pp. 1257–1268. [CrossRef]
151. Verma, A.; Saha, R.; Kumar, N.; Kumar, G.; Tai-Hoon-Kim. A detailed survey of denial of service for IoT and multimedia systems: Past, present and futuristic development. *Multimed. Tools Appl.* **2022**, *81*, 19879–19944. [CrossRef]
152. Zkik, K.; Sebbar, A.; Fadi, O.; Kamble, S.; Belhadi, A. Securing blockchain-based crowdfunding platforms: An integrated graph neural networks and machine learning approach. *Electron. Commer. Res.* **2024**, *24*, 497–533. [CrossRef]
153. Rishiwal, V.; Agarwal, U.; Yadav, M.; Alotaibi, A.; Yadav, P.; Tanwar, S. Blockchain-Secure Gaming Environments: A Comprehensive Survey. *IEEE Access* **2024**, *12*, 183466–183488. [CrossRef]
154. Xia, P.; Wang, H.; Yu, Z.; Liu, X.; Luo, X.; Xu, G.; Tyson, G. Challenges in Decentralized Name Management: The Case of ENS. In Proceedings of the 22nd ACM Internet Measurement Conference (IMC), Nice, France, 25–27 October 2022; pp. 65–82. [CrossRef]
155. Wang, Y.; Hsin, W.J.; Lamsal, M. EdGENI: Making GENI User-Friendly for General Computer Education. In Proceedings of the 53rd ACM Technical Symposium on Computer Science Education (SIGCSE), Providence, RI, USA, 3–5 March 2022; pp. 801–807. [CrossRef]
156. Huang, J.; He, D.; Obaidat, M.S.; Vijayakumar, P.; Luo, M.; Choo, K.K.R. The Application of the Blockchain Technology in Voting Systems: A Review. *ACM Comput. Surv.* **2022**, *54*, 60. [CrossRef]
157. Ahmed-Rengers, M. FrameProv: Towards End-to-End Video Provenance. In Proceedings of the New Security Paradigms Workshop (NSPW), San Carlos, Costa Rica, 23–26 September 2019; pp. 68–77. [CrossRef]
158. Chen, W.; Li, X.; Sui, Y.; He, N.; Wang, H.; Wu, L.; Luo, X. SADPonzi: Detecting and Characterizing Ponzi Schemes in Ethereum Smart Contracts. *Proc. ACM Meas. Anal. Comput. Syst.* **2021**, *5*, 26. [CrossRef]
159. Faccia, A.; Moşteanu, N.R.; Cavaliere, L.P.L.; Mataruna-Dos-Santos, L.J. Electronic Money Laundering, The Dark Side of Fintech: An Overview of the Most Recent Cases. In Proceedings of the 2020 12th International Conference on Information Management and Engineering (ICIME), Amsterdam, The Netherlands, 18–20 September 2020; pp. 29–34. [CrossRef]
160. Fan, S.; Min, T.; Wu, X.; Cai, W. Altruistic and Profit-Oriented: Making Sense of Roles in Web3 Community from Airdrop Perspective. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI), Hamburg, Germany, 23–28 April 2023; pp. 1–16. [CrossRef]
161. Gürfidan, R. Suspicious transaction alert and blocking system for cryptocurrency exchanges in metaverse's social media universes: RG-guard. *Neural Comput. Appl.* **2024**, *36*, 18825–18840. [CrossRef]
162. He, B.; Feng, T.; Fang, J.; Liu, C.; Su, C. A Secure and Efficient Charitable Donation System Based on Ethereum Blockchain and Searchable Encryption. *IEEE Trans. Consum. Electron.* **2024**, *70*, 263–276. [CrossRef]
163. Huang, Y.; Wang, H.; Wu, L.; Tyson, G.; Luo, X.; Zhang, R.; Liu, X.; Huang, G.; Jiang, X. Understanding (Mis)Behavior on the EOSIO Blockchain. *Proc. ACM Meas. Anal. Comput. Syst.* **2020**, *4*, 37. [CrossRef]
164. Iyer, V.; Shah, K.; Rane, S.; Shankarmani, R. Decentralised Peer-to-Peer Crop Insurance. In Proceedings of the 3rd ACM International Symposium on Blockchain and Secure Critical Infrastructure (BSCI), Hong Kong, China, 7–11 June 2021; pp. 3–12. [CrossRef]
165. Kuchumov, A.; Pecheritsa, E.; Chaikovskaya, A.; Maslova, E. Digitalization of Economics: Modern Financial Technologies and Their Influence on Economic Security. In Proceedings of the IV International Scientific and Practical Conference, St. Petersburg, Russia, 15–17 September 2021; pp. 1–7. [CrossRef]
166. Pan, B.; Stakhanova, N.; Zhu, Z. EtherShield: Time-interval Analysis for Detection of Malicious Behavior on Ethereum. *ACM Trans. Internet Technol.* **2024**, *24*, 2. [CrossRef]
167. Ralli, R.; Jugran, G.; Gaurav, M.; Goyal, M. An Ensemble Based Fraudulent Blockchain Account Detection System. In Proceedings of the 2024 Sixteenth International Conference on Contemporary Computing (IC3), Noida, India, 8–10 August 2024; pp. 337–342. [CrossRef]
168. Saranya, A.; Naresh, R. Block chain-based trusted smart contract for secure mobile user payment gateway in e-health systems. *Soft Comput.* **2024**, *28*, 10139–10151. [CrossRef]

169. Su, J.; Lin, X.; Fang, Z.; Zhu, Z.; Chen, J.; Zheng, Z.; Lv, W.; Wang, J. DeFiWarder: Protecting DeFi Apps from Token Leaking Vulnerabilities. In Proceedings of the 2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE), Luxembourg, 11–15 September 2023; pp. 1664–1675. [CrossRef]
170. Wu, C.; Chen, J.; Wang, Z.; Liang, R.; Du, R. Semantic Sleuth: Identifying Ponzi Contracts via Large Language Models. In Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering (ASE), Sacramento, CA, USA, 16–20 September 2024; pp. 582–593. [CrossRef]
171. Wu, C.; Chen, J.; Zhao, Z.; He, K.; Xu, G.; Wu, Y.; Wang, H.; Li, H.; Liu, Y.; Xiang, Y. TokenScout: Early Detection of Ethereum Scam Tokens via Temporal Graph Learning. In Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS), Salt Lake City, UT, USA, 14–18 October 2024; pp. 956–970. [CrossRef]
172. Wu, S.; Wu, L.; Zhou, Y.; Li, R.; Wang, Z.; Luo, X.; Wang, C.; Ren, K. Time-travel Investigation: Toward Building a Scalable Attack Detection Framework on Ethereum. *ACM Trans. Softw. Eng. Methodol.* **2022**, *31*, 54. [CrossRef]
173. Zhou, Y.; Sun, J.; Ma, F.; Chen, Y.; Yan, Z.; Jiang, Y. Stop Pulling My Rug: Exposing Rug Pull Risks in Crypto Token to Investors. In Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), Lisbon, Portugal, 14–20 April 2024; pp. 228–239. [CrossRef]
174. Amin Rezaei, A.; Precht, H.; Marx Gómez, J. Supporting Prostitutes Protection Act through DLT. In Proceedings of the 2021 The 3rd International Conference on Blockchain Technology (ICBCT), Shanghai, China, 26–28 March 2021; pp. 102–108. [CrossRef]
175. Le, M.Q.; Le, H.D.; Dinh-Duc, A.V.; Tran, T.T. IU-TransCert: A Blockchain-Based System for Academic Credentials with Auditability. In Proceedings of the 12th International Symposium on Information and Communication Technology (SoICT), Ho Chi Minh City, Vietnam, 7–8 December 2023; pp. 746–753. [CrossRef]
176. Grüner, A.; Mühle, A.; Lockenvitz, N.; Meinel, C. Analyzing and comparing the security of self-sovereign identity management systems through threat modeling. *Int. J. Inf. Secur.* **2023**, *22*, 1231–1248. [CrossRef]
177. Putz, B.; Vielberth, M.; Pernul, G. BISCUIT—Blockchain Security Incident Reporting Based on Human Observations. In Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES), Vienna, Austria, 23–26 August 2022; pp. 1–6. [CrossRef]
178. Singh, J.; Sinha, A.; Goli, P.; Subramanian, V.; Shukla, S.K.; Vyas, O.P. Insider attack mitigation in a smart metering infrastructure using reputation score and blockchain technology. *Int. J. Inf. Secur.* **2022**, *21*, 527–546. [CrossRef]
179. Fang, Q. Designing of music copyright protection system based on deep belief network and blockchain. *Soft Comput.* **2024**, *28*, 1669–1684. [CrossRef]
180. Hamza, R.; Dao, M.S.; Ito, S.; Koji, Z. Towards Intellectual Property Rights Protection in Big Data. In Proceedings of the 3rd ACM Workshop on Intelligent Cross-Data Analysis and Retrieval (ICDAR), Newark, NJ, USA, 1 July 2022; pp. 50–57. [CrossRef]
181. Wang, F.; Fu, Z.; Zhang, X. A Self-Defense Copyright Protection Scheme for NFT Image Art Based on Information Embedding. *ACM Trans. Multimed. Comput. Commun. Appl.* **2025**, *21*, 42. [CrossRef]
182. Xu, L.; Chen, L.; Gao, Z.; Fan, X.; Shi, W. DL-DP: Improving the Security of Industrial IoT with Decentralized Ledger Defined Perimeter. In Proceedings of the 2nd ACM International Symposium on Blockchain and Secure Critical Infrastructure, Taipei, Taiwan, 27–29 October 2020; pp. 53–62. [CrossRef]
183. Thomasset, C.; Barrera, D. SERENIoT: Distributed Network Security Policy Management and Enforcement for Smart Homes. In Proceedings of the 2020 Annual Computer Security Applications Conference (ACSAC), Austin, TX, USA, 7–11 December 2020; pp. 542–555. [CrossRef]
184. Sharma, T.; Prasad, S.K. Enhancing cybersecurity in IoT networks: SLSTM-WCO algorithm for anomaly detection. *Peer-to-Peer Netw. Appl.* **2024**, *17*, 2237–2258. [CrossRef]
185. Saxena, A.; Mittal, S. Internet of Medical Things (IoMT) Security and Privacy: A Survey of Recent Advances and Enabling Technologies. In Proceedings of the 2022 Fourteenth International Conference on Contemporary Computing (IC3), Noida, India, 4–6 August 2022; pp. 550–559. [CrossRef]
186. Rui, H.; Huan, L.; Yang, H.; YunHao, Z. Research on secure transmission and storage of energy IoT information based on Blockchain. *Peer-to-Peer Netw. Appl.* **2020**, *13*, 1225–1235. [CrossRef]
187. Prajisha, C.; Vasudevan, A.R. An efficient intrusion detection system for MQTT-IoT using enhanced chaotic salp swarm algorithm and LightGBM. *Int. J. Inf. Secur.* **2022**, *21*, 1263–1282. [CrossRef]
188. Paul, A.K.; Qu, X.; Wen, Z. Blockchain–a promising solution to internet of things: A comprehensive analysis, opportunities, challenges and future research issues. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 2926–2951. [CrossRef]
189. Olawale, O.P.; Ebadinezhad, S. Cybersecurity Anomaly Detection: AI and Ethereum Blockchain for a Secure and Tamperproof IoHT Data Management. *IEEE Access* **2024**, *12*, 131605–131620. [CrossRef]

190. Niavis, H.; Loupos, K. ConSenseIoT: A Consensus Algorithm for Secure and Scalable Blockchain in the IoT context. In Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES 2022), Vienna, Austria, 23–26 August 2022; pp. 1–6. [CrossRef]
191. Mathew, S.S.; Hayawi, K.; Dawit, N.A.; Taleb, I.; Trabelsi, Z. Integration of blockchain and collaborative intrusion detection for secure data transactions in industrial IoT: A survey. *Clust. Comput.* **2022**, *25*, 4129–4149. [CrossRef]
192. Liu, Y.; Zhang, J.; Zhan, J. Privacy protection for fog computing and the internet of things data based on blockchain. *Clust. Comput.* **2021**, *24*, 1331–1345. [CrossRef]
193. Lai, E.; Luo, W. Static Analysis of Integer Overflow of Smart Contracts in Ethereum. In Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy (CSP 2020), Nanjing, China, 10–12 January 2020; pp. 110–115. [CrossRef]
194. Kokila, M.; Srinivasa Reddy, K. BlockDLO: Blockchain Computing With Deep Learning Orchestration for Secure Data Communication in IoT Environment. *IEEE Access* **2024**, *12*, 134521–134540. [CrossRef]
195. Kamal, R.; Hemdan, E.E.D.; El-Fishway, N. A review study on blockchain-based IoT security and forensics. *Multimed. Tools Appl.* **2021**, *80*, 36183–36214. [CrossRef]
196. Janani, K.; Ramamoorthy, S. A security framework to enhance IoT device identity and data access through blockchain consensus model. *Clust. Comput.* **2024**, *27*, 2877–2900. [CrossRef]
197. Ilakkiya, N.; Rajaram, A. A secured trusted routing using the structure of a novel directed acyclic graph-blockchain in mobile ad hoc network internet of things environment. *Multimed. Tools Appl.* **2024**, *83*, 87903–87928. [CrossRef]
198. Ghadi, Y.Y.; Mazhar, T.; Shahzad, T.; Amir Khan, M.; Abd-Alrazaq, A.; Ahmed, A.; Hamam, H. The role of blockchain to secure internet of medical things. *Sci. Rep.* **2024**, *14*, 18422. [CrossRef]
199. Bansal, K.; Singhrova, A. Review on intrusion detection system for IoT/IIo -brief study. *Multimed. Tools Appl.* **2023**, *83*, 23083–23108. [CrossRef]
200. Alfandi, O.; Khanji, S.; Ahmad, L.; Khattak, A. A survey on boosting IoT security and privacy through blockchain: Exploration, requirements, and open issues. *Clust. Comput.* **2021**, *24*, 37–55. [CrossRef]
201. Jawahar, A.; Kaythry, P.; Vinoth Kumar, C.; Vinu, R.; Amrish, R.; Bavapriyan, K.; Gopinaath, V. DDoS mitigation using blockchain and machine learning techniques. *Multimed. Tools Appl.* **2024**, *83*, 60265–60278. [CrossRef]
202. Moussaileb, R.; Cuppens, N.; Lanet, J.L.; Bouder, H.L. A Survey on Windows-based Ransomware Taxonomy and Detection Mechanisms. *ACM Comput. Surv.* **2022**, *54*, 117. [CrossRef]
203. England, P.; Malvar, H.S.; Horvitz, E.; Stokes, J.W.; Fournet, C.; Burke-Aguero, R.; Chamayou, A.; Clebsch, S.; Costa, M.; Deutscher, J.; et al. AMP: Authentication of Media via Provenance. In Proceedings of the 12th ACM Multimedia Systems Conference (MMSys '21), Istanbul, Turkey, 28–30 September 2021; pp. 108–121. [CrossRef]
204. Gambín, Á.F.; Yazidi, A.; Vasilakos, A.; Haugerud, H.; Djenouri, Y. Deepfakes: Current and future trends. *Artif. Intell. Rev.* **2024**, *57*, 64. [CrossRef]
205. Masood, M.; Nawaz, M.; Malik, K.M.; Javed, A.; Irtaza, A.; Malik, H. Deepfakes generation and detection: State-of-the-art, open challenges, countermeasures, and way forward. *Appl. Intell.* **2023**, *53*, 3974–4026. [CrossRef]
206. Singleton, L.; Zhao, R.; Song, M.; Siy, H. CryptoTutor: Teaching Secure Coding Practices through Misuse Pattern Detection. In Proceedings of the 21st Annual Conference on Information Technology Education (SIGITE '20), Omaha, NE, USA, 7–9 October 2020; pp. 403–408. [CrossRef]
207. Hua, B.; Ouyang, W.; Jiang, C.; Fan, Q.; Pan, Z. Rupair: Towards Automatic Buffer Overflow Detection and Rectification for Rust. In Proceedings of the Annual Computer Security Applications Conference (ACSAC '21), Austin, TX, USA, 6–10 December 2021; pp. 812–823. [CrossRef]
208. Alfriehat, N.; Anbar, M.; Aladaileh, M.; Hasbullah, I.; Shurbaji, T.A.; Karuppayah, S.; Almomani, A. RPL-based attack detection approaches in IoT networks: Review and taxonomy. *Artif. Intell. Rev.* **2024**, *57*, 248. [CrossRef]
209. Coretti, S.; Kiayias, A.; Moore, C.; Russell, A. The Generals' Scuttlebutt: Byzantine-Resilient Gossip Protocols. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22), Los Angeles, CA, USA, 7–11 November 2022; pp. 595–608. [CrossRef]
210. Dai, T.; Jeitner, P.; Shulman, H.; Waidner, M. From IP to transport and beyond: Cross-layer attacks against applications. In Proceedings of the 2021 ACM SIGCOMM 2021 Conference, Virtual Event, USA, 23–27 August 2021; pp. 836–849. [CrossRef]
211. Das, D.; Banerjee, S.; Ghosh, U.; Biswas, U.; Bashir, A.K. A decentralized vehicle anti-theft system using Blockchain and smart contracts. *Peer- Netw. Appl.* **2021**, *14*, 2775–2788. [CrossRef]
212. Falzon, F.; Elkhiyaoui, K.; Manevich, Y.; De Caro, A. Short Privacy-Preserving Proofs of Liabilities. In Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23), Copenhagen, Denmark, 26–30 November 2023; pp. 1805–1819. [CrossRef]

213. Luo, R.; Luo, F.; Wang, B.; Chen, T. Smart Contract Vulnerability Detection Based on Variant LSTM. In Proceedings of the 2022 International Conference on Big Data, IoT, and Cloud Computing (ICBDICC '22), Chengdu, China, 23–25 December 2022; pp. 1–4. [CrossRef]
214. Mutemwa, M.; Masango, M.G.; Gcaza, N. Managing the Shift in the Enterprise Perimeter in Order to Delay a Cybersecurity Breach. In Proceedings of the International Conference on Artificial Intelligence and Its Applications, Virtual Event, Mauritius, 18–20 August 2021; pp. 1–10. [CrossRef]
215. Su, Y.; Wang, B.; Xing, Q.; Wang, X. DISCA: Decentralized Infrastructure for Secure Community Attribute Certifying. In Proceedings of the 3rd International Conference on Advanced Information Science and System, Sanya, China, 19–21 November 2021; pp. 1–8. [CrossRef]
216. Arora, S.; Li, Y.; Feng, Y.; Xu, J. SecPLF: Secure Protocols for Loanable Funds against Oracle Manipulation Attacks. In Proceedings of the 19th ACM Asia Conference on Computer and Communications Security (AsiaCCS), Singapore, 1–5 July 2024; pp. 1394–1405. [CrossRef]
217. Chen, Z.; Beillahi, S.M.; Long, F. FlashSyn: Flash Loan Attack Synthesis via Counter Example Driven Approximation. In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering (ICSE), Lisbon, Portugal, 14–20 April 2024; pp. 1–13. [CrossRef]
218. Eskandari, S.; Salehi, M.; Gu, W.C.; Clark, J. SoK: Oracles from the Ground Truth to Market Manipulation. In Proceedings of the 3rd ACM Conference on Advances in Financial Technologies, Arlington, VA, USA, 20–22 September 2021; pp. 127–141. [CrossRef]
219. Xie, M.; Hu, M.; Kong, Z.; Zhang, C.; Feng, Y.; Wang, H.; Xue, Y.; Zhang, H.; Liu, Y.; Liu, Y. DeFort: Automatic Detection and Analysis of Price Manipulation Attacks in DeFi Applications. In Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA), Vienna, Austria, 15–19 July 2024; pp. 402–414. [CrossRef]
220. Cao, C.; Wu, J.; Qi, H.; Eda, S. Blockchain-based Runtime Attestation against Physical Fault Injection Attacks on Edge Devices. In Proceedings of the Eighth ACM/IEEE Symposium on Edge Computing (SEC), Wilmington, DE, USA, 6–9 December 2023; pp. 133–144. [CrossRef]
221. Kavya, S.; Sumathi, D. Staying ahead of phishers: A review of recent advances and emerging methodologies in phishing detection. *Artif. Intell. Rev.* **2024**, *58*, 50. [CrossRef]
222. Li, S.; Gou, G.; Liu, C.; Xiong, G.; Li, Z.; Xiao, J.; Xing, X. TGC: Transaction Graph Contrast Network for Ethereum Phishing Scam Detection. In Proceedings of the Annual Computer Security Applications Conference (ACSAC), Austin, TX, USA, 4–8 December 2023; pp. 352–365. [CrossRef]
223. Ali, W.; Kumar, R.; Zhou, X.; Shao, J. Responsible Recommendation Services with Blockchain Empowered Asynchronous Federated Learning. *ACM Trans. Intell. Syst. Technol.* **2024**, *15*, 78. [CrossRef]
224. Alja'afreh, M.; Al Mallah, R.; Karime, A.; El Saddik, A. Cybersecurity in the Metaverse: Challenges and Approaches. In Proceedings of the 2023 International Conference on Intelligent Metaverse Technologies & Applications (iMETA), Tartu, Estonia, 2–4 October 2023; pp. 1–8. [CrossRef]
225. Dargahi, T.; Ahmadvand, H.; Alraja, M.N.; Yu, C.M. Integration of Blockchain with Connected and Autonomous Vehicles: Vision and Challenge. *J. Data Inf. Qual.* **2022**, *14*, 5. [CrossRef]
226. Desai, N.; Maesa, D.D.F.; Sastry, N.; Schneider, S.; Ricci, L. Preserving Privacy of Vulnerable Users across Heterogeneous Sensitive Sensor Data Streams using Smart Contracts. In Proceedings of the Fifth ACM International Workshop on Blockchain-enabled Networked Sensor Systems, Istanbul, Turkiye, 13 November 2023; pp. 2–8. [CrossRef]
227. Haque, S.M.U.; Sofi, S.A.; Sholla, S. A privacy-preserving deep learning framework for highly authenticated blockchain secure storage system. *Multimed. Tools Appl.* **2024**, *83*, 84299–84329. [CrossRef]
228. He, Z.; Song, S.; Bai, Y.; Luo, X.; Chen, T.; Zhang, W.; He, P.; Li, H.; Lin, X.; Zhang, X. TokenAware: Accurate and Efficient Bookkeeping Recognition for Token Smart Contracts. *ACM Trans. Softw. Eng. Methodol.* **2023**, *32*, 26. [CrossRef]
229. Jia, Y.; Madathil, V.; Kate, A. HomeRun: High-efficiency Oblivious Message Retrieval, Unrestricted. In Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security, Salt Lake City, UT, USA, 14–18 October 2024; pp. 2012–2026. [CrossRef]
230. Kashif, M.; Kalkan, K. Differential privacy preserving based framework using blockchain for internet-of-things. *Peer-to-Peer Netw. Appl.* **2025**, *18*, 33. [CrossRef]
231. Kasyap, H.; Tripathy, S. Privacy-preserving Decentralized Learning Framework for Healthcare System. *ACM Trans. Multimed. Comput. Commun. Appl.* **2021**, *17*, 68. [CrossRef]
232. Li, Y.; Soska, K.; Huang, Z.; Bellemare, S.; Quintyne-Collins, M.; Wang, L.; Liu, X.; Song, D.; Miller, A. Ratel: MPC-extensions for Smart Contracts. In Proceedings of the 19th ACM Asia Conference on Computer and Communications Security, Singapore, 1–5 July 2024; pp. 336–352. [CrossRef]
233. Liang, W.; Ji, N. Privacy challenges of IoT-based blockchain: A systematic review. *Clust. Comput.* **2022**, *25*, 2203–2221. [CrossRef]

234. Mahajan, H.B.; Junnarkar, A.A. Smart healthcare system using integrated and lightweight ECC with private blockchain for multimedia medical data processing. *Multimed. Tools Appl.* **2023**, *82*, 44335–44358. [CrossRef]
235. Wang, Q.; He, L.; Zhu, X.; Huang, Y.; Li, Z. Privacy Protection of Blockchain Security Development Status. In Proceedings of the 2021 4th International Conference on Information Systems and Computer Aided Education (ICISCAE), Dalian, China, 17–19 September 2021; pp. 2592–2596. [CrossRef]
236. Wang, Q.; Chen, S.; Xiang, Y. Anonymous Blockchain-based System for Consortium. *ACM Trans. Manag. Inf. Syst.* **2021**, *12*, 26. [CrossRef]
237. Yang, H.; Yuan, L.; Wang, S. Design of Blockchain Smart Contract Based on Ring Signature. In Proceedings of the 2021 9th International Conference on Communications and Broadband Networking (ICCBN), Shanghai, China, 15–17 January 2021; pp. 108–114. [CrossRef]
238. Ye, Z.; Chen, C.L.; Weng, W.; Sun, H.; Tsaur, W.J.; Deng, Y.Y. An anonymous and fair auction system based on blockchain. *J. Supercomput.* **2023**, *79*, 13909–13951. [CrossRef]
239. Zhang, Y.; Wu, Y.; Wang, J.; Liu, B.; Liu, A.; Chen, X. Blockchain Query Framework Based on Trusted Execution Environment. In Proceedings of the 2024 5th International Conference on Computing, Networks and Internet of Things, Tokyo, Japan, 26–28 January 2024; pp. 182–185. [CrossRef]
240. Zhang, Z.; Lei, Y.; Yan, M.; Yu, Y.; Chen, J.; Wang, S.; Mao, X. Reentrancy Vulnerability Detection and Localization: A Deep Learning Based Two-phase Approach. In Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering (ASE), Rochester, MI, USA, 10–14 October 2022; pp. 1–13. [CrossRef]
241. Liao, J.; Zhang, F.; Sun, W.; Shi, W. Speedster: An Efficient Multi-party State Channel via Enclaves. In Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security, Nagasaki, Japan, 30 May–3 June 2022; pp. 637–651. [CrossRef]
242. Alexopoulos, N.; Vasilomanolakis, E.; Roux, S.L.; Rowe, S.; Mühlhäuser, M. TRIDEnT: Towards a decentralized threat indicator marketplace. In Proceedings of the 35th Annual ACM Symposium on Applied Computing, Brno, Czech Republic, 30 March–3 April 2020; pp. 332–341. [CrossRef]
243. Ashizawa, N.; Yanai, N.; Cruz, J.P.; Okamura, S. Eth2Vec: Learning Contract-Wide Code Representations for Vulnerability Detection on Ethereum Smart Contracts. In Proceedings of the 3rd ACM International Symposium on Blockchain and Secure Critical Infrastructure, Hong Kong, China, 3–5 June 2021; pp. 47–59. [CrossRef]
244. Bhamidipati, V.S.V.; Chan, M.; Chamorro, D.; Jain, A.; Murthy, A. Adaptive Security for Smart Contracts using High Granularity Metrics. In Proceedings of the 3rd International Conference on Vision, Image and Signal Processing, Vancouver, BC, Canada, 28–30 November 2019; pp. 1–6. [CrossRef]
245. Chen, W.; Sun, Z.; Wang, H.; Luo, X.; Cai, H.; Wu, L. WASAI: Uncovering vulnerabilities in Wasm smart contracts. In Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis, Daejeon, Republic of Korea, 18–22 July 2022; pp. 703–715. [CrossRef]
246. Chen, Z.; Liu, Y.; Beillahi, S.M.; Li, Y.; Long, F. Demystifying Invariant Effectiveness for Securing Smart Contracts. *Proc. ACM Softw. Eng.* **2024**, *1*, 1772–1795. [CrossRef]
247. Chua, M.Y.K.; Yee, G.O.M.; Gu, Y.X.; Lung, C.H. Threats to Online Advertising and Countermeasures: A Technical Survey. *Digit. Threat. Res. Pract.* **2020**, *1*, 11. [CrossRef]
248. Coblenz, M.; Oei, R.; Etzel, T.; Koronkevich, P.; Baker, M.; Bloem, Y.; Myers, B.A.; Sunshine, J.; Aldrich, J. Obsidian: Typestate and Assets for Safer Blockchain Programming. *ACM Trans. Program. Lang. Syst.* **2020**, *42*, 14. [CrossRef]
249. Crincoli, G.; Iadarola, G.; La Rocca, P.E.; Martinelli, F.; Mercaldo, F.; Santone, A. Vulnerable Smart Contract Detection by Means of Model Checking. In Proceedings of the Fourth ACM International Symposium on Blockchain and Secure Critical Infrastructure, Nagasaki, Japan, 18–20 May 2022; pp. 3–10. [CrossRef]
250. Cui, S.; Zhao, G.; Gao, Y.; Tavu, T.; Huang, J. VRust: Automated Vulnerability Detection for Solana Smart Contracts. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, Los Angeles, CA, USA, 7–11 November 2022; pp. 639–652. [CrossRef]
251. Eshghie, M.; Artho, C.; Gurov, D. Dynamic Vulnerability Detection on Smart Contracts Using Machine Learning. In Proceedings of the Evaluation and Assessment in Software Engineering, Trondheim, Norway, 21–22 June 2021; pp. 305–312. [CrossRef]
252. Gao, C.; Yang, W.; Ye, J.; Xue, Y.; Sun, J. sGuard+: Machine Learning Guided Rule-Based Automated Vulnerability Repair on Smart Contracts. *ACM Trans. Softw. Eng. Methodol.* **2024**, *33*, 114. [CrossRef]
253. Garfatta, I.; Klai, K.; Graïet, M.; Gaaloul, W. Model Checking of Vulnerabilities in Smart Contracts: A Solidity-to-CPN Approach. In Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing, Virtual Event, 25–29 April 2022; pp. 316–325. [CrossRef]

254. Ghaleb, A. Towards Effective Static Analysis Approaches for Security Vulnerabilities in Smart Contracts. In Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering, Rochester, MI, USA, 10–14 October 2022; pp. 1–5. [CrossRef]
255. Hamdi, A.; Fourati, L.; Ayed, S. Vulnerabilities and attacks assessments in blockchain 1.0, 2.0 and 3.0: Tools, analysis and countermeasures. *Int. J. Inf. Secur.* **2024**, *23*, 713–757. [CrossRef]
256. Hao, X.; Ren, W.; Zheng, W.; Zhu, T. SCScan: A SVM-Based Scanning System for Vulnerabilities in Blockchain Smart Contracts. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 29 December–1 January 2021; pp. 1598–1605. [CrossRef]
257. Hettmer, M.; Severin, B.; Blum, F.; Gruhn, V. Towards Assessing the Real-World Impact of Defects in Blockchain-Based Smart Contracts. In Proceedings of the 1st International Workshop on Software Defect Datasets, San Francisco, CA, USA, 11 July 2023; pp. 6–10. [CrossRef]
258. Huang, Y.; Jiang, B.; Chan, W.K. EOSFuzzer: Fuzzing EOSIO Smart Contracts for Vulnerability Detection. In Proceedings of the 12th Asia-Pacific Symposium on Internetware, Singapore, 28–30 December 2020; pp. 99–109. [CrossRef]
259. Ibba, G.; Aufiero, S.; Neykova, R.; Bartolucci, S.; Ortu, M.; Tonelli, R.; Destefanis, G. A Curated Solidity Smart Contracts Repository of Metrics and Vulnerability. In Proceedings of the 20th International Conference on Predictive Models and Data Analytics in Software Engineering, Porto de Galinhas, Brazil, 14–18 October 2024; pp. 32–41. [CrossRef]
260. J J, L.; Singh, K.; Chakravarthi, B. Digital forensic framework for smart contract vulnerabilities using ensemble models. *Multimed. Tools Appl.* **2023**, *83*, 51469–51512. [CrossRef]
261. Jiao, T.; Xu, Z.; Qi, M.; Wen, S.; Xiang, Y.; Nan, G. A Survey of Ethereum Smart Contract Security: Attacks and Detection. *Distrib. Ledger Technol. Res. Pract.* **2024**, *3*, 23. [CrossRef]
262. Khor, J.; Masama, M.A.; Sidorov, M.; Leong, W.; Lim, J. An Improved Gas Efficient Library for Securing IoT Smart Contracts Against Arithmetic Vulnerabilities. In Proceedings of the 2020 9th International Conference on Software and Computer Applications, Langkawi, Malaysia, 18–21 February 2020; pp. 326–330. [CrossRef]
263. Kitzler, S.; Victor, F.; Saggese, P.; Haslhofer, B. Disentangling Decentralized Finance (DeFi) Compositions. *ACM Trans. Web* **2023**, *17*, 10. [CrossRef]
264. Lê Hồng, B.; Lê Đức, T.; Đoàn Minh, T.; Trần Tuấn, D.; Phan Thế, D.; Phạm Văn, H. 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 (SoICT), Ho Chi Minh City, Vietnam, 7–8 December 2023; pp. 739–745. [CrossRef]
265. Li, P.; Wang, G.; Xing, X.; Zhu, J.; Gu, W.; Zhai, G. A smart contract vulnerability detection method based on deep learning with opcode sequences. *Peer-to-Peer Netw. Appl.* **2024**, *17*, 3222–3238. [CrossRef]
266. Li, P.; Li, S.; Ding, M.; Yu, J.; Zhang, H.; Zhou, X.; Li, J. A Vulnerability Detection Framework for Hyperledger Fabric Smart Contracts Based on Dynamic and Static Analysis. In Proceedings of the International Conference on Evaluation and Assessment in Software Engineering (EASE), Gothenburg, Sweden, 13–15 June 2022; pp. 366–374. [CrossRef]
267. Li, W.; Li, X.; Li, Z.; Zhang, Y. COBRA: Interaction-Aware Bytecode-Level Vulnerability Detector for Smart Contracts. In Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering (ASE), Sacramento, CA, USA, 21–25 October 2024; pp. 1358–1369. [CrossRef]
268. Li, Z.; Zhou, Y.; Guo, S.; Xiao, B. SolSaviour: A Defending Framework for Deployed Defective Smart Contracts. In Proceedings of the Annual Computer Security Applications Conference (ACSAC), Austin, TX, USA, 6–10 December 2021; pp. 748–760. [CrossRef]
269. Liao, X. Smart Contract Vulnerability Detection Based on Dynamic and Static Combination. In Proceedings of the International Conference on Digital Economy, Blockchain and Artificial Intelligence, Guangzhou, China, 10–11 April 2024; pp. 412–416. [CrossRef]
270. Liao, Z.; Zheng, Z.; Chen, X.; Nan, Y. SmartDagger: A Bytecode-Based Static Analysis Approach for Detecting Cross-Contract Vulnerability. In Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA), Daejeon, Republic of Korea, 18–22 July 2022; pp. 752–764. [CrossRef]
271. Ma, C.; Song, W.; Huang, J. TransRacer: Function Dependence-Guided Transaction Race Detection for Smart Contracts. In Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), San Francisco, CA, USA, 11–17 December 2023; pp. 947–959. [CrossRef]
272. Mi, F.; Zhao, C.; Wang, Z.; Halim, S.M.; Li, X.; Wu, Z.; Khan, L.; Thuraisingham, B. An Automated Vulnerability Detection Framework for Smart Contracts. *Distributed Ledger Technologies: Research and Practice* **2024**, *3*, 3705616. [CrossRef]
273. Nguyen, H.H.; Nguyen, N.M.; Doan, H.P.; Ahmadi, Z.; Doan, T.N.; Jiang, L. MANDO-GURU: Vulnerability Detection for Smart Contract Source Code by Heterogeneous Graph Embeddings. In Proceedings of the 30th ACM Joint European Software

Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), Singapore, 14–18 November 2022; pp. 1736–1740. [CrossRef]
274. Otoni, R.; Marescotti, M.; Alt, L.; Eugster, P.; Hyvärinen, A.; Sharygina, N. A Solicitous Approach to Smart Contract Verification. *ACM Trans. Priv. Secur.* **2023**, *26*, 15. [CrossRef]
275. Pani, S.; Nallagonda, H.V.; Vigneswaran; Medicherla, R.K.; Rajan, M. SmartFuzzDriverGen: Smart Contract Fuzzing Automation for Golang. In Proceedings of the 16th Innovations in Software Engineering Conference (ISEC), Allahabad, India, 23–25 February 2023; pp. 1–11. [CrossRef]
276. Patel, A.; Chauhan, K.; Maini, S.; Goyal, M. Smart Contract Vulnerabilities Detection Using Deep Learning. In Proceedings of the 2024 Sixteenth International Conference on Contemporary Computing (IC3), Noida, India, 1–3 August 2024; pp. 349–356. [CrossRef]
277. Qian, P.; Liu, Z.; Yin, Y.; He, Q. Cross-Modality Mutual Learning for Enhancing Smart Contract Vulnerability Detection on Bytecode. In Proceedings of the ACM Web Conference 2023 (WWW '23), Austin, TX, USA, 30 April–4 May 2023; pp. 2220–2229. [CrossRef]
278. Russo, A.; Lax, G.; Dromard, B.; Mezred, M. A System to Access Online Services with Minimal Personal Information Disclosure. *Inf. Syst. Front.* **2022**, *24*, 1563–1575. [CrossRef]
279. Su, P.; Hu, J. Smart contract vulnerabilities detection with bidirectional encoder representations from transformers and control flow graph. *Multimed. Syst.* **2024**, *30*, 204. [CrossRef]
280. Wang, B.; Yuan, X.; Duan, L.; Ma, H.; Wang, B.; Su, C.; Wang, W. DeFiScanner: Spotting DeFi Attacks Exploiting Logic Vulnerabilities on Blockchain. *IEEE Trans. Comput. Soc. Syst.* **2024**, *11*, 1577–1588. [CrossRef]
281. Wang, C.; Li, Y.; Gao, J.; Wang, K.; Zhang, J.; Guan, Z.; Chen, Z. SolaSim: Clone Detection for Solana Smart Contracts via Program Representation. In Proceedings of the 32nd IEEE/ACM International Conference on Program Comprehension (ICPC), Lisbon, Portugal, 16–17 May 2024; pp. 258–269. [CrossRef]
282. Wang, H.; Hu, Y.; Wu, H.; Liu, D.; Peng, C.; Wu, Y.; Fan, M.; Liu, T. Skyeye: Detecting Imminent Attacks via Analyzing Adversarial Smart Contracts. In Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering (ASE), Sacramento, CA, USA, 27 October–1 November 2024; pp. 1570–1582. [CrossRef]
283. Wen, H.; Liu, H.; Song, J.; Chen, Y.; Guo, W.; Feng, Y. FORAY: Towards Effective Attack Synthesis against Deep Logical Vulnerabilities in DeFi Protocols. In Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS), Salt Lake City, UT, USA, 14–18 October 2024; pp. 1001–1015. [CrossRef]
284. Xue, Y.; Ma, M.; Lin, Y.; Sui, Y.; Ye, J.; Peng, T. Cross-contract Static Analysis for Detecting Practical Reentrancy Vulnerabilities in Smart Contracts. In Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering (ASE), Melbourne, Australia, 21–25 September 2020; pp. 1029–1040. [CrossRef]
285. Yan, C.; Zhang, C.; Lu, Z.; Wang, Z.; Liu, Y.; Liu, B. Blockchain abnormal behavior awareness methods: A survey. *Cybersecurity* **2022**, *5*, 5. [CrossRef]
286. Zheng, Z.; Zhang, N.; Su, J.; Zhong, Z.; Ye, M.; Chen, J. Turn the Rudder: A Beacon of Reentrancy Detection for Smart Contracts on Ethereum. In Proceedings of the 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), Melbourne, Australia, 14–20 May 2023; pp. 295–306. [CrossRef]
287. Ding, M.; Li, P.; Li, S.; Zhang, H. HFContractFuzzer: Fuzzing Hyperledger Fabric Smart Contracts for Vulnerability Detection. In Proceedings of the Evaluation and Assessment in Software Engineering (EASE), Trondheim, Norway, 21–23 June 2021; pp. 321–328. [CrossRef]
288. Dong, C.; Huang, H.; Shang, Y. Erinys: Efficient Fuzzing by Function Invoke Sequence Generation for Smart Contracts. In Proceedings of the 2024 8th International Conference on Big Data and Internet of Things (BDIOT), Macau, China, 23–25 August 2024; pp. 236–241. [CrossRef]
289. Jang, D.; Askar, A.; Yun, I.; Tong, S.; Cai, Y.; Kim, T. Fuzzing@Home: Distributed Fuzzing on Untrusted Heterogeneous Clients. In Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses (RAID), Limassol, Cyprus, 26–28 October 2022; pp. 1–16. [CrossRef]
290. Ren, M.; Yin, Z.; Ma, F.; Xu, Z.; Jiang, Y.; Sun, C.; Li, H.; Cai, Y. Empirical Evaluation of Smart Contract Testing: What Is the Best Choice? In Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA), Aarhus, Denmark, 11–17 July 2021; pp. 566–579. [CrossRef]
291. Ye, M.; Nan, Y.; Dai, H.N.; Yang, S.; Luo, X.; Zheng, Z. FunFuzz: A Function-Oriented Fuzzer for Smart Contract Vulnerability Detection with High Effectiveness and Efficiency. *ACM Trans. Softw. Eng. Methodol.* **2024**, *33*, 191. [CrossRef]
292. Zhao, X.; Qu, H.; Xu, J.; Li, X.; Lv, W.; Wang, G.G. A systematic review of fuzzing. *Soft Comput.* **2024**, *28*, 5493–5522. [CrossRef]
293. Aniello, L.; Halak, B.; Chai, P.; Dhall, R.; Mihalea, M.; Wilczynski, A. Anti-BlUFf: Towards counterfeit mitigation in IC supply chains using blockchain and PUF. *Int. J. Inf. Secur.* **2021**, *20*, 445–460. [CrossRef]

294. Marjanović, J.; Dalčeković, N.; Sladić, G. Improving Critical Infrastructure Protection by Enhancing Software Acquisition Process Through Blockchain. In Proceedings of the 7th Conference on the Engineering of Computer Based Systems, Novi Sad, Serbia, 2–3 September 2021; pp. 1–7. [CrossRef]
295. Xuan, S.; Tang, H.; Wang, W.; Yang, W. Application of Block Chain Technology in Constructing Network Threat Intelligence System. In Proceedings of the 2020 2nd International Conference on Blockchain Technology, Hilo, HI, USA, 2–4 March 2020; pp. 144–149. [CrossRef]
296. Lew, C.C.; Torres, C.F.; Shinde, S.; Brandenburger, M. Revisiting Rollbacks on Smart Contracts in TEE-protected Private Blockchains. In Proceedings of the 2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Vienna, Austria, 8–12 July 2024; pp. 217–224. [CrossRef]
297. Rajmohan, R.; Kumar, T.A.; Sandhya, S.G.; Hu, Y.C. R-GCN: A residual-gated recurrent unit convolution network model for anomaly detection in blockchain transactions. *Multimed. Tools Appl.* **2024**, *83*, 87527–87551. [CrossRef]
298. Shetty, M.; Tamane, S. Unveiling bitcoin network attack using deep reinforcement learning with Boltzmann exploration. *Peer-to-Peer Netw. Appl.* **2025**, *18*, 47. [CrossRef]
299. Yang, S.; Zhang, F.; Huang, K.; Chen, X.; Yang, Y.; Zhu, F. SoK: MEV Countermeasures. In Proceedings of the Workshop on Decentralized Finance and Security (DeFiSec), Salt Lake City, UT, USA, 14 April 2024; pp. 21–30. [CrossRef]
300. Masmoudi, M.; Amous, I.; Zayani, C.A.; Sèdes, F. Trust attack prevention based on Spark-blockchain in social IoT: A survey. *Int. J. Inf. Secur.* **2024**, *23*, 3179–3198. [CrossRef]
301. Wang, P.; Ding, Y.; Sun, M.; Wang, H.; Li, T.; Zhou, R.; Chen, Z.; Jing, Y. Building and Maintaining a Third-Party Library Supply Chain for Productive and Secure SGX Enclave Development. In Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), Seoul, Republic of Korea, 27 June–19 July 2020; pp. 100–109. [CrossRef]
302. Zichichi, M.; Ferretti, S.; D'Angelo, G.; Rodríguez-Doncel, V. Data governance through a multi-DLT architecture in view of the GDPR. *Clust. Comput.* **2022**, *25*, 4515–4542. [CrossRef]
303. Zhong, Z.; Zheng, Z.; Dai, H.N.; Xue, Q.; Chen, J.; Nan, Y. PrettySmart: Detecting Permission Re-delegation Vulnerability for Token Behaviors in Smart Contracts. In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering (ICSE), Lisbon, Portugal, 14–20 April 2024; pp. 1–12. [CrossRef]
304. Zhang, Q.; Zhao, Z. Distributed storage scheme for encryption speech data based on blockchain and IPFS. *J. Supercomput.* **2023**, *79*, 897–923. [CrossRef]
305. Zhang, J.; Bai, F.; Shen, T.; Gong, B.; Luo, J. Trusted Blockchain-Based Data Fingerprinting Differential-Traceability and SkipList Indexing Methods in Privacy Protection. In Proceedings of the 2022 4th Blockchain and Internet of Things Conference (BIOTC), Tokyo, Japan, 5–7 August 2022; pp. 73–83. [CrossRef]
306. Yuan, B.; Wan, J.; Wu, Y.H.; Zou, D.Q.; Jin, H. On the Security of Smart Home Systems: A Survey. *J. Comput. Sci. Technol.* **2023**, *38*, 228–247. [CrossRef]
307. Yi, X.; Zhou, Y.; Lin, Y.; Xie, B.; Chen, J.; Wang, C. Digital rights management scheme based on redactable blockchain and perceptual hash. *Peer-to-Peer Netw. Appl.* **2023**, *16*, 2630–2648. [CrossRef]
308. Yang, S.; Chen, J.; Huang, M.; Zheng, Z.; Huang, Y. Uncover the Premeditated Attacks: Detecting Exploitable Reentrancy Vulnerabilities by Identifying Attacker Contracts. In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering (ICSE), Lisbon, Portugal, 14–20 April 2024; pp. 1–12. [CrossRef]
309. Wu, J.; Wang, Y.; Wang, R.; Chen, J.; Zheng, Z. Can Neural Networks Help Smart Contract Testing? An Empirical Study. In Proceedings of the 14th Asia-Pacific Symposium on Internetware, Hangzhou, China, 21–23 October 2023; pp. 79–89. [CrossRef]
310. Wu, A.; Guo, Y.; Guo, Y. A decentralized lightweight blockchain-based authentication mechanism for Internet of Vehicles. *Peer-to-Peer Netw. Appl.* **2023**, *16*, 1340–1353. [CrossRef]
311. Sun, Y.; Wu, D.; Xue, Y.; Liu, H.; Wang, H.; Xu, Z.; Xie, X.; Liu, Y. GPTScan: Detecting Logic Vulnerabilities in Smart Contracts by Combining GPT with Program Analysis. In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering (ICSE), Lisbon, Portugal, 14–20 April 2024; pp. 1–13. [CrossRef]
312. Pailoor, S.; Chen, Y.; Wang, F.; Rodríguez, C.; Van Geffen, J.; Morton, J.; Chu, M.; Gu, B.; Feng, Y.; Dillig, I. Automated Detection of Under-Constrained Circuits in Zero-Knowledge Proofs. *Proc. ACM Program. Lang.* **2023**, *7*, 1510–1532. [CrossRef]
313. Ooi, S.K.; Ooi, C.A.; Yeap, J.A.L.; Goh, T.H. Embracing Bitcoin: Users' perceived security and trust. *Qual. Quant.* **2021**, *55*, 1219–1237. [CrossRef]
314. Oh, S.J.; Xiao, S.; Park, B.I.; Roh, T. Coping or threat? Unraveling the mechanisms enabling user acceptance of blockchain technologies. *Inf. Technol. Manag.* **2023**, *24*, 261–278. [CrossRef]
315. Ma, F.; Ren, M.; Ouyang, L.; Chen, Y.; Zhu, J.; Chen, T.; Zheng, Y.; Dai, X.; Jiang, Y.; Sun, J. Pied-Piper: Revealing the Backdoor Threats in Ethereum ERC Token Contracts. *ACM Trans. Softw. Eng. Methodol.* **2023**, *32*, 61. [CrossRef]

316. Luo, F.; Luo, R.; Chen, T.; Qiao, A.; He, Z.; Song, S.; Jiang, Y.; Li, S. SCVHunter: Smart Contract Vulnerability Detection Based on Heterogeneous Graph Attention Network. In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering (ICSE), Lisbon, Portugal, 14–20 April 2024; pp. 1–13. [CrossRef]
317. Li, W.; Jia, S.; Liu, L.; Zheng, F.; Ma, Y.; Lin, J. CryptoGo: Automatic Detection of Go Cryptographic API Misuses. In Proceedings of the 38th Annual Computer Security Applications Conference (ACSAC), Austin, TX, USA, 5–9 December 2022; pp. 318–331. [CrossRef]
318. Lewis, T.G. Digital Economy: The Economics of the Digital Economy. *Ubiquity* **2023**, *2023*, 1. [CrossRef]
319. Kong, Q.; Chen, J.; Wang, Y.; Jiang, Z.; Zheng, Z. DeFiTainter: Detecting Price Manipulation Vulnerabilities in DeFi Protocols. In Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA), Seattle, WA, USA, 17–21 July 2023; pp. 1144–1156. [CrossRef]
320. Karode, T.; Werapun, W. Robustness against fraudulent activities of a blockchain-based online review system. *Peer-to-Peer Netw. Appl.* **2022**, *15*, 92–106. [CrossRef]
321. Ipchi Sheshgelani, M.; Pashazadeh, S.; Salehpoor, P. Cooperative hybrid consensus with function optimization for blockchain. *Clust. Comput.* **2023**, *26*, 3565–3576. [CrossRef]
322. Ellul, J.; Galea, J.; Ganado, M.; Mccarthy, S.; Pace, G.J. Regulating Blockchain, DLT and Smart Contracts: A technology regulator's perspective. *ERA Forum* **2020**, *21*, 209–220. [CrossRef]
323. Chen, Y.; Sun, Z.; Gong, Z.; Hao, D. Improving Smart Contract Security with Contrastive Learning-based Vulnerability Detection. In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering (ICSE), Lisbon, Portugal, 14–20 April 2024; pp. 1–11. [CrossRef]
324. Aljabri, A.; Jemili, F.; Korbaa, O. Intrusion detection in cyber-physical system using rsa blockchain technology. *Multimed. Tools Appl.* **2023**, *83*, 48119–48140. [CrossRef]
325. Andreica, T.; Musuroi, A.; Anistoroaei, A.; Jichici, C.; Groza, B. Blockchain integration for in-vehicle CAN bus intrusion detection systems with ISO/SAE 21434 compliant reporting. *Sci. Rep.* **2024**, *14*, 8169. [CrossRef] [PubMed]
326. Bakar, A.; Zouhair, A.; En-Naimi, E.M. Review of Vulnerabilities and Countermeasures Against Sybil Attacks on Decentralized Systems Based on Machine Learning Algorithms. In Proceedings of the 6th International Conference on Networking, Intelligent Systems & Security (NISS), Larache, Morocco, 24–26 May 2023; pp. 1–5. [CrossRef]
327. Bargayary, B.; Medhi, N. Preserving flow table integrity in OpenFlow networks through smart contract. *Clust. Comput.* **2024**, *27*, 4569–4588. [CrossRef]
328. Bhan, R.; Pamula, R.; Faruki, P.; Gajrani, J. Blockchain-enabled secure and efficient data sharing scheme for trust management in healthcare smartphone network. *J. Supercomput.* **2023**, *79*, 16233–16274. [CrossRef] [PubMed]
329. Bhudia, A.; Cartwright, A.; Cartwright, E.; Hurley-Smith, D.; Hernandez-Castro, J. Game Theoretic Modelling of a Ransom and Extortion Attack on Ethereum Validators. In Proceedings of the 18th International Conference on Availability, Reliability and Security (ARES), Benevento, Italy, 29 August–1 September 2023; pp. 1–11. [CrossRef]

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

future internet

MDPI

Review

Securing Blockchain Systems: A Layer-Oriented Survey of Threats, Vulnerability Taxonomy, and Detection Methods

Mohammad Jaminur Islam [1,†], Saminur Islam [2,†], Mahmud Hossain [3], Shahid Noor [4] and S. M. Riazul Islam [5,*]

[1] Department of Computer Science, University of California, Riverside, CA 92521, USA
[2] Department of Computer Science, North Carolina State University, Raleigh, NC 27695, USA
[3] Department of Private Certificate Authority, Amazon Web Services (AWS), Herndon, VA 20171, USA
[4] Department of Computer Science, Northern Kentucky University, Highland Heights, KY 41099, USA
[5] School of Natural and Computing Sciences, University of Aberdeen, Aberdeen AB24 3FX, UK
* Correspondence: riazul.islam@abdn.ac.uk; Tel.: +44-1224-273887
† These authors contributed equally to this work.

Abstract: Blockchain technology is emerging as a pivotal framework to enhance the security of internet-based systems, especially as advancements in machine learning (ML), artificial intelligence (AI), and cyber–physical systems such as smart grids and IoT applications in healthcare continue to accelerate. Although these innovations promise significant improvements, security remains a critical challenge. Blockchain offers a secure foundation for integrating diverse technologies; however, vulnerabilities—including adversarial exploits—can undermine performance and compromise application reliability. To address these risks effectively, it is essential to comprehensively analyze the vulnerability landscape of blockchain systems. This paper contributes in two key ways. First, it presents a unique layer-based framework for analyzing and illustrating security attacks within blockchain architectures. Second, it introduces a novel taxonomy that classifies existing research on blockchain vulnerability detection. Our analysis reveals that while ML and deep learning offer promising approaches for detecting vulnerabilities, their effectiveness often depends on access to extensive and high-quality datasets. Additionally, the layer-based framework demonstrates that vulnerabilities span all layers of a blockchain system, with attacks frequently targeting the consensus process, network integrity, and smart contract code. Overall, this paper provides a comprehensive overview of blockchain security threats and detection methods, emphasizing the need for a multifaceted approach to safeguard these evolving systems.

Keywords: blockchain security; vulnerability detection; machine learning; deep learning; consensus attacks; smart contract security; cyber–physical systems

1. Introduction

Blockchain is considered one of the most popular technologies in recent times due to its inherent potential to become a source of reliability. Due to the decentralized nature and trustworthiness of blockchain, it has been widely adopted in different applications such as cryptocurrency or electronic cash, financial transactions, healthcare, insurance, IoT, manufacturing, education, etc. IBM pointed out in [1] that the significant advantage of blockchain is having no single-point failure. Hence, it is very unlikely to be influenced. However, it is not entirely immune to adversarial vulnerabilities, as adversaries are becoming smarter in parallel to new technological development. It has always been a game between adversaries and defenders. Such adversarial vulnerabilities can affect—directly or

indirectly—the integrity, availability, privacy, and confidentiality of the system. These are known as security vulnerabilities.

A series of high-profile breaches underscore these risks. For example, the theft of nearly USD 73 million in bitcoins from Bitfinex [2] and the exploitation of the Decentralized Autonomous Organization (DAO), which resulted in more than USD 60 million in lost ether [3], demonstrate that even large-scale, trusted systems are vulnerable. Similarly, an incident at Bithumb, where an employee's compromised computer led to a hacking event affecting 30,000 users and resulting in a loss of USD 870,000 in bitcoin [4], further highlights the fragility of blockchain security. These incidents collectively emphasize the need for a systematic study of the attack surface in blockchain-based systems so that vulnerabilities can be detected and prevented effectively and efficiently.

Recent studies also highlight a wide array of vulnerabilities across different layers and components of blockchain systems but often lack a systematic classification approach. For example, Mollajafari et al. [5] presented a seven-layer perspective emphasizing smart contract vulnerabilities and centralization threats. However, their discussion lacked detailed coverage of vulnerabilities in emerging blockchain ecosystems, and their approach did not comprehensively address vulnerabilities beyond smart contracts at a detailed operational level. Similarly, Dwivedi et al. [6] presented an in-depth classification of blockchain attacks, categorizing vulnerabilities and mitigations across blockchain layers. Yet, their work provided limited integration of novel attack surfaces appearing in newer blockchain platforms such as Polkadot, Solana, or Avalanche, which increasingly face distinct vulnerabilities due to different consensus mechanisms and layer interactions.

Gurjar et al. [7], Hejazi et al. [8], and Chu [9] thoroughly explored smart contract vulnerabilities and detection techniques. While comprehensive in the smart contract context, their work lacked broader integration into the overall blockchain security landscape. This narrow focus overlooked critical vulnerabilities inherent in other layers, such as network, consensus, or application layers. while recent research by Rajawat et al. [10] emphasized AI-driven approaches for blockchain security, the integration and systematic analysis of machine learning methodologies across different blockchain layers remain limited. Magalhães et al. [11] systematically reviewed current blockchain vulnerabilities but did not fully capture or specifically analyze layer-specific vulnerabilities emerging within newer blockchain ecosystems. These modern platforms introduce novel attack vectors due to unique consensus algorithms, interoperability features, and sophisticated transaction processing methods, thus necessitating a revised and expanded taxonomy to address their distinct security challenges comprehensively.

Beyond these, Verma et al. [12] proposes a federated learning framework to tackle zero-day vulnerabilities, indicating the growing role of machine learning in both defending and attacking blockchain infrastructures. While existing surveys and research have explored blockchain security issues, they often lack a systematic approach to categorizing and analyzing vulnerabilities. Many existing reviews focus primarily on trending approaches and applications within specific domains such as Internet of Things (IoT), without dedicated examination of security and privacy in decentralized federated learning (DFL) [13,14]. Similarly, surveys on security and privacy in centralized federated learning (FL) do not adequately address the unique challenges posed by DFL [13–15]. Some studies have attempted to address security issues in centralized FL through blockchain integration, but a comprehensive security analysis of DFL threats and their potential defense mechanisms remains lacking [13,14,16,17]. Additionally, while existing surveys may mention attacks on different layers of the blockchain system, they rarely provide a detailed, layer-by-layer breakdown of vulnerabilities, their impact, and corresponding countermeasures [18,19].

This lack of a systematic approach can hinder a comprehensive understanding of blockchain security threats and their potential mitigations.

In this research, we address these limitations by proposing a comprehensive, novel, layer-based analysis to systematically categorize blockchain attacks. We provide an explicitly detailed and unified taxonomy structured around all blockchain layers, not restricted merely to smart contract vulnerabilities, but extensively covering network, consensus, data, application, and infrastructure layers, offering systematic insights into vulnerabilities, attack impacts, and countermeasures. Our paper provides a detailed analysis of common blockchain attacks, including their layer information, impact, potential countermeasures, and the taxonomy of blockchain vulnerability detection. Furthermore, we propose a novel taxonomy of blockchain vulnerability detection, organized into seven distinct categories based on their methodologies and focal problems.

The motivation behind adopting a structured, layer-based classification in this research stems from the critical need to systematically address and mitigate the growing complexity and diversity of blockchain security vulnerabilities. Existing literature often lacks comprehensive coverage of vulnerabilities across different blockchain layers, particularly neglecting how these vulnerabilities interact across layers and impact overall system integrity. Moreover, emerging blockchain ecosystems introduce unique cross-layer interactions and novel attack vectors, making a unified and adaptable classification increasingly essential. Therefore, a clearly defined, layer-based taxonomy provides researchers and practitioners with a practical and systematic framework to precisely identify, analyze, and respond to vulnerabilities, thereby significantly enhancing targeted security interventions. Additionally, this approach inherently accommodates continuous updates and scalability, ensuring its relevance as blockchain technologies advance, new threats emerge, and innovative defense mechanisms evolve.

In addition to that, this paper also explores the evolving role of machine learning (ML) in both facilitating and combating blockchain security attacks. In summary, we want to answer the following concerns in this survey:

- RQ1: How can blockchain security threats be categorized and analyzed based on the specific layer of the blockchain system they target?
- RQ2: What are the common attack vectors and their potential impact on different blockchain layers? What countermeasures can be implemented to mitigate these threats?
- RQ3: What are the prominent categories of research on blockchain vulnerability detection, and how can they be systematically classified based on their methodologies and focal problems?
- RQ4: What are the strengths and weaknesses of each vulnerability detection category, and how can this comparative analysis inform the selection of appropriate tools and techniques?

The remainder of this article is structured as follows: Section 2 provides a comprehensive overview of blockchain architecture and the system's operational mechanisms. In Section 3, we introduce a unique layer-based framework to systematically categorize and analyze blockchain security attacks, presenting a detailed table that outlines common attack vectors, their target layers, potential impact, and countermeasures. This section aims to provide a clear and structured understanding of how vulnerabilities manifest and can be mitigated across different layers of a blockchain system. Section 4 presents a vulnerability detection taxonomy and provides a detailed analysis of various vulnerability detection approaches, along with a preliminary discussion on the evolving role of machine learning (ML) in both enhancing and threatening blockchain security. More

details on specific aspects of ML are presented in Section 5. Finally, Section 6 concludes the article by summarizing key findings, emphasizing the significance of the proposed taxonomy and layer-based framework, and highlighting crucial areas for future research in blockchain security.

2. Blockchain: An Overview

Blockchain technology can be implemented in various forms, such as private, public, and consortium-based systems. Despite the differences in architecture, each blockchain system consists of a fixed set of core components that collectively handle transaction processing. In this section, we describe these key components and outline the general transaction life cycle, thereby providing the foundation for our subsequent discussion on security vulnerabilities.

2.1. Key Components of Blockchain

Based on a review of existing works [20–25], we identify several core components that are common across different blockchain implementations. These include the distributed ledger, consensus algorithm, smart contract, immutable records (or transactions), and nodes.

2.1.1. Node

A node represents an individual user or computer that participates in the blockchain network. Nodes are interconnected via the Internet, and their membership criteria vary depending on whether the blockchain is private or public. Each node possesses a unique pair of private and public keys, which are essential for securely initiating and validating transactions.

2.1.2. Immutable Records (Transactions)

Transactions, or immutable records, are the fundamental units of information exchanged within a blockchain network. When a transaction is completed and added to the shared ledger, it becomes immutable, ensuring that the recorded information remains tamper-proof. This immutability is central to establishing trust among network participants.

2.1.3. Distributed Ledger

The distributed ledger, also known as the shared ledger, is the database that records all transactions across the network. Since every node has access to the ledger, it ensures transparency and consistency across the blockchain. The management and updating of this ledger are governed by smart contracts and validated through consensus mechanisms.

2.1.4. Smart Contract

Smart contracts are self-executing programs embedded within the blockchain. They define a set of rules for processing transactions and automatically enforce these rules when certain conditions are met. By validating and controlling the propagation of transactions among nodes, smart contracts play a critical role in maintaining the integrity of the ledger.

2.1.5. Consensus Algorithm

The consensus algorithm is the core mechanism that enables a decentralized network to agree on the validity of transactions. It ensures that, once a transaction is initiated, all nodes reach a majority-based agreement on its legitimacy before the transaction is permanently added to the blockchain. This process, which involves proof-of-work, proof-of-stake, or other consensus techniques, is essential for maintaining security and trust in a system without a central authority.

2.2. How Blockchain Works

In this section, we describe the typical life cycle of a transaction in a blockchain system. This overview is informed by previous studies [18,19,21,24,26–29] and highlights the stages where security breaches may occur.

2.2.1. Request Transaction

Any node within the blockchain network can initiate a transaction request. For example, in Bitcoin, one node might request to transfer a certain amount of cryptocurrency to another. Each transaction is first digitally signed with the node's private key and then broadcast across the network along with the corresponding public key. This process ensures that all nodes are informed of the pending transaction.

2.2.2. Validate Transaction

Once a transaction is broadcast, it is subject to validation by smart contracts, which enforce the predefined rules for transactions. If a transaction fails to meet these criteria, it is immediately discarded, thereby preventing any invalid data from entering the ledger.

2.2.3. Verify Transaction

Validated transactions then undergo verification through the consensus algorithm. This step involves combining the transaction with a timestamp and ensuring that it conforms to the network's consensus rules. Only transactions that pass this verification are considered eligible for inclusion in a new block.

2.2.4. Append Block and Complete

After verification, the transaction is grouped into a block. Nodes then check that the new block correctly references the hash of its predecessor and that all contained transactions are valid. If these conditions are met, the block is appended to the blockchain. Otherwise, the block is rejected, ensuring the integrity and continuity of the ledger.

This overview of blockchain components and the transaction life cycle sets the stage for a detailed analysis of the security challenges and vulnerabilities discussed in the following sections.

3. Security Threats

In this section, we discuss the security vulnerabilities that can arise in a blockchain system. Before delving into the specific security issues, it is important to understand why ensuring security in blockchain-based systems is inherently challenging.

3.1. Methodology for Attack Classification

Our approach to classifying blockchain security threats involved a targeted literature review across major academic databases, including IEEE Xplore, ACM Digital Library, SpringerLink, and Google Scholar. The review process followed three key steps:

- Article Selection: We searched for papers using terms such as "blockchain attacks", "security taxonomy", and "vulnerabilities in distributed ledgers". Studies were included if they (i) focused on blockchain-based systems, and (ii) presented or discussed a classification of security threats or attack types.
- Relevance Screening: Titles and abstracts were reviewed to filter out papers that were either not technical, not related to blockchain, or purely focused on cryptographic primitives. This helped ensure focus on architectural and systemic security threats.

- Classification Analysis: For each selected study, we examined how attacks were categorized (e.g., by type, impact, architecture layer). This comparative analysis informed the design of our own five-layer taxonomy, which maps attacks to the data, network, consensus, contract, and application layers of blockchain systems.

This structured yet flexible approach ensured that our taxonomy is informed by prior literature, while offering a more architecture-aligned and extensible classification framework.

3.2. Reviewed Security Literatures

In traditional centralized systems, components are grouped and organized so that every transaction or information exchange follows a defined route dictated by established policies or security controls. This arrangement helps minimize the potential scope of exploitation. In contrast, blockchain-based decentralized architectures lack well-defined boundaries among components, making it difficult to design and implement robust measures against adversarial manipulation. Furthermore, as noted in [18], authentication is the primary security control in blockchain systems, requiring open access to network nodes (or users) for autonomous operation. This structure limits the implementation of additional security measures beyond strong, cryptography-based authentication.

Security vulnerability analysis is a critical area of blockchain research, with various studies offering different perspectives on potential attacks. For example, the work in [30] identifies six different types of attacks. In [27], blockchain security threats are broadly classified into four types, discussing eleven distinct attacks. Similarly, Dasgupta et al. [31] groups eighteen different attacks into eight categories, while [32] divides sixteen attack types into seven classes. In [33], the authors identify nine classes covering twenty-four different attacks, and Saad et al. [28] categorizes blockchain attacks into three broad types encompassing twenty-two attacks. Moreover, Lee at al. [18] documents twenty-three different types of attack, and Mosakheil et al. [29] discusses five different security threat types, under which thirty attack incidents are identified.

Together, these studies underscore the complexity and diversity of security vulnerabilities in blockchain systems and highlight the pressing need for systematic analysis and effective countermeasures. Table 1 provides a summary of attack classifications by different works.

3.3. Layer-Based Attack Classification

In Figure 1, a layer-wise decomposition of the blockchain network and the primary vulnerabilities are shown on the respective sides. Later, Table 2 provides a summary of different kinds of attacks on different layers, and their countermeasures are shown.

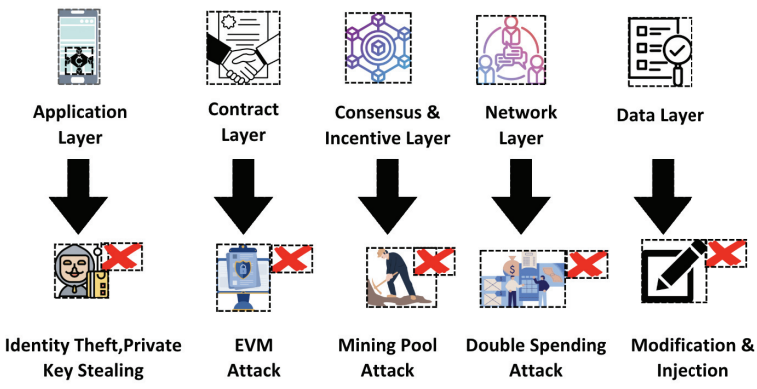


Figure 1. Layer-based classification of blockchain attacks, highlighting representative vulnerabilities at each layer (application, contract, consensus and incentive, network, and data).

Table 1. Comparison of our layer-based taxonomy with existing classifications.

Study	Taxonomy Type	Categories	Key Features/Limitations
Bhushan et al. (2021) [27]	Threat-based	4	General classification of threats; lacks architecture-specific alignment.
Saad et al. (2019) [28]	Surface-based	3	Abstract attack surfaces; does not directly align with blockchain system layers.
Mosakheil (2018) [29]	Impact-based	5	Categorizes by severity and consequences; limited structural insight.
Dasgupta et al. (2019) [31]	Functional-based	8	Detailed, but mixes threat types and system components.
Anita et al. (2019) [32]	Type-based	7	Catalog of known attacks; lacks architectural grouping.
Lee (2019) [18]	System-component	6	Focuses on blockchain node internals; limited generalizability.
Mollajafari & Bechkoum (2023) [5]	Layer-based	7	Strong emphasis on smart contract centralization risks; contract layer focus.
Gao et al. (2022) [34]	Consensus-attack	6	Comprehensive categorization of consensus-specific threats; does not cover full system layers.
Cheng et al. (2021) [35]	Threat-centric layered	5	Maps attacks to technical stack; lacks emphasis on adaptability across architectures.
Our Work	**Layer-based**	**5**	Scalable, architecture-aligned mapping of attacks to blockchain layers; supports DAG, BFT, sharded, hybrid chains.

Note: The last row is highlighted in bold to emphasize our proposed taxonomy and distinguish it from existing classifications.

To systematically assign attacks to blockchain layers, we established explicit classification criteria based on each attack's primary mechanism and its point of impact within the blockchain architecture. This structured approach aligns vulnerabilities clearly with the specific blockchain layer they affect, enabling targeted and efficient mitigation strategies. By understanding attacks within this layered context, security researchers and blockchain developers can more effectively implement defense mechanisms that address the root architectural causes of vulnerabilities. Specifically, attacks targeting transaction structures, cryptographic integrity, or ledger consistency, such as the transaction malleability attack, where an adversary manipulates transaction IDs by altering digital signatures, are categorized within the **Data Layer** [36]. Attacks disrupting node-to-node communication, routing protocols, or network connectivity, such as the Sybil attack, which involves creating numerous fake identities to disproportionately influence network decisions, are classified under the **Network Layer** [37]. Vulnerabilities compromising consensus algorithms or incentive mechanisms designed to regulate miner or validator behavior are assigned to the **Consensus & Incentive Layer**; a prominent example includes the 51% attack, wherein an attacker seeks majority control over the network's mining resources to manipulate blockchain state. Smart contract-specific vulnerabilities arising from programming flaws or execution logic errors, such as the re-entrancy attack, involving repeated calls to vulnerable functions before completion of their initial execution, are classified into the **Contract Layer** [5]. Lastly, vulnerabilities exploiting end-user interfaces, wallet security weaknesses, or external software applications such as cryptojacking, where attackers illicitly utilize

computing resources to mine cryptocurrencies, are grouped under the **Application Layer**. This comprehensive, layer-centric approach enhances clarity, facilitates reproducibility, and supports the deployment of precise and effective blockchain security measures.

From Table 3, we can infer that while our taxonomy is designed around traditional chain-based blockchains (e.g., Bitcoin, Ethereum), it can be extended to accommodate other architectures such as DAG-based systems (e.g., IOTA [38], Hedera), hybrid blockchains (e.g., Hyperledger Fabric [39]), sharded blockchains (e.g., Ethereum 2.0 [40]), and BFT-based platforms (e.g., Cosmos, Tendermint [41]).

Table 2. Summary of different attacks on blockchain.

Attack Name	Layer	Impact	Counter Measure
Transaction Malleability	Data	Network reliability and Integrity.	Hashing of the transactions, Time commitment scheme.
Quantum	Data	Cryptographic algorithm.	Quantum resistant key Scheme.
Race	Network	Double spending, System integrity.	Enforcing multiple transaction confirmation.
Finney	Network	Double spending, Integrity, Trust and reliability.	Increased the required number of Block confirmations for transactions.
Replay	Network	Transaction duplication, Double spending, Network congestion.	Transaction sequencing techniques, Digital signatures.
Eclipse	Network	Network partitioning, Isolation, Forking and reorganization.	Detection through monitoring, Introducing peer discovery protocols.
DDoS	Network	Service disruption, unavailability, Network congestion, and performance degradation.	Traffic filtering, Rate limiting, IP blacklisting.
BGP Hijacking	Network	Routing manipulation, Network isolation, Dropped transactions.	BGP monitoring, Relay network.
Routing	Network	Network partitioning, Delay in transactions.	Audit, Verification, Cryptography.
DNS	Network	Service disruption and unavailability, Phishing, and Information leakage.	Improved authentication Scheme, Monitoring DNS traffic.
51%	Consensus	Double spending, Network disruption.	Diversify mining pool, Increased hash rate, improved consensus algorithms.
Time-jacking	Consensus	Consensus disruption, Undermined integrity, Trust, and Reliability.	Enhanced timestamp validation.
Sybil	Network	Identity, Integrity, Consensus, Double spending.	Robust identity verification such as PoW, PoS.

Table 2. *Cont.*

Attack Name	Layer	Impact	Counter Measure
Vector 76	Consensus	Double spending, Loss of trust	Multiple confirmation, Improved consensus.
Selfish Mining	Consensus	Mining pool, Double spending, and centralization.	Fork resolution scheme, Rational mining scheme.
Block Withholding	Consensus	Disruption of consensus, Centralization, Reduced transaction speed.	Zeroblock algorithm, Pool incentives.
Bribery	Consensus	Integrity, Reduced confidence, Trust Pool.	Improved consensus such as PoS and strong incentives.
Re-entrancy	Contract	Wallet theft, Disruption of application.	Security Audits, Secure coding practices, Checks-effects-interactions (CEI) pattern.
Integer Overflow	Contract	Unintended token transfer, DoS, Smart contract failure.	Careful smart contract design, Symbolic execution for detection.
Short Address	Contract	Validation error	Automatic sanitization techniques.
Balance	Contract	Double Spending, Reduced Network Efficiency.	Fast block confirmation, Monitoring.
Cryptojacking	Application	Performance degradation, Increased operational cost.	Detection software, Network monitoring.
Cold wallet	Application	Loss of funds.	Careful Smart Contract design, Multifactor authentication.
Hot wallet	Application	Loss of funds and trust.	Regular software upgrade, Multifactor authentication.
Dictionary	Application	Private key compromise, Unauthorized access and control.	Strong password, Hardware wallet.
Malware	Application	Private key compromise, Network integrity.	Usage of antimalware or detection software.

Table 3. Applicability of layer-based taxonomy to blockchain architectures.

Architecture	Applies	Adjustments	Remarks
Chain-based (Bitcoin, Ethereum)	Yes	No	Fully compatible with all five layers.
DAG-based (IOTA [38], Hedera)	Yes	Minor	Non-linear consensus, but threats map to Network & Consensus layers.
Hybrid (Hyperledger Fabric [39])	Yes	Yes	No token layer; chaincode maps to Contract. Rename Consensus & Incentive.
Sharded (Ethereum 2.0 [40])	Yes	Yes	Cross-shard comm. affects Network/ Consensus layers.
BFT-based (Cosmos, Tendermint [41])	Yes	Minor	Consensus layer must model validator-specific attacks.

The core architectural concepts—data persistence, peer communication, consensus coordination, contract logic, and user-facing applications—are present in all these systems,

albeit with architectural nuances. DAGs remove block serialization but retain vulnerabilities across the network and consensus layers. Hyperledger lacks incentive mechanisms but retains contract-level logic via chaincode. Sharded platforms require coordinated consensus across shards and routing mechanisms, while BFT-based protocols must address equivocation and round-skipping attacks. Despite these differences, the five-layer taxonomy remains applicable with only minor adjustments to consensus or contract layers, validating its utility across diverse decentralized architectures.

4. Security Attacks

In this section, we present some of the well-known attacks discovered per layer from our study focusing on security issues.

4.1. Data Layer

In this section, we discuss several security vulnerabilities that can arise at the data layer of blockchain systems. These vulnerabilities exploit weaknesses in transaction handling and cryptographic mechanisms, often resulting in double-spending or unauthorized manipulation of funds.

4.1.1. Transaction Malleability Attack

In a transaction malleability attack [18,27,29,31,42], the adversary exploits a design flaw where certain non-essential parts of a transaction can be modified without changing its underlying intent. This manipulation results in a different transaction ID, leading the sender to believe that the original transaction has failed or not been confirmed on the network. Consequently, the sender may inadvertently reissue the same transaction, effectively causing the funds to be transferred twice.

Figure 2 illustrates the process of a transaction malleability attack, which targets the integrity of transaction identifiers in a blockchain network. The attack begins with a sender node (top left) that initiates a legitimate transaction, generating an original transaction ID. Meanwhile, an attacker node (top right) intercepts this transaction as it is being broadcast to the network. The attacker slightly alters the transaction's structure, such as by modifying a non-critical field, which changes its ID while keeping the core transaction valid. This results in an altered transaction ID, shown in the right branch. Both the original and the modified versions may reach the blockchain network at the same time, causing confusion over which transaction is valid. If the network accepts the altered version and discards the original, the sender may believe their payment did not go through and issue a second transaction. This leads to a double-spending scenario, where the same funds are spent more than once. The figure clearly shows how a small change introduced by an attacker at the transaction level can mislead the network and compromise the integrity of the ledger.

If the network confirms the modified transaction before the original, the sender ends up paying twice the intended amount. Historically, this vulnerability played a significant role in the collapse of Mt. Gox in 2014, a major Bitcoin exchange that controlled approximately 70% of the market at the time. Although the exact impact of malleability on the Mt. Gox incident is subject to debate, it is widely cited as a contributing factor. Modern blockchain implementations, such as the upgraded Bitcoin protocols with Segregated Witness, have largely mitigated this risk, though the process can still introduce delays in transaction confirmations.

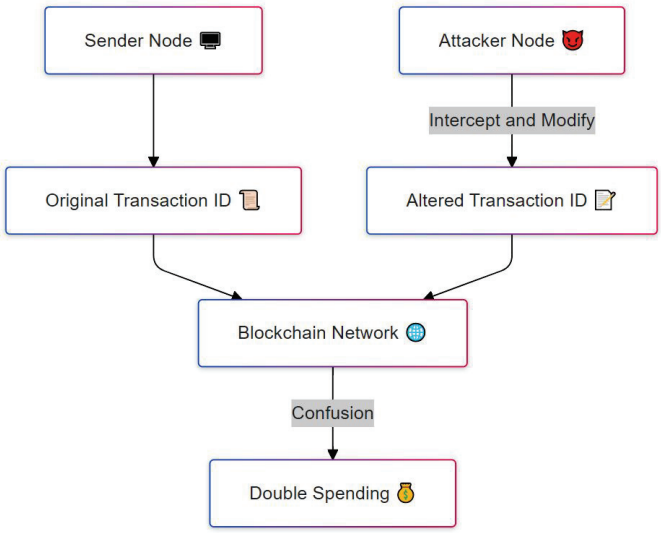


Figure 2. Transaction malleability attack where the transaction ID is modified to enable double-spending.

4.1.2. Quantum Attack

Blockchain systems currently depend on cryptographic schemes—such as RSA and elliptic curve (EC) cryptography—to secure transactions and generate private–public key pairs. However, advances in quantum computing have raised concerns about the future security of these systems. Quantum attacks [42] refer to the theoretical capability of quantum computers to break conventional cryptographic algorithms, potentially rendering these security measures ineffective. Although practical quantum attacks have not yet materialized, the substantial computational power of quantum systems suggests that no cryptography-based system can be considered entirely secure in the long term.

For instance, researchers estimate that a classical computer with a 5 GHz CPU would require approximately 13.7 billion years [43] to break a 2048-bit RSA cipher—the current standard for secure encryption. In stark contrast, a quantum computer operating at 10 MHz could potentially break the same cipher in roughly 42 min. This dramatic difference underscores the significant threat that quantum computing poses to blockchain security.

To address this issue, the development and implementation of quantum-resistant key generation techniques have become a priority [44]. These techniques are designed to withstand attacks from both classical and quantum computers, thereby ensuring the long-term security and integrity of blockchain systems.

4.2. Network Layer

The network layer plays a critical role in the functioning of blockchain systems by enabling communication among nodes and facilitating the propagation of transactions and blocks. However, the open and decentralized nature of the network also makes it susceptible to a variety of attacks. In this section, we discuss several common network-layer attacks, their mechanisms, and possible countermeasures.

4.2.1. Sybil Attack

A Sybil attack [18,27,29,31,32,42] occurs when an adversary creates multiple fake identities (or nodes) within the network. By doing so, the attacker can gain disproportionate influence over the network, disrupting consensus and manipulating the flow of information. To prevent Sybil attacks, robust identity management and reputation systems are essential.

These systems help detect and mitigate the presence of fake nodes, ensuring that only legitimate participants contribute to the network.

4.2.2. Eclipse Attack

An eclipse attack [27,29,30,32,42] aims to isolate a targeted node by monopolizing its peer connections. In this attack, the adversary gains control over a large number of IP addresses and replaces the legitimate entries in the target node's address table. When the target node restarts, it connects exclusively to these malicious nodes, effectively isolating it from the rest of the network. As a result, the attacker can feed the victim false blockchain data and manipulate its view of the network.

Figure 3 illustrates the process of an eclipse attack, a type of network-layer threat in blockchain systems where an attacker isolates a victim node from the rest of the network. The attack begins with malicious nodes that aim to cut off a victim node from the legitimate peer-to-peer network (top left). This is done by disconnecting the victim from honest peers, leading to the isolation of the victim (middle left). Once isolated, the attacker surrounds the victim with only malicious nodes and begins to send fake information (top right). Since the victim is now cut off from trustworthy sources, it relies entirely on the attacker-controlled nodes for blockchain data. As a result, the victim node is misled into accepting fake blockchain information, such as false transaction histories or manipulated block data. This can have serious consequences, including enabling double-spending or censoring transactions. The diagram clearly shows how isolation and misinformation are used together to compromise the victim's view of the blockchain.

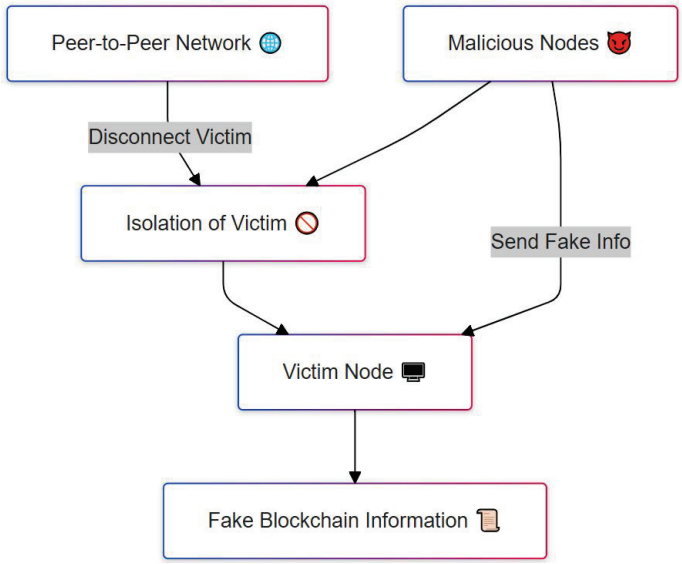


Figure 3. Eclipse attack isolating a victim node and feeding it false blockchain data.

4.2.3. DDoS Attack

Distributed denial of service (DDoS) attacks [18,29,31,32,42] target the availability of blockchain systems by overwhelming network resources. In these attacks, an adversary leverages a network of compromised devices (a botnet) to flood the network with an excessive number of requests. This flood exhausts processing power and bandwidth, rendering the system unable to service legitimate traffic. High-profile incidents, such as those experienced by Bitfinex in 2017 and 2020, demonstrate the disruptive potential of DDoS attacks. Effective countermeasures include traffic filtering, rate limiting, and enhancements in network resilience and scalability to manage sudden spikes in traffic.

4.2.4. BGP Hijacking Attack

BGP hijacking attacks [18,30,42] exploit vulnerabilities in the border gateway protocol (BGP) to manipulate network-level traffic. In such an attack, an adversary alters the routing information so that network traffic is rerouted through malicious channels. This can delay the propagation of messages, adversely affecting mining operations, or even enable the theft of cryptocurrency. To mitigate BGP hijacking, network operators can implement BGP monitoring, route origin validation, and encrypted communication channels to safeguard routing information.

4.2.5. Routing Attack

Routing attacks [29,42] can occur in two primary forms: network partitioning and message delay. In a partitioning attack, adversaries split the network into disjoint segments, which may lead to the creation of parallel blockchains. Once the attack ceases, the smaller chain is typically discarded, resulting in significant loss of mining power and revenue. In delay attacks, adversaries tamper with the propagation of messages, causing blocks to reach victim nodes later than expected. This delay can facilitate double-spending or lead to wasted mining resources. Due to their stealthy nature, these attacks are challenging to detect. Preventative measures include using secure communication protocols and network segmentation to minimize the risk of such routing attacks.

4.2.6. DNS Attack

DNS attacks [18] target the domain name system (DNS) to compromise blockchain nodes, particularly new ones. In this scenario, an adversary manipulates the DNS cache by injecting false node information. As a result, a new node, during its bootstrapping process, may connect to a malicious set of peers instead of legitimate nodes. This false neighborhood can be exploited to manipulate transactions or the consensus process. Common defenses against DNS attacks include secure bootstrapping mechanisms, periodic DNS cache flushing, and robust peer verification protocols to ensure that connections are only established with trusted nodes.

4.3. Consensus and Incentive Layer

The consensus and incentive layer is crucial for maintaining the integrity and security of blockchain systems. It not only governs how transactions are agreed upon and appended to the ledger but also ensures that miners and validators are appropriately rewarded. However, this layer is vulnerable to several sophisticated attacks that exploit the incentives and consensus mechanisms. In the following subsections, we describe a range of attacks targeting this layer, explain their mechanisms, and discuss potential countermeasures.

4.3.1. 51% Majority Attacks

In a 51% majority attack [18,27,29,32,42,45], an adversary gains control of more than half of the network's mining power. With this majority control, the attacker can determine which transactions are included in new blocks and which are excluded. This control enables the adversary to reverse previously confirmed transactions (facilitating double-spending) or prevent other miners from validating transactions and creating new blocks, effectively halting network operations.

Figure 4 depicts the sequence of actions in a 51% attack, also known as a majority attack, which occurs when a single entity or group gains control of more than half of the total computational power in a blockchain network. The diagram starts at the top with the blockchain network, which typically consists of many participants, including honest miners who follow the rules. However, if a majority attacker (shown on the right) gains control of over 50% of

the mining power, they can create new blocks faster than the honest miners. This allows them to initiate a chain reorganization, replacing the legitimate blockchain history with their own version. As shown in the lower part of the diagram, this manipulation can lead to serious consequences: the attacker can exclude certain transactions from being recorded (left path) or perform double spending (right path), where the same cryptocurrency is spent multiple times. The image highlights how control over block creation enables manipulation of the ledger, undermining the core trust model of blockchain. A notable real-world example is the Ethereum Classic (ETC) attack in August 2020, where approximately $5.6 million worth of ETC was double-spent during a chain reorganization event. To mitigate such attacks, increasing the overall network hash rate makes it more difficult for an adversary to accumulate the required mining power. Improvements in consensus algorithms and continuous network monitoring also serve as effective countermeasures.

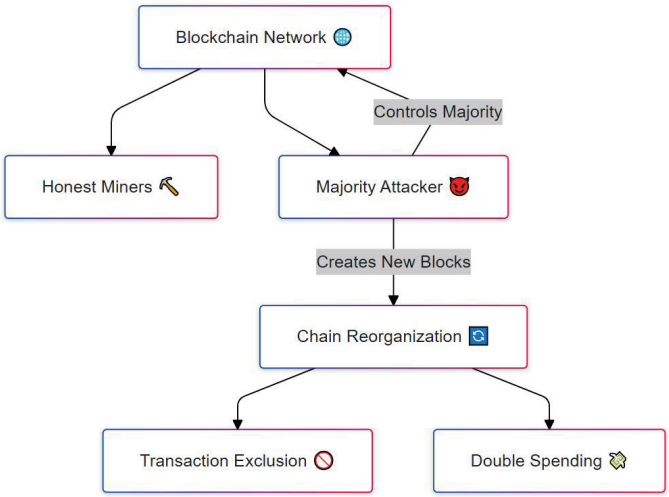


Figure 4. Attack where the attacker controls 51% of mining power, manipulating transactions.

4.3.2. Replay Attacks

Replay attacks [28,31,32,46] primarily occur during hard forks, when the blockchain splits into two distinct versions, one following the older protocol and the other adopting the upgraded protocol. In this scenario, a transaction that was valid before the fork remains valid on both chains. A malicious attacker can exploit this situation by broadcasting a pre-fork transaction on both chains, effectively causing the same funds to be transferred to the attacker's wallet on one chain while duplicating the transaction effect. Although hard forks are relatively infrequent, the potential financial losses resulting from replay attacks can be substantial.

To mitigate these risks, several countermeasures have been proposed. For example, transactions can be marked so that they are only valid on one specific chain; timestamps can be incorporated to prevent the reuse of transactions; and session keys can be employed to add an additional layer of security.

4.3.3. Race Attack

A race attack [27,29,31,42] exploits the time gap between the initiation of a transaction and its subsequent confirmation. In this attack, an adversary creates two transactions using the same funds: one legitimate and one fraudulent. The attacker leverages the delay in network confirmation so that if the fraudulent transaction is confirmed before the valid one, the attacker effectively spends the same funds twice. Initially, the valid transaction is

broadcast to build trust within the target network, but the attacker later exploits this trust by rapidly broadcasting the conflicting transaction.

To mitigate race attacks, it is crucial to enforce a sufficient waiting period for transaction confirmations. Additionally, robust consensus mechanisms—such as Proof of Work or Proof of Stake—can further reduce the risk by ensuring the integrity of the ledger.

4.3.4. Finney Attack

A Finney attack [27–29,31,42] is a variant of the race attack that requires the attacker to control significant mining power. In this scenario, the attacker pre-mines a block containing a double-spending transaction but withholds the block from immediate broadcast. Instead, the attacker first uses the funds from the unbroadcast transaction to make a purchase from a merchant. After the goods or services have been received, the attacker then releases the pre-mined block, which includes the conflicting transaction, thereby effectively reversing the original payment. This method not only relies on network propagation delays but also on the attacker's ability to manipulate block creation and confirmation timing. As with race attacks, increasing the waiting time for multiple confirmations before finalizing transactions can help mitigate the risk of Finney attacks.

4.3.5. Time Jacking

In a time jacking attack [28,29,31], an adversary manipulates the perceived network time of a target node. This is typically achieved by having the target node connect to multiple malicious peers that intentionally report falsified timestamps. As a consequence, the target node adopts these inaccurate timestamps, which then influence its block validation process. When the node's internal clock is skewed, it may reject otherwise valid new blocks that appear to have timestamps outside a predetermined acceptable range, thereby disrupting the consensus mechanism.

To mitigate time jacking attacks, several countermeasures can be implemented. These include using the node's system time as a trusted reference or enforcing strict limits on the acceptable deviation of incoming timestamps. In addition, securing time synchronization through network encryption, randomizing peer connections to reduce prolonged exposure to malicious nodes, increasing the diversity of the peer network, and monitoring round-trip times for anomalies can collectively enhance resilience against such attacks.

4.3.6. Vector 76 Attack

In the Vector 76 attack [27,31,42], an adversary exploits a double-spending opportunity using two separate nodes. One node connects to the service (e.g., an E-Wallet), while the other is connected to the broader blockchain network. The attacker initiates two transactions—one with a high value and another with a low value. The high-value transaction is directed to the service node to create a false impression of a substantial deposit. Meanwhile, the attacker continues mining until a target transaction is detected and confirmed on the blockchain. At that moment, the attacker withdraws the high-value amount from the service node while broadcasting the low-value transaction to the network. As a result, the network accepts the low-value transaction, invalidating the high-value one. To mitigate this attack, blockchain networks can reduce transaction confirmation times, adopt consensus mechanisms less susceptible to this attack (e.g., Proof of Stake), and enforce stricter mempool validation rules to prevent unauthorized transactions from entering the system.

4.3.7. Selfish Mining

Selfish mining [27,30,32,42] occurs when an adversarial miner withholds mined blocks from the public network. Instead of immediately broadcasting a newly mined block, the selfish miner keeps it private, extending a private chain. The attacker's goal is to strategically release the withheld blocks to force honest miners to work on an outdated chain. This causes regular miners to waste their computational resources, while the selfish miner reaps disproportionate rewards.

Figure 5 illustrates the workflow of a selfish mining attack, a strategy where a malicious miner manipulates the block publication process to gain more rewards than they would fairly earn. The attack begins with a selfish miner (top box) who chooses to mine blocks privately instead of immediately broadcasting them to the network. These withheld blocks are kept secret, giving the selfish miner a temporary lead over honest miners, who are unaware of the hidden progress. At a carefully chosen moment, the attacker strategically releases the withheld blocks, shown as broadcasted blocks, to the blockchain network.

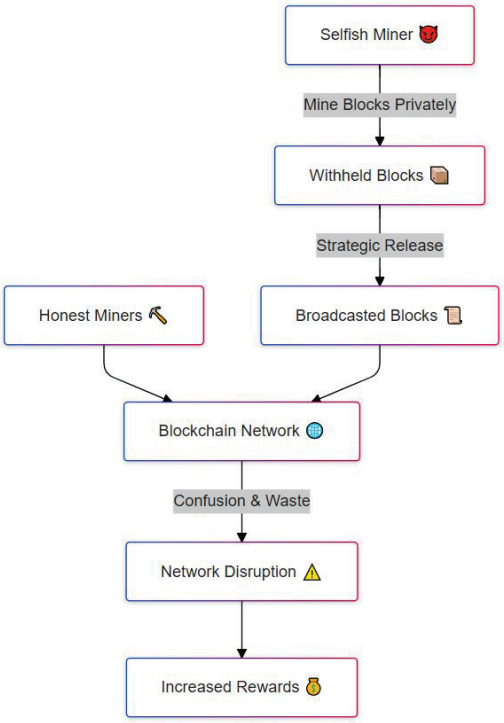


Figure 5. Selfish mining attack where a miner withholds blocks to waste resources and gain additional rewards.

At this point, the blockchain network, now receiving conflicting information from both honest and selfish miners, experiences confusion and waste. Honest miners may have unknowingly mined on a shorter chain, causing their work to be discarded when the longer withheld chain is revealed. This results in network disruption (shown with a warning symbol), which ultimately benefits the selfish miner. By manipulating the timing of block releases, the attacker can force honest miners to waste computational resources and thereby secure increased rewards for themselves, gaining an unfair advantage without necessarily controlling the majority of the network's hash power.

4.3.8. Block-Withholding Attack (BWH)

Block withholding is a variant of selfish mining [27,29] where an attacker intentionally withholds valid blocks from the network. In this attack, a miner who discovers a valid block

refrains from broadcasting it, instead keeping it secret to gain a competitive advantage. By continuing to mine privately, the attacker can eventually force other miners to work on an outdated or incomplete blockchain, which leads to an increase in stale (or orphaned) blocks. This behavior not only disrupts the fair reward distribution among miners but also threatens the overall decentralization of the network. To counteract block withholding attacks, blockchain systems can implement protocols that incentivize timely block propagation and penalize miners who exhibit selfish behavior.

4.3.9. Bribery Attack

Bribery attacks [29] involves an adversary attempting to influence the blockchain by manipulating transaction confirmations. In this scenario, the attacker first creates a legitimate transaction that becomes confirmed after a certain number of confirmations, ensuring it is irreversible. Then, the attacker initiates a conflicting fraudulent transaction and constructs a new, fraudulent block. By bribing or renting a significant portion of the network's mining power, the attacker extends the fraudulent branch until it becomes longer than the legitimate one. Once the fraudulent chain overtakes the original, the conflicting transaction becomes valid, effectively reversing the original transaction. Preventing bribery attacks is challenging, as they often involve ethical and governance considerations. However, enhancing decentralization and ensuring that decision-making is distributed can reduce the influence of any single participant.

4.4. Contract Layer

The contract layer is where smart contracts reside and execute on blockchain platforms, such as Ethereum. These self-executing programs manage digital assets, enforce agreements, and facilitate transactions without intermediaries. However, vulnerabilities in smart contracts can be exploited by attackers to disrupt contract behavior, steal funds, or manipulate contract state. In the following subsections, we discuss common contract layer attacks, their mechanisms, and potential countermeasures.

4.4.1. Re-Entrancy Attack

A re-entrancy attack [18,30,42,47] occurs when a smart contract's function is called recursively before the initial invocation completes its execution. In this scenario, an attacker repeatedly calls a vulnerable function, manipulating the contract's state and draining funds before the original transaction has been finalized. The primary impact of a re-entrancy attack is the unauthorized withdrawal of funds or assets stored within the contract, along with unintended modifications to critical contract parameters.

Figure 6 visually illustrates the sequence of actions in a re-entrancy attack, using a step-by-step flow. It begins with a malicious attacker (top box) who initiates the process by calling a vulnerable smart contract. This contract contains a flaw: it does not update its internal state (such as the user's balance) before sending out funds. Once the initial function call is triggered (center of the diagram), the contract sends funds to the attacker's external address or contract. However, instead of completing the transaction, the attacker's contract calls back into the vulnerable contract before it finishes execution. This triggers a re-entrant call, shown on the left branch, which reactivates the same withdrawal logic repeatedly. Since the contract has not yet marked the initial transaction as complete, it continues to send funds, resulting in funds being drained, as shown on the right side of the flow. The image clearly shows how the recursive loop is formed and how control flow returns to the vulnerable contract multiple times, emphasizing the core vulnerability. This

attack exploits the contract's failure to follow safe design patterns, such as updating state before making external calls.

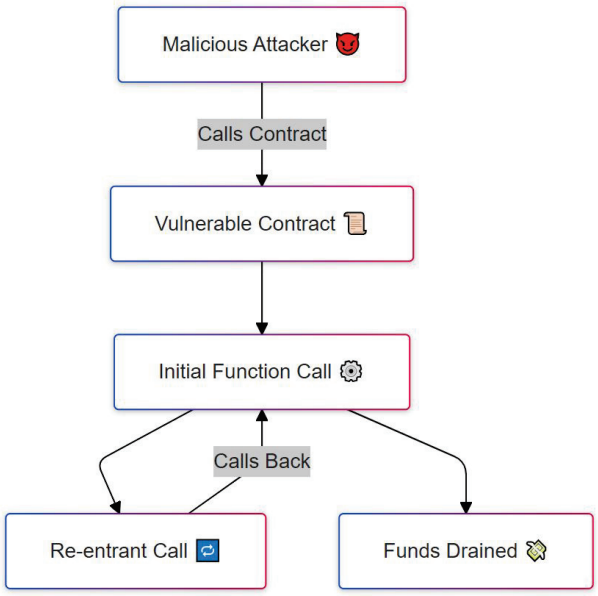


Figure 6. Re-entrancy attack where a contract is repeatedly called to drain funds.

4.4.2. Overflow Attacks

Overflow attacks exploit arithmetic vulnerabilities in smart contracts [28,48]. These occur when a variable exceeds its maximum or falls below its minimum value, causing it to "wrap around". For example, a 32-bit integer can only represent values from −2,147,483,648 to 2,147,483,647. In a smart contract for online betting, if a user sends an amount that exceeds this limit, the resulting value may wrap around to 0, leading to unexpected behavior. A notable case of an overflow attack occurred in 2018 on the Beauty Chain smart contract, where an attacker exploited this vulnerability to bypass verification checks and withdraw tokens exceeding the contract balance [49]. Employing safe arithmetic libraries and adhering to secure programming practices in Solidity can help prevent such attacks.

4.4.3. Short Address Attacks

Short address attacks [28,48] target vulnerabilities in Ethereum's virtual machine, particularly affecting ERC20 tokens. In this attack, an adversary creates an Ethereum wallet with an address ending in zeroes and then manipulates the address by removing the trailing zeroes during a transaction. If the contract does not verify the sender's address length correctly, the Ethereum Virtual Machine (EVM) automatically pads the address with missing zeroes. This discrepancy can lead to incorrect token transfers. For instance, an attacker might exploit this flaw such that for every 1000 tokens purchased, the contract erroneously credits 256,000 tokens. Ensuring strict input validation and address length verification can mitigate this risk.

4.4.4. Balance Attack

The balance attack [29] targets PoW-based blockchains by disrupting communication between subgroups of miners with similar mining power. In this attack, an adversary introduces delays between these subgroups—one subgroup (the transaction subgroup) processes transactions, while the attacker mines blocks in another subgroup (the block subgroup). By strategically delaying communication, the attacker ensures that the block subgroup's chain becomes longer than the transaction subgroup's chain. Consequently,

even if a transaction is committed, the attacker can override or remove the block containing that transaction. The balance attack highlights that PoW-based blockchains may be "block oblivious", where there is always a probability that an adversary can override a transaction-containing block. Implementing protocols that enhance communication and synchronization among mining nodes can help reduce the likelihood of such attacks.

4.5. Application Layer

The application layer represents the interface through which users interact with blockchain systems. This layer comprises software applications, wallets, and services that manage digital assets and facilitate transactions. Due to its direct interaction with end users, the application layer is a prime target for various attacks. In the following subsections, we outline common attacks on both cold and hot wallets, as well as other application-level threats, and discuss potential countermeasures.

4.5.1. Cryptojacking

Cryptojacking [28,31] is an attack targeting web and cloud-based services to illicitly perform Proof of Work (PoW) calculations for cryptocurrency mining without the user's consent. The most prevalent form of cryptojacking today is in-browser cryptojacking, where compromised websites turn visitors' devices into mining nodes. PoW requires intensive computational resources to solve complex mathematical problems. As the overall network hash rate increases, the mining difficulty also rises, leading to the use of specialized hardware such as GPUs and ASICs in professional mining operations. Cryptojacking can burden unsuspecting users by draining device resources and increasing energy consumption. To combat cryptojacking, it is essential to implement robust security measures on web platforms, such as script blocking, frequent security audits, and user awareness programs regarding unauthorized mining activities.

4.5.2. Attacks on Cold Wallets

Cold wallets, also known as blockchain hardware wallets [27,31], store private keys offline and are generally considered more secure than their online counterparts. However, these wallets can still be compromised through sophisticated methods such as bug implantation. For example, an Evil Maid attack involves an adversary physically accessing a device and implanting malicious software that extracts private keys, PINs, recovery seeds, and passphrases from the wallet. In one notable incident in 2019, during a routine transfer of funds to a cold wallet at the UPbit cryptocurrency exchange, attackers exploited timing vulnerabilities to steal $342,000 worth of ETH. This incident highlights that even offline storage solutions can be vulnerable if physical or procedural security measures are inadequate.

4.5.3. Attacks on Hot Wallets

Hot wallets [27,31] are internet-connected applications that store private cryptographic keys. Although cryptocurrency exchanges often claim to keep user data in wallets isolated from the internet, history has shown otherwise. For instance, a $500 million attack on Coincheck in 2018 exposed significant vulnerabilities in hot wallet security. In June 2019, an attack on GateHub resulted in unauthorized access to dozens of native XRP wallets and the theft of various crypto assets. Similarly, the Singapore-based crypto exchange Bitrue experienced a hot wallet breach around the same period, with attackers stealing funds worth over $4.5 million in XRP and $237,500 in ADA. These incidents underscore the importance of robust security measures and constant monitoring for online wallets.

4.5.4. Dictionary Attack

A dictionary attack [50] involves an adversary systematically attempting to guess a target node's password by trying a large number of likely possibilities. This attack becomes feasible when the target employs weak or easily guessable passwords. The effectiveness of such an attack is directly related to the randomness and strength of the hash generator used to secure the password. To mitigate dictionary attacks, it is essential to enforce strong password policies and implement multifactor authentication schemes, which add additional layers of security beyond just a password.

4.5.5. Malware Attack

Malware attacks [31] on the application layer are designed to stealthily compromise the target device and potentially its connected network. In a blockchain context, malware can hijack a device's computational resources to perform unauthorized tasks, such as mining blocks in the background without the user's consent. Ransomware attacks are a common variant, wherein the malware locks the target out of its own system resources until a ransom is paid. These attacks not only disrupt normal device operations but can also lead to significant financial losses if critical assets or credentials are compromised. Implementing robust anti-malware solutions and maintaining up-to-date security practices are key measures in defending against such threats.

In summary, we have examined a wide range of attacks spanning multiple layers of blockchain systems. These include network-level exploits—such as 51% majority attacks, Sybil, eclipse, DDoS, BGP hijacking, routing, and DNS attacks—as well as consensus and contract layer threats such as vector 76, selfish mining, block withholding, bribery, reentrancy, overflow, and short address attacks. We have also discussed application-level vulnerabilities that target both cold and hot wallets. For a quick overview of these attacks, please refer to Table 2. This comprehensive analysis highlights the multifaceted security challenges inherent in blockchain technology. With a solid understanding of these attack vectors, we now shift our focus to vulnerability detection. The following section will explore state-of-the-art techniques and emerging approaches designed to identify, analyze, and mitigate these vulnerabilities, ultimately strengthening the overall security posture of blockchain systems.

5. Vulnerability Detection

In recent years, the rapid development of blockchain technologies and decentralized applications has highlighted the critical importance of smart contract security [51]. Researchers have proposed various methods to detect and mitigate vulnerabilities in smart contracts, each leveraging different techniques and addressing unique challenges. To systematically address these security concerns, we have conducted a comprehensive review and developed a structured taxonomy that categorizes existing literature into clearly defined methodological clusters.

5.1. A Systematic Taxonomy of Vulnerability Detection

Our taxonomy provides a coherent framework, allowing a nuanced comparison of methods and identification of gaps in current research. In Figure 7, we present a systematic taxonomy that classifies research papers focused on blockchain vulnerability detection. This taxonomy was developed through a structured process that involved several key steps, ensuring a comprehensive and accurate categorization of the state-of-the-art literature in this field.

The classification emerged from a rigorous, multi-step analytical process.

Paper Identification: Initially we performed comprehensive keyword searches in academic databases to gather relevant state-of-the art literature. This step ensured that all significant and recent contributions were included.

Understanding the Core Problem: Each identified paper was critically examined to identify its primary research problem. This involved recognizing whether the focus was on specific vulnerabilities, novel detection methodologies, or particular domains within blockchain security.

Identifying Key Features and Approaches: In this step, we extracted the methodologies and techniques used by the papers. Key methodologies such as deep learning, static analysis, fuzzing, or hybrid approaches were extracted. This allowed us to map techniques clearly to identified vulnerabilities and research problems.

Grouping Based on Similar Features: Papers were then grouped together based on shared characteristics and approaches. This clustering process ensured that each category represented a distinct theme or focus. This facilitated a detailed comprehensive analysis.

Formalizing the Taxonomy: Clusters were systematically organized into a coherent taxonomy, clearly delineating methodological boundaries and ensuring each paper was categorized based on its primary contributions.This organization reflects the common themes and features identified in the previous steps.

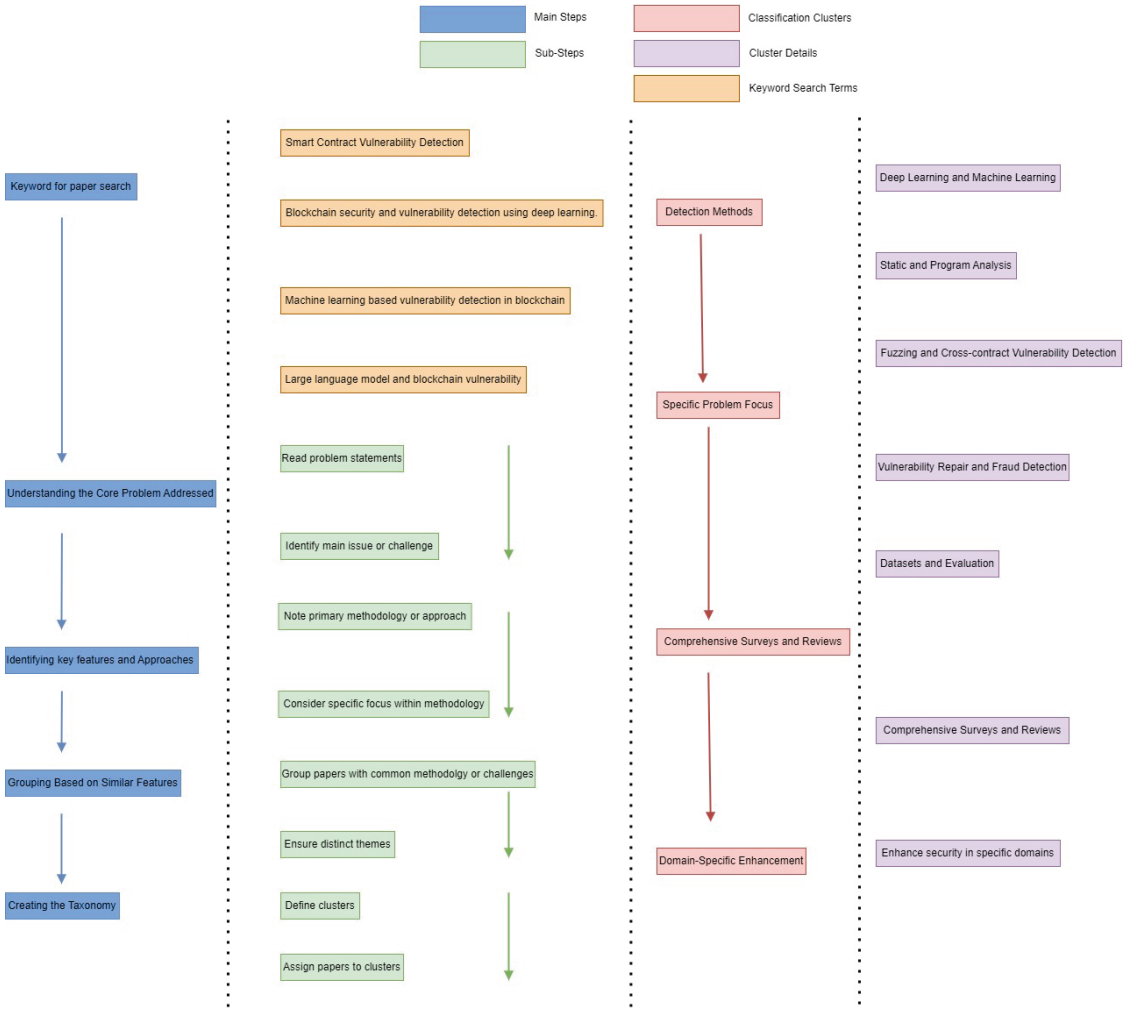


Figure 7. Systematic taxonomy creation for blockchain vulnerability detection research papers.

KeyAspects of the Taxonomy

The taxonomy is organized around several key aspects, which provide insights into current research trends and help identify potential gaps.

1. Detection Methods:

Deep Learning and Machine Learning: Papers using neural networks, graph neural networks, attention models, or other machine learning techniques for vulnerability detection.

Static Program Analysis: Research focusing on static code analysis, program analysis, and formal methods to identify vulnerabilities in smart contracts and blockchain applications.

Fuzzing: Studies that utilize fuzz testing techniques to uncover vulnerabilities by inputting random data into blockchain systems.

2. Specific Problem Focus:

Vulnerability Repair and Fraud Detection: Research dedicated to the automated repair of detected vulnerabilities and the identification of fraudulent transactions within blockchain networks.

Datasets and Evaluation: Papers that create or leverage large datasets for evaluating the performance and robustness of smart contract analysis techniques.

3. Comprehensive Surveys and Reviews: These works provide broad overviews of the field, classifying existing tools and methodologies while outlining future research directions in blockchain vulnerability detection.

4. Domain-Specific Enhancements: This category includes research focused on enhancing security in specific domains, such as the Internet of Things (IoT) or blockchain consensus mechanisms, to address unique challenges in those areas.

In summary, this taxonomy not only clarifies the landscape of blockchain vulnerability detection research but also offers guidance for future studies by highlighting current trends and identifying areas that require further exploration. As illustrated in Figure 7, the classification provides a structured framework that can help researchers select suitable techniques and address potential gaps in the literature.

5.2. Detailed Comparative Analysis on Vulnerability Detection Methods

Blockchain vulnerability detection is a diverse research area in which various methodologies are employed to address different aspects of blockchain security. Our proposed taxonomy enables us to identify papers in distinct clusters, thereby providing a detailed comparative analysis. In Table 4, we summarize the strengths and weaknesses of the different categories of vulnerability detection techniques.

Table 4. Taxonomy of blockchain vulnerability detection and analysis approaches.

Categories	References	Strengths	Weaknesses
Deep Learning and Machine Learning	[52–69]	• Automated and multi-modal feature extraction • High detection accuracy and performance • Efficiency and scalability (can handle large datasets) • Adaptability and flexibility (can evolve with new data) • Reduction of manual intervention	• High computational and resource cost • Data dependency and potential quality issues • Black-box nature (lack of interpretability) • Limited generalization for novel vulnerabilities • Lack of standardized datasets • Complexity of implementation

Table 4. *Cont.*

Categories	References	Strengths	Weaknesses
Static and Program Analysis	[70–82]	• Formal and comprehensive verification • Reduced false positives via pattern matching • Flexibility and expandability • Cost-effectiveness (no runtime overhead) • Detailed vulnerability analysis • Often independent of source code availability	• Resource-intensive and limited scalability (for large contracts) • Soundness vs. completeness tradeoff • Limited language/tool support • Requires expert knowledge to define checks • Difficulty with complex flows (loops, recursion) • Potential for false positives or inaccuracies
Vulnerability Repair and Fraud Detection	[79,82–86]	• Automated methods for repairing complex vulnerabilities • Reduction of false positives and high F1 scores • Decentralized or transaction-based detection • Scalability with potential for transfer learning • Helps filter out suspicious/attack transactions	• Often specialized for specific vulnerabilities (e.g., re-entrancy) • Complexity of implementation and potential overhead • Performance trade-offs (precision vs. recall) • Some solutions target general blockchain transactions, not SC logic • Dependency on particular models or heuristics
Fuzzing and Cross-Contract Analysis	[66,80,87,88]	• Dynamic analysis with real-world simulation • Well suited for complex/cross-contract vulnerabilities • Improved code coverage through guided testing • Allows for informed decision-making on potential exploits	• Possibly incomplete coverage of all paths • Can be computationally expensive • Difficulty scaling to large or highly complex contracts • Dependence on specialized fuzzing techniques
Datasets and Evaluations	[65,73,78,89–94]	• Real-world, large-scale data for robust evaluations • Fine-grained vulnerability detection • Facilitate comparison of multiple tools • Potential use of best practices	• May overlook certain or novel vulnerabilities • Data imbalance or incomplete data (some only have bytecode) • High computational cost for large datasets • Non-standard labeling and limited interpretability

Table 4. *Cont.*

Categories	References	Strengths	Weaknesses
Comprehensive Surveys and Reviews	[63,65,68,73, 77,79,81,93, 95–99]	• Broad overview and categorization of vulnerabilities • Comparison of tools and analysis methods • Practical/future-oriented recommendations • Highlight research gaps for future work	• Rapidly outdated in this fast-evolving field • Often lack specific solutions or detailed evaluations • May overfocus on Ethereum-based platforms • Limited novel contributions (mostly summarizing existing work)
Domain-Specific Enhance-ments	[63,64,69,79, 93,99–104]	• Application-specific insights and tailored solutions • Enhanced security for specialized domains (e.g., IoV, automotive) • Potential for zero-day or niche threat detection • Improved scalability or energy efficiency in specific contexts	• Often limited in scope or domain generalizability • High computa-tional/implementation overhead • Reliability depends on specialized training data • Challenges in interpretability or real-world validation

5.2.1. Machine Learning and Deep Learning Approaches

This cluster has shown remarkable promise due to its ability to recognize complex vulnerability patterns beyond conventional analysis. Techniques involving attention mechanisms [105,106] and transfer learning allow models to focus on the most relevant parts of the code, increasing detection accuracy. Approaches such as those presented in [54] transform contract bytecode or abstract syntax trees into graphs, using graph neural networks (GNNs) to identify structural patterns indicative of vulnerabilities. Studies such as [55,64] incorporate attention layers (e.g., self-attention, dual-attention) to highlight critical code fragments—often referred to as "hotspots"—where bugs are more likely. Moreover, frameworks such as [67–69] leverage distributed machine learning to detect new (zero-day) vulnerabilities while safeguarding sensitive code and data. Transfer learning further addresses the data scarcity problem by reusing feature representations learned from large code corpora. Hybrid approaches, such as combining CNNs with RNNs or merging symbolic execution with deep learning, have shown promise in balancing interpretability with accuracy.

Issues in the Existing Approaches

Despite these advances, the computational intensity and complexity of these models remain significant challenges for large-scale practical deployment. In addition, ensuring high-quality, representative training data is essential, and strategies for enhanced data collection, pre-processing, and augmentation are needed. Moving beyond simplistic binary classification to more nuanced detection strategies can further improve detection accuracy.

5.2.2. Static and Program Analysis-Based Methods

Static analysis and program analysis-based methods play a crucial role in reducing false positives and reliably detecting vulnerabilities. Tools such as SmartCheck [75] and SafeCheck [70] apply rule-based analysis to Solidity code, effectively detecting known anti-patterns such as re-entrancy and integer overflow. Other approaches, including RA [74] and SmartFast [78], introduce specialized symbolic execution engines tailored to uncover logic bugs in Ethereum bytecode. Formal verification methods [73,76] use rigorous mathematical proofs, such as model checking, to guarantee the absence of certain classes of bugs.

Issues in the Existing Approaches

While these methods are cost-effective and efficient in detecting vulnerabilities, their implementation complexity, resource requirements, and need for expert configuration can limit their accessibility. Balancing cost and performance, simplifying the implementation process, and reducing dependency on specific tools can improve the overall usability of these methods.

5.2.3. Fuzzing and Cross-Contract Methods

Fuzzing techniques have emerged as effective means of improving code coverage and vulnerability detection in smart contracts. Intelligent guidance and prioritization—as demonstrated by LLM4Fuzz [88]—along with approaches such as CrossFuzz [87] have proven effective in exploring multiple execution paths to identify vulnerabilities. Evo-Fuzzer [80] employs evolutionary algorithms to generate inputs more intelligently.

Issues in the Existing Approaches

Although these methods enhance detection capabilities, they are often resource intensive and complex to implement. Their success depends heavily on fine-tuning, the use of multiple large language models (LLMs) for consensus, and expert configuration. Scalability and adaptability remain challenging for the widespread adoption of fuzzing techniques.

5.2.4. Vulnerability Repair and Fraud Detection

A key finding in vulnerability repair and fraud detection research is the importance of achieving high precision and recall. Approaches such as bytecode-based repair [82] focus on patching re-entrancy vulnerabilities at the Ethereum Virtual Machine (EVM) level, thereby bypassing the need for source code modifications. Fraud detection methods [85,86] leverage heuristic or machine learning-based classification to identify suspicious transactions, thus mitigating financial risks in decentralized applications (DApps). Tools such as sGuard+ [83] automatically repair vulnerable contracts by modifying them to fix known logic errors (e.g., integer overflows) before redeployment. The scalability of these methods is a significant advantage, particularly in diverse smart contract environments, though their dependency on specific models emphasizes the need for continuous improvement and adaptation.

5.2.5. Datasets and Evaluation-Based Methods

Datasets and evaluation-based methods provide detailed, comprehensive insights into vulnerabilities. For instance, DAppSCAN [89] plays a crucial role in improving our understanding of smart contract weaknesses. Fine-grained vulnerability detection techniques [90] maintain granular labels (e.g., re-entrancy, integer overflow) for each function or code block, thereby enhancing training and evaluation processes. Tools such as Oyente and Mythril [73,92] have contributed partial datasets for benchmarking new

solutions, while SmartFast [78] and ReDefender [94] offer curated test cases focusing on common vulnerabilities. Despite these advances, there remains a need for continuous updating and expansion of vulnerability databases, such as the SWC Registry [107], to ensure comprehensive vulnerability coverage.

5.2.6. Comprehensive Surveys and Domain-Specific Enhancements

Numerous surveys synthesize existing research, highlight trends, and suggest promising directions. For example, works such as [77,79,93] provide broad overviews of blockchain vulnerabilities, ranging from protocol-level to DApp-level weaknesses. Other studies [63,98] focus on the evolution of analysis tools and the rise of deep learning in detection. Additional surveys [81,95,96] concentrate on Ethereum-specific issues, while others [65,68,99] highlight modern challenges such as zero-day vulnerabilities and advanced ML approaches. The consensus emerging from these surveys is that no single method can cover the extensive, evolving threat landscape; instead, hybrid strategies are necessary.

Domain-specific enhancements are also crucial. In the context of blockchain and IoT security, research [100,101] adapts ML-based intrusion detection and consensus strategies to accommodate IoT constraints, such as limited computational power. Similarly, automotive security research [103] addresses vulnerabilities in automotive blockchain networks, focusing on real-time and safety-critical requirements. Other works [64,69,104] propose domain-focused modifications to block structure, consensus, or communication patterns, aiming to balance security and efficiency. Future research should focus on optimizing these approaches to minimize resource consumption without compromising security, and on developing robust data collection and preprocessing techniques to ensure high-quality training data.

5.2.7. Key Insights from the Vulnerability Detection Taxonomy

Through comprehensive analysis, several key insights emerge regarding vulnerability detection and security within blockchain environments.

Hybrid Approaches Enhance Security: No single approach—whether based on machine learning, static analysis, or fuzzing—can fully cover the vast and evolving threat landscape in blockchain environments. Hybrid strategies, such as combining symbolic execution with deep neural networks [57] or using static analysis to identify suspicious points before applying fuzzing, are emerging as best practices. The integration of multiple methods—including formal verification for critical invariants, deep learning for dynamic anomaly detection, and fuzzing for real-world scenario testing—provides a more robust defense against malicious actors.

Adaptability and Machine Learning: Machine learning provides the flexibility and adaptability necessary for real-time anomaly detection and novel vulnerability identification, complementing traditional security measures. Its capability for advanced pattern recognition makes it highly effective against evolving threats.

Persistent Challenges Require Attention: Several challenges persist, demanding ongoing research and community collaboration: (1) Scalability and Efficiency: Techniques must effectively handle large-scale contracts and cross-chain interactions without incurring excessive overhead. (2) Data Quality and Benchmarking: High-quality, standardized datasets and consensus-driven benchmarks are critical for developing and evaluating reliable security tools. (3) Interpretability and Transparency: Tools must provide clear, understandable explanations suitable for end-users, auditors, and regulatory compliance.

(4) Future-Proofing Against Evolving Threats: Solutions must continually evolve to counter emerging threats, including quantum-based vulnerabilities and sophisticated MEV exploits.

Community-Driven Collaboration is Essential: Successfully addressing blockchain vulnerabilities demands collaboration across disciplines such as software engineering, cryptography, artificial intelligence, and other specialized fields. A multidisciplinary approach is essential to keep blockchain systems innovative and secure.

Collectively, these insights underline the necessity for an integrated, dynamic approach that leverages the strengths of various methods to provide comprehensive and adaptive security solutions. Ultimately, ongoing innovation, proactive collaboration, and a commitment to addressing emerging challenges will ensure the resilience and continued evolution of secure blockchain technologies.

Having thoroughly analyzed the diverse approaches to vulnerability detection, it is clear that traditional methods alone cannot fully address the rapidly evolving threat landscape in blockchain environments. In this context, machine learning (ML) emerges as a powerful tool with the potential to complement and enhance existing security techniques. ML approaches not only improve detection accuracy through advanced pattern recognition and anomaly detection but also offer the adaptability needed to identify novel vulnerabilities in real time. As we transition to the next Section 6, we will explore how ML is being leveraged to fortify blockchain systems, examine its integration with conventional methods, and discuss the unique challenges and opportunities it presents in maintaining robust blockchain security.

6. ML and the Security of Blockchain

The growing adoption of blockchain technology across various sectors has underscored the need to address security concerns inherent in decentralized systems. Although the core characteristics of blockchain—decentralization, immutability, and transparency—provide a strong foundation for secure transactions, vulnerabilities persist that can be exploited by adversaries. In response, machine learning (ML) approaches have emerged as a promising solution to strengthen blockchain security. ML-based techniques are now employed to detect and mitigate attacks, improve system performance, and enhance the overall integrity of blockchain networks.

6.1. Anomaly Detection and Intrusion Prevention

One of the primary applications of ML in blockchain security is anomaly detection. Given the decentralized nature of blockchain, detecting malicious behavior or anomalies can be particularly challenging. Supervised ML models, such as decision trees, random forests, and support vector machines (SVMs), can be trained to detect abnormal activities on the blockchain network. These anomalies may include fraudulent transactions, irregular mining patterns, or unusual block generation times, which could signal attempts at double-spending or other forms of fraud [108].

In addition to supervised techniques, unsupervised learning methods, such as clustering algorithms, can learn normal behavior patterns and subsequently flag deviations. These approaches are particularly useful for detecting novel attacks that have not been previously encountered [109,110]. Together, these ML models help bolster blockchain security by providing real-time intrusion prevention and early warning systems.

6.2. Smart Contract Security

Smart contracts are self-executing agreements whose terms are directly embedded in code. Despite their power in automating transactions, they are vulnerable to a range

of security issues, including re-entrancy attacks, integer overflows, and faulty logic. ML techniques—particularly deep learning and natural language processing (NLP)—have increasingly been used to detect such vulnerabilities before contracts are deployed.

For instance, deep learning models can be trained on large datasets of known smart contract vulnerabilities to classify and identify problematic code. NLP techniques, on the other hand, analyze the semantic structure of the contract source code to detect patterns indicative of security flaws. As reported in [111], bidirectional encoder representations from transformers (BERT) is leveraged to capture semantic features from smart contract opcodes. When combined with a bidirectional LSTM (BiLSTM), the model effectively handles long-term dependencies in the contract code, while an attention mechanism prioritizes critical segments. These innovations enable blockchain developers to automate vulnerability detection and reduce the risk of deploying insecure contracts.

6.3. Consensus Mechanism Enhancement

ML techniques are also applied to enhance blockchain consensus mechanisms. For example, ML models can analyze mining patterns and participant behavior to predict potential 51% attacks. Reinforcement learning has been used to develop adaptive consensus mechanisms that dynamically adjust based on real-time network conditions. This adaptability is crucial in networks utilizing Proof of Work (PoW) or Proof of Stake (PoS), where miners or validators might be incentivized to act maliciously. By integrating ML, consensus algorithms can mitigate risks such as centralization and enhance overall network resilience.

6.4. Sybil Attack Mitigation

Sybil attacks, which involve the creation of multiple fake identities to gain undue influence over a network, can also be countered using ML techniques. Graph-based ML algorithms, such as graph neural networks (GNNs), are effective in analyzing peer-to-peer relationships within the network. These algorithms can detect anomalous patterns indicative of clusters of fake or suspicious nodes [109]. By classifying nodes based on behavior and contribution, ML models can help prevent adversaries from manipulating consensus through artificial node creation.

6.5. Quantum Attack Resistance

Although quantum attacks remain largely theoretical, ML can contribute to the development of quantum-resistant cryptographic methods. ML models can simulate quantum attack scenarios on various cryptographic algorithms, enabling researchers to identify potential vulnerabilities. These insights facilitate the development and optimization of post-quantum cryptographic techniques [112]. ML thus plays a dual role by both simulating advanced attack vectors and assisting in the design of more robust, quantum-resistant security measures.

6.6. Network Traffic Analysis

ML-based intrusion detection systems (IDS) are critical for analyzing network traffic in real time and mitigating distributed denial of service (DDoS) attacks. Unsupervised learning methods enable these systems to learn normal network behavior patterns and flag unusual traffic spikes or irregular patterns. Furthermore, reinforcement learning techniques can be applied to develop dynamic traffic-filtering mechanisms that adjust in real time, ensuring continued network availability even during DDoS attacks.

6.7. Blockchain Forensics

In the event of a security breach, ML-based forensic tools can analyze blockchain data to trace the origin and flow of malicious activities. Supervised learning algorithms can identify suspicious addresses and flag those involved in known attack patterns. These ML techniques facilitate comprehensive blockchain forensics by tracking stolen assets and mapping the methods used by attackers, thereby providing critical insights for preventing future incidents.

6.8. BlockGPT Framework

A recent breakthrough in this domain is the BlockGPT framework [113], a novel real-time anomaly detection system tailored for blockchain transactions, particularly within decentralized finance (DeFi) applications. BlockGPT utilizes a large language model (LLM) to analyze Ethereum transaction execution traces without relying on predefined patterns, making it highly adaptable to various types of attacks. Trained on a dataset of 68 million Ethereum transactions, including those involving smart contract attacks, BlockGPT captures detailed execution traces such as function calls, state changes, and event logs. This enables the framework to rank transactions by their likelihood of being abnormal, demonstrating strong performance in real-time detection with high throughput and low false-positive rates.

6.9. Adversarial Use of ML in Blockchain Security

While ML offers significant benefits for enhancing blockchain security, it also introduces new vulnerabilities. Adversaries can leverage AI and ML-based systems to execute sophisticated attacks, such as 51% attacks or Sybil attacks, at scale. For instance, research [114] discusses how AI bots may be exploited to carry out these attacks, and [115] reports that an AI bot attack resulted in a $20 million loss for Ethereum. Additionally, [116] demonstrates how large language platforms can be used to identify smart contract vulnerabilities, potentially enabling malicious actors to exploit them. Despite these challenges, the rapid advancements in ML and generative AI (GenAI) provide an opportunity to design more effective systems for early detection and prevention of adversarial attacks.

6.10. Broader Perspectives on ML in Blockchain Security

Overall, while many blockchain vulnerabilities are detectable, the primary challenge lies in the timeliness of detection to mitigate losses. AI technologies can accelerate breach detection and enhance the quality of security measures. With the emergence of generative AI and large language models (LLMs), there is significant potential for designing systems capable of real-time detection and prevention of adversarial attacks. For a broader perspective on the application of ML in blockchain security, readers are encouraged to refer to [117–119].

In this section, we explored the multifaceted role of machine learning (ML) in enhancing the security of blockchain systems. We began by discussing anomaly detection and intrusion prevention techniques that leverage both supervised and unsupervised ML models to identify irregular behaviors in blockchain transactions. Next, we examined how ML aids smart contract security by employing deep learning and NLP to detect vulnerabilities before deployment, followed by an analysis of ML-driven enhancements to consensus mechanisms and mitigation strategies against Sybil attacks. We also highlighted the application of ML in network traffic analysis, blockchain forensics, and innovative frameworks such as BlockGPT for real-time anomaly detection. Finally, we addressed the emerging dual role of ML, noting that while it offers powerful defenses against attacks, it also introduces

new vulnerabilities that adversaries can exploit. Overall, ML technologies present both opportunities and challenges in maintaining robust blockchain security, underscoring the need for continuous research and adaptive strategies.

7. Discussion and Future Directions

This survey underscores the multifaceted nature of blockchain security, emphasizing that comprehensive security measures require structured analysis of vulnerabilities and methodological clarity in their detection. The proposed taxonomies offer practical frameworks that help blockchain practitioners, developers, and cybersecurity analysts categorize attacks systematically and design targeted mitigation strategies. Practically, this means smart contract developers can leverage existing auditing frameworks (e.g., Slither or Mythril) in alignment with the vulnerabilities identified by our taxonomy, potentially increasing the effectiveness and efficiency of vulnerability detection and mitigation processes.

More broadly, the taxonomy can serve as a foundational tool to inform real-world blockchain security policies by enabling system designers and security teams to map out threat surfaces layer by layer, assess which operational components are at risk, and define security controls that directly correspond to those architectural layers. This structured alignment between vulnerabilities and architectural responsibilities can help formalize internal audit procedures, incident response planning, and regulatory compliance strategies in both enterprise and public blockchain settings.

An important consideration is the adaptability of our taxonomies to evolving blockchain paradigms. Future iterations of our taxonomies can explicitly address emerging threats and technological advancements such as quantum computing and artificial intelligence-driven vulnerabilities. Quantum-resistant cryptographic schemes and consensus algorithms will become increasingly critical as quantum computing capabilities advance. Furthermore, as machine learning, particularly federated learning (FL), becomes more prevalent, its application to decentralized, privacy-preserving security analytics emerges as a promising research direction. Another critical avenue is the integration of zero-trust security paradigms to enable continuous authentication and verification within blockchain networks, addressing the growing sophistication of attacks.

The emergence of large language models (LLMs) also opens new possibilities for automated, sophisticated code analysis and real-time vulnerability detection. However, adversaries could leverage similar models for vulnerability identification and exploitation. This duality emphasizes the urgent need for robust, explainable, and adaptive ML-based defense mechanisms. Addressing these challenges will necessitate continued interdisciplinary collaboration among blockchain specialists, cybersecurity researchers, cryptographers, quantum computing experts, and policymakers. Such collaboration will ensure that blockchain ecosystems remain resilient against current vulnerabilities and future threats.

Lastly, although we did not perform a comparative empirical study, our layer-based framework can help identify which architectural layers may be more susceptible in specific blockchain architectures. For instance, platforms using DAG-based structures such as IOTA may face unique data-layer vulnerabilities, while PoW-based systems are more exposed to consensus-layer threats such as selfish mining. Such insights offer a strong starting point for future research into architecture-specific threat modeling.

8. Conclusions

Blockchain technology is inherently robust due to its decentralized architecture and immutable data integrity. Nonetheless, persistent security incidents highlight existing vul-

nerabilities and potential exploits within blockchain systems. In this work, we introduced two structured frameworks to enhance blockchain security: a layer-based taxonomy for categorizing blockchain attacks and a methodological taxonomy for blockchain vulnerability detection. The proposed layer-based taxonomy systematically categorizes vulnerabilities across blockchain layers, including the application, contract, consensus and incentive, network, and data layers facilitating precise identification of vulnerabilities and enabling targeted, layer-specific mitigation strategies. This structured approach promotes a nuanced understanding of attack vectors, revealing that vulnerabilities frequently target consensus mechanisms, network infrastructure, or smart contract logic. Effective mitigation thus requires comprehensive technical solutions, secure coding standards, robust consensus protocols, and strategic network governance practices, such as enhancing miner diversity and peer selection criteria.

The methodological taxonomy presented in this study systematically categorizes existing vulnerability detection approaches into coherent clusters: machine learning (ML) and deep learning (DL) techniques, static program analysis, fuzz testing, vulnerability repair methods, and fraud detection. By organizing existing research methodologies clearly, this taxonomy provides insights into the strengths and limitations inherent in each approach. Our analysis reveals that, while ML and DL methods generally achieve high accuracy and performance, they frequently rely on extensive, high-quality datasets and may lack explainability. Conversely, vulnerability repair and fraud detection methods show strong practical effectiveness but may exhibit inconsistent performance across various vulnerability classes.

Overall, our proposed frameworks provide clarity, structure, and actionable guidance for both academia and industry, significantly enhancing the blockchain security community's capability to prioritize security investments and align research agendas effectively. By systematically addressing blockchain vulnerabilities, this work lays a solid foundation for future research aimed at strengthening blockchain resilience against existing and emerging threats.

Author Contributions: Conceptualization, M.H. and S.M.R.I.; methodology, M.J.I., S.I., and S.M.R.I.; validation, M.J.I., S.I., and S.M.R.I.; formal analysis, M.J.I. and S.I.; investigation, M.J.I., S.I., M.H., S.N., and S.M.R.I.; writing—original draft preparation, M.J.I., S.I., and S.M.R.I.; writing—review and editing, M.H. and S.N.; visualization, M.J.I., S.I., and S.M.R.I.; supervision, S.M.R.I.; project administration, M.J.I., S.I., and S.N.; funding acquisition, S.M.R.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study is only contained in the article itself.

Conflicts of Interest: Mahmud Hossain was employed by the company Amazon Web Services (AWS). The authors declare no conflicts of interest.

References

1. IBM. What Is Blockchain Security? Available online: https://www.ibm.com/topics/blockchain-security (accessed on 27 January 2024).
2. Baldwin, C. Bitcoin Worth 72 Million Stolen from Bitfinex Exchange in Hong Kong. 2016. Available online: https://www.reuters.com/article/us-bitfinex-hacked-hongkong-idUSKCN10E0KP (accessed on 15 March 2024).
3. Popper, N. A Hacking of More than 50 Million Dashes Hopes in the World of Virtual Currency. 2016. Available online: https://www.nytimes.com/2016/06/18/business/dealbook/hacker-may-have-removed-more-than-50-million-from-experimental-cybercurrency-project.html (accessed on 13 February 2024).

4. Zhao, W. Crypto Exchange Bithumb Hacked for 13 Million in Suspected Insider Job. 2019. Available online: https://www.coindesk.com/markets/2019/03/30/crypto-exchange-bithumb-hacked-for-13-million-in-suspected-insider-job (accessed on 10 April 2025).
5. Mollajafari, S.; Bechkoum, K. Blockchain technology and related security risks: Towards a seven-layer perspective and taxonomy. *Sustainability* **2023**, *15*, 13401. [CrossRef]
6. Dwivedi, K.; Agrawal, A.; Bhatia, A.; Tiwari, K. A novel classification of attacks on blockchain layers: Vulnerabilities, attacks, mitigations, and research directions. *arXiv* **2024**, arXiv:2404.18090.
7. Gurjar, A.; Chandavarkar, B. Smart Contract Vulnerabilities and Detection Methods: A Survey. In Proceedings of the 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 24–28 June 2024; IEEE: New York, NJ, USA, 2024; pp. 1–7.
8. Hejazi, N.; Lashkari, A.H. A comprehensive survey of smart contracts vulnerability detection tools: Techniques and methodologies. *J. Netw. Comput. Appl.* **2025**, *237*, 104142. [CrossRef]
9. Chu, H.; Zhang, P.; Dong, H.; Xiao, Y.; Ji, S.; Li, W. A survey on smart contract vulnerabilities: Data sources, detection and repair. *Inf. Softw. Technol.* **2023**, *159*, 107221. [CrossRef]
10. Rajawat, A.S.; Goyal, S.; Kumar, M.; Singh, T.P. An AI-Enabled Blockchain Algorithm: A Novel Approach to Counteract Blockchain Network Security Attacks. *EAI Endorsed Trans. Internet Things* **2024**, *10*, 1–9. [CrossRef]
11. Magalhães, J. Current Vulnerabilities and Risks of Blockchain: A Systematic Literature Review. In *Digital Technologies and Transformation in Business, Industry and Organizations*; Springer: Cham, Switzerland, 2025; pp. 159–175.
12. Verma, P.; Bharot, N.; Breslin, J.G.; O'Shea, D.; Vidyarthi, A.; Gupta, D. Zero-day guardian: A dual model enabled federated learning framework for handling zero-day attacks in 5G enabled IIoT. *IEEE Trans. Consum. Electron.* **2023**, *70*, 3856–3866. [CrossRef]
13. Hallaji, E.; Razavi-Far, R.; Saif, M.; Wang, B.; Yang, Q. Decentralized federated learning: A survey on security and privacy. *IEEE Trans. Big Data* **2024**, *10*, 194–213. [CrossRef]
14. Huan, N.T.Y.; Zukarnain, Z.A. A Survey on Addressing IoT Security Issues by Embedding Blockchain Technology Solutions: Review, Attacks, Current Trends, and Applications. *IEEE Access* **2024**, *12*, 69765–69782. [CrossRef]
15. Li, D.; Han, D.; Weng, T.H.; Zheng, Z.; Li, H.; Liu, H.; Castiglione, A.; Li, K.C. Blockchain for federated learning toward secure distributed machine learning systems: A systemic survey. *Soft Comput.* **2022**, *26*, 4423–4440. [CrossRef]
16. Dai, H.N.; Zheng, Z.; Zhang, Y. Blockchain for Internet of Things: A survey. *IEEE Internet Things J.* **2019**, *6*, 8076–8094. [CrossRef]
17. Liu, P.; Xu, X.; Wang, W. Threats, attacks and defenses to federated learning: Issues, taxonomy and perspectives. *Cybersecurity* **2022**, *5*, 4. [CrossRef]
18. Lee, J.H. Systematic Approach to Analyzing Security and Vulnerabilities of Blockchain Systems. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2019.
19. Zamani, E.; He, Y.; Phillips, M. On the security risks of the blockchain. *J. Comput. Inf. Syst.* **2020**, *60*, 495–506. [CrossRef]
20. Singh, S.; Hosen, A.S.; Yoon, B. Blockchain security attacks, challenges, and solutions for the future distributed iot network. *IEEE Access* **2021**, *9*, 13938–13959. [CrossRef]
21. Zarrin, J.; Wen Phang, H.; Babu Saheer, L.; Zarrin, B. Blockchain for decentralization of internet: Prospects, trends, and challenges. *Clust. Comput.* **2021**, *24*, 2841–2866. [CrossRef]
22. Manu, M.; Musthafa, N.; Balamurugan, B.; Chauhan, R. Blockchain components and concept. In *Blockchain Technology and Applications*; Auerbach Publications: Boca Raton, FL, USA, 2020; pp. 21–50.
23. Patel, V.; Khatiwala, F.; Shah, K.; Choksi, Y. A review on blockchain technology: Components, issues and challenges. In Proceedings of the ICDSMLA 2019: Proceedings of the 1st International Conference on Data Science, Machine Learning and Applications, Hyderabad, India, 26–28 July 2019; Springer: Berlin/Heidelberg, Germany, 2020; pp. 1257–1262.
24. Puthal, D.; Malik, N.; Mohanty, S.P.; Kougianos, E.; Das, G. Everything you wanted to know about the blockchain: Its promise, components, processes, and problems. *IEEE Consum. Electron. Mag.* **2018**, *7*, 6–14. [CrossRef]
25. IBM. What Is Blockchain Technology? 2022. Available online: https://www.ibm.com/topics/what-is-blockchain (accessed on 10 September 2024).
26. Zhang, R.; Xue, R.; Liu, L. Security and privacy on blockchain. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–34. [CrossRef]
27. Bhushan, B.; Sinha, P.; Sagayam, K.M.; Andrew, J. Untangling blockchain technology: A survey on state of the art, security threats, privacy services, applications and future research directions. *Comput. Electr. Eng.* **2021**, *90*, 106897. [CrossRef]
28. Saad, M.; Spaulding, J.; Njilla, L.; Kamhoua, C.; Shetty, S.; Nyang, D.; Mohaisen, A. Exploring the attack surface of blockchain: A systematic overview. *arXiv* **2019**, arXiv:1904.03487.
29. Mosakheil, J.H. Security Threats Classification in Blockchains. Master's Thesis, Herberger School of Business, St. Cloud, MN, USA, 2018.

30. Li, X.; Jiang, P.; Chen, T.; Luo, X.; Wen, Q. A survey on the security of blockchain systems. *Future Gener. Comput. Syst.* **2020**, *107*, 841–853. [CrossRef]
31. Dasgupta, D.; Shrein, J.M.; Gupta, K.D. A survey of blockchain from security perspective. *J. Bank. Financ. Technol.* **2019**, *3*, 1–17. [CrossRef]
32. Anita, N.; Vijayalakshmi, M. Blockchain security attack: A brief survey. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; IEEE: New York, NY, USA, 2019; pp. 1–6.
33. Putz, B.; Pernul, G. Detecting blockchain security threats. In Proceedings of the 2020 IEEE International Conference on Blockchain (Blockchain), Rhodes, Greece, 2–6 November 2020; IEEE: New York, NJ, USA, 2020; pp. 313–320.
34. Gao, R.; Zhang, Q.; Chen, Y.; Ma, J. A Taxonomy of Consensus Attacks in Blockchain. *Future Gener. Comput. Syst.* **2022**, *128*, 325–338. [CrossRef]
35. Cheng, J.; Xie, L.; Tang, X.; Xiong, N.; Liu, B. A survey of security threats and defense on Blockchain. *Multimed. Tools Appl.* **2021**,*80*, 30623–30652. [CrossRef]
36. Toapanta, T.S.M.; Del Pozo, D.R.; Izurieta, R.R.; Guamán, J.A.; Orizaga, J.A.; Arellano, R.M.; Hifóng, M.M.B. Blockchain-based Security Model to Mitigate the Risks of a Database for a Public Organization. *J. Internet Serv. Inf. Secur. (JISIS)* **2024**, *14*, 78–98. [CrossRef]
37. Parino, F.; Beiro, M.; Gauvin, L. Analysis of the Bitcoin blockchain: Socio-economic factors behind the adoption. *EPJ Data Sci.* **2018**, *7*, 38. [CrossRef]
38. Popov, S. *The Tangle*; White Paper; IOTA: Berlin, Germany, 2018.
39. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In Proceedings of the EuroSys, Porto, Portugal, 23–26 April 2018.
40. Buterin, V. Ethereum 2.0 Phase 0—Beacon Chain, 2020. Available online: https://ethereum.org/en/upgrades/beacon-chain/ (accessed on 10 August 2024).
41. Kwon, J.; Buchman, E. Cosmos: A Network of Distributed Ledgers. *Cosm. Whitepaper* **2019**, 1–32.
42. Wen, Y.; Lu, F.; Liu, Y.; Huang, X. Attacks and countermeasures on blockchains: A survey from layering perspective. *Comput. Netw.* **2021**, *191*, 107978. [CrossRef]
43. Kearney, J.J.; Perez-Delgado, C.A. Vulnerability of blockchain technologies to quantum attacks. *Array* **2021**, *10*, 100065. [CrossRef]
44. Allende, M.; León, D.L.; Cerón, S.; Pareja, A.; Pacheco, E.; Leal, A.; Da Silva, M.; Pardo, A.; Jones, D.; Worrall, D.J.; et al. Quantum-resistance in blockchain networks. *Sci. Rep.* **2023**, *13*, 5664. [CrossRef]
45. Hasanova, H.; Baek, U.J.; Shin, M.G.; Cho, K.; Kim, M.S. A survey on blockchain cybersecurity vulnerabilities and possible countermeasures. *Int. J. Netw. Manag.* **2019**, *29*, e2060. [CrossRef]
46. Cooper, E.; Weese, E.; Fortson, A.; Lo, D.; Shi, Y. Cyber Security in Blockchain. In Proceedings of the 2023 IEEE Conference on Dependable and Secure Computing (DSC), Tampa, FL, USA, 7–9 November 2023; IEEE: New York, NJ, USA, 2023; pp. 1–11.
47. Wang, H.; Wang, Y.; Cao, Z.; Li, Z.; Xiong, G. An overview of blockchain security analysis. In Proceedings of the Cyber Security: 15th International Annual Conference, CNCERT 2018, Beijing, China, 14–16 August 2018; Revised Selected Papers 15; Springer: Singapore, 2019; pp. 55–72.
48. Sayeed, S.; Marco-Gisbert, H.; Caira, T. Smart contract: Attacks and protections. *IEEE Access* **2020**, *8*, 24416–24427. [CrossRef]
49. Georgia, W. Arithmetic Underflow and Overflow Vulnerabilities in Smart Contracts. 2023. Available online: https://101blockchains.com/underflow-and-overflow-vulnerabilities-in-smart-contracts/ (accessed on 22 April 2024).
50. Guggenberger, T.; Schlatt, V.; Schmid, J.; Urbach, N. A Structured Overview of Attacks on Blockchain Systems. In Proceedings of the PACIS, Dubai, United Arab Emirates, 12–14 July 2021; p. 100.
51. Atzei, N.; Bartoletti, M.; Cimoli, T. A survey of attacks on ethereum smart contracts (sok). In Proceedings of the 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, 22–29 April 2017; Proceedings 6; Springer: Berlin/Heidelberg, Germany, 2017; pp. 164–186.
52. Chen, Y.; Sun, Z.; Gong, Z.; Hao, D. Improving Smart Contract Security with Contrastive Learning-based Vulnerability Detection. In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering, Lisbon, Portugal, 14–20 April 2024; pp. 1–11.
53. Luo, F.; Luo, R.; Chen, T.; Qiao, A.; He, Z.; Song, S.; Jiang, Y.; Li, S. Scvhunter: Smart contract vulnerability detection based on heterogeneous graph attention network. In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering, Lisbon, Portugal, 14–20 April 2024; pp. 1–13.

54. Zhen, Z.; Zhao, X.; Zhang, J.; Wang, Y.; Chen, H. DA-GNN: A smart contract vulnerability detection method based on Dual Attention Graph Neural Network. *Comput. Netw.* **2024**, *242*, 110238. [CrossRef]
55. Sun, Y.; Gu, L. Attention-based machine learning model for smart contract vulnerability detection. *J. Phys. Conf. Ser.* **2021**, *1820*, 012004. [CrossRef]
56. Lutz, O.; Chen, H.; Fereidooni, H.; Sendner, C.; Dmitrienko, A.; Sadeghi, A.R.; Koushanfar, F. Escort: Ethereum smart contracts vulnerability detection using deep neural network and transfer learning. *arXiv* **2021**, arXiv:2103.12607.
57. Colin, L.S.H.; Mohan, P.M.; Pan, J.; Keong, P.L.K. An Integrated Smart Contract Vulnerability Detection Tool Using Multi-layer Perceptron on Real-time Solidity Smart Contracts. *IEEE Access* **2024**, *12*, 23549–23567. [CrossRef]
58. Jain, V.K.; Tripathi, M. An integrated deep learning model for Ethereum smart contract vulnerability detection. *Int. J. Inf. Secur.* **2024**, *23*, 557–575. [CrossRef]
59. Zhou, Q.; Zheng, K.; Zhang, K.; Hou, L.; Wang, X. Vulnerability analysis of smart contract for blockchain-based IoT applications: A machine learning approach. *IEEE Internet Things J.* **2022**, *9*, 24695–24707. [CrossRef]
60. Narayana, K.L.; Sathiyamurthy, K. Automation and smart materials in detecting smart contracts vulnerabilities in Blockchain using deep learning. *Mater. Today Proc.* **2023**, *81*, 653–659. [CrossRef]
61. Rabieinejad, E.; Yazdinejad, A.; Parizi, R.M. A deep learning model for threat hunting in ethereum blockchain. In Proceedings of the 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Shenyang, China, 20—22 October 2021; IEEE: New York, NJ, USA, 2021; pp. 1185–1190.
62. Ismail, S.; Nouman, M.; Dawoud, D.W.; Reza, H. Towards a lightweight security framework using blockchain and machine learning. *Blockchain Res. Appl.* **2024**, *5*, 100174. [CrossRef]
63. Bresil, M.; Prasad, P.; Sayeed, M.S.; Bukar, U.A. Deep Learning-based Vulnerability Detection Solutions in Smart Contracts: A Comparative and Meta-Analysis of Existing Approaches. *IEEE Access* **2025**, *13*, 28894–28919. [CrossRef]
64. Chen, J.; Wang, W.; Liu, B.; Cai, S.; Towey, D.; Wang, S. Hybrid semantics-based vulnerability detection incorporating a Temporal Convolutional Network and Self-attention Mechanism. *Inf. Softw. Technol.* **2024**, *171*, 107453. [CrossRef]
65. Crisostomo, J.; Bacao, F.; Lobo, V. Machine learning methods for detecting smart contracts vulnerabilities within Ethereum blockchain- A review. *Expert Syst. Appl.* **2025**, *268*, 126353. [CrossRef]
66. Duy, P.T.; Khoa, N.H.; Quyen, N.H.; Trinh, L.C.; Kien, V.T.; Hoang, T.M.; Pham, V.H. Vulnsense: Efficient vulnerability detection in ethereum smart contracts by multimodal learning with graph neural network and language model. *Int. J. Inf. Secur.* **2025**, *24*, 48. [CrossRef]
67. Mohammed, H.A.; Husien, I.M. A Deep Transfer Learning Framework for Robust IoT Attack Detection: A Review. *Informatica* **2024**, *48*, 55–64. [CrossRef]
68. Korba, A.A.; Boualouache, A.; Ghamri-Doudane, Y. Zero-X: A Blockchain-Enabled Open-Set Federated Learning Framework for Zero-Day Attack Detection in IoV. *IEEE Trans. Veh. Technol.* **2024**, *73*, 12399–12414. [CrossRef]
69. Mbaya, E.B.; Adetiba, E.; Badejo, J.A.; Wejin, J.S.; Oshin, O.; Isife, O.; Thakur, S.C.; Moyo, S.; Adebiyi, E.F. SecFedIDM-V1: A secure federated intrusion detection model with blockchain and deep bidirectional long short-term memory network. *IEEE Access* **2023**, *11*, 116011–116025. [CrossRef]
70. Chen, H.; Zhao, X.; Wang, Y.; Zhen, Z. SafeCheck: Detecting smart contract vulnerabilities based on static program analysis methods. *Secur. Priv.* **2024**, *7*, e393. [CrossRef]
71. Sun, Y.; Wu, D.; Xue, Y.; Liu, H.; Wang, H.; Xu, Z.; Xie, X.; Liu, Y. Gptscan: Detecting logic vulnerabilities in smart contracts by combining gpt with program analysis. In Proceedings of the IEEE/ACM 46th International Conference on Software Engineering, Lisbon, Portugal, 14–20 April 2024; pp. 1–13.
72. Pise, R.G.; Patil, S. Pioneering automated vulnerability detection for smart contracts in blockchain using KEVM: Guardian ADRGAN. *Int. J. Inf. Secur.* **2024**, *23*, 1805–1819. [CrossRef]
73. Singh, A.; Parizi, R.M.; Zhang, Q.; Choo, K.K.R.; Dehghantanha, A. Blockchain smart contracts formalization: Approaches and challenges to address vulnerabilities. *Comput. Secur.* **2020**, *88*, 101654. [CrossRef]
74. Chinen, Y.; Yanai, N.; Cruz, J.P.; Okamura, S. RA: Hunting for re-entrancy attacks in ethereum smart contracts via static analysis. In Proceedings of the 2020 IEEE International Conference on Blockchain (Blockchain), Virtual, 2–6 November 2020; IEEE: New York, NJ, USA, 2020; pp. 327–336.
75. Tikhomirov, S.; Voskresenskaya, E.; Ivanitskiy, I.; Takhaviev, R.; Marchenko, E.; Alexandrov, Y. Smartcheck: Static analysis of ethereum smart contracts. In Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain, Gothenburg, Sweden, 27 May–3 June 2018; pp. 9–16.
76. Krichen, M.; Lahami, M.; Al-Haija, Q.A. Formal methods for the verification of smart contracts: A review. In Proceedings of the 2022 15th International Conference on Security of Information and Networks (SIN), Sousse, Tunisia, 11–13 November 2022; IEEE: New York, NJ, USA, 2022; pp. 01–08.

77. Kushwaha, S.S.; Joshi, S.; Singh, D.; Kaur, M.; Lee, H.N. Systematic review of security vulnerabilities in ethereum blockchain smart contract. *IEEE Access* **2022**, *10*, 6605–6621. [CrossRef]
78. Li, Z.; Lu, S.; Zhang, R.; Xue, R.; Ma, W.; Liang, R.; Zhao, Z.; Gao, S. SmartFast: An accurate and robust formal analysis tool for Ethereum smart contracts. *Empir. Softw. Eng.* **2022**, *27*, 197. [CrossRef]
79. Cao, X.; Zhang, J.; Wu, X.; Liu, B. A survey on security in consensus and smart contracts. *Peer-to-Peer Netw. Appl.* **2022**, *15*, 1008–1028. [CrossRef]
80. Li, B.; Pan, Z.; Hu, T. EvoFuzzer: An Evolutionary Fuzzer for Detecting Reentrancy Vulnerability in Smart Contracts. *IEEE Trans. Netw. Sci. Eng.* **2024**, *11*, 5790–5802. [CrossRef]
81. Praitheeshan, P.; Pan, L.; Yu, J.; Liu, J.; Doss, R. Security analysis methods on ethereum smart contract vulnerabilities: A survey. *arXiv* **2019**, arXiv:1908.08605.
82. Feng, Z.; Feng, Y.; He, H.; Zhang, W.; Zhang, Y. A bytecode-based integrated detection and repair method for reentrancy vulnerabilities in smart contracts. *IET Blockchain* **2024**, *4*, 235–251. [CrossRef]
83. Gao, C.; Yang, W.; Ye, J.; Xue, Y.; Sun, J. sGuard+: Machine learning guided rule-based automated vulnerability repair on smart contracts. *ACM Trans. Softw. Eng. Methodol.* **2024**, *33*, 1–55. [CrossRef]
84. Senanayake, J.; Kalutarage, H.; Petrovski, A.; Piras, L.; Al-Kadri, M.O. Defendroid: Real-time Android code vulnerability detection via blockchain federated neural network with XAI. *J. Inf. Secur. Appl.* **2024**, *82*, 103741. [CrossRef]
85. Priya, J.C.; Rudzki, K.; Nguyen, X.H.; Nguyen, H.P.; Chotechuang, N.; Pham, N.D.K. Blockchain-Enabled Transfer Learning for Vulnerability Detection and Mitigation in Maritime Logistics. *Pol. Marit. Res.* **2024**, *31*, 135–145. [CrossRef]
86. He, N.; Wang, H.; Wu, L.; Luo, X.; Guo, Y.; Chen, X. A survey on eosio systems security: Vulnerability, attack, and mitigation. *Front. Comput. Sci.* **2025**, *19*, 196806. [CrossRef]
87. Yang, H.; Gu, X.; Chen, X.; Zheng, L.; Cui, Z. CrossFuz: Cross-contract fuzzing for smart contract vulnerability detection. *Sci. Comput. Program.* **2024**, *234*, 103076. [CrossRef]
88. Shou, C.; Liu, J.; Lu, D.; Sen, K. LLM4Fuzz: Guided fuzzing of smart contracts with large language models. *arXiv* **2024**, arXiv:2401.11108.
89. Zheng, Z.; Su, J.; Chen, J.; Lo, D.; Zhong, Z.; Ye, M. DAppSCAN: Building large-scale datasets for smart contract weaknesses in dapp projects. *IEEE Trans. Softw. Eng.* **2024**, *50*, 1360–1373. [CrossRef]
90. Cai, J.; Li, B.; Zhang, T.; Zhang, J.; Sun, X. Fine-grained smart contract vulnerability detection by heterogeneous code feature learning and automated dataset construction. *J. Syst. Softw.* **2024**, *209*, 111919. [CrossRef]
91. Victor, F.; Lüders, B.K. Measuring ethereum-based erc20 token networks. In Proceedings of the Financial Cryptography and Data Security: 23rd International Conference, FC 2019, Frigate Bay, Saint Kitts and Nevis, 18–22 February 2019; Revised Selected Papers 23; Springer: Berlin/Heidelberg, Germany, 2019; pp. 113–129.
92. Luu, L.; Chu, D.H.; Olickel, H.; Saxena, P.; Hobor, A. Making smart contracts smarter. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 254–269.
93. Taylor, P.J.; Dargahi, T.; Dehghantanha, A.; Parizi, R.M.; Choo, K.K.R. A systematic literature review of blockchain cyber security. *Digit. Commun. Netw.* **2020**, *6*, 147–156. [CrossRef]
94. Li, B.; Pan, Z.; Hu, T. Redefender: Detecting reentrancy vulnerabilities in smart contracts automatically. *IEEE Trans. Reliab.* **2022**, *71*, 984–999. [CrossRef]
95. Wu, G.; Wang, H.; Lai, X.; Wang, M.; He, D.; Chan, S. A comprehensive survey of smart contract security: State of the art and research directions. *J. Netw. Comput. Appl.* **2024**, *226*, 103882. [CrossRef]
96. He, D.; Wu, R.; Li, X.; Chan, S.; Guizani, M. Detection of vulnerabilities of blockchain smart contracts. *IEEE Internet Things J.* **2023**, *10*, 12178–12185. [CrossRef]
97. Yao, Y.; Duan, J.; Xu, K.; Cai, Y.; Sun, Z.; Zhang, Y. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confid. Comput.* **2024**, *4*, 100211. [CrossRef]
98. Homoliak, I.; Venugopalan, S.; Hum, Q.; Szalachowski, P. A security reference architecture for blockchains. In Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), Seoul, Republic of Korea, 14–17 May 2019; IEEE: New York, NY, USA, 2019; pp. 390–397.
99. He, Z.; Li, Z.; Yang, S.; Qiao, A.; Zhang, X.; Luo, X.; Chen, T. Large language models for blockchain security: A systematic literature review. *arXiv* **2024**, arXiv:2403.14280.
100. Venkatesan, K.; Rahayu, S.B. Blockchain security enhancement: An approach towards hybrid consensus algorithms and machine learning techniques. *Sci. Rep.* **2024**, *14*, 1149. [CrossRef]
101. Ali, S.; Li, Q.; Yousafzai, A. Blockchain and federated learning-based intrusion detection approaches for edge-enabled industrial IoT networks: A survey. *Ad Hoc Netw.* **2024**, *152*, 103320. [CrossRef]

102. Taher, S.S.; Ameen, S.Y.; Ahmed, J.A. Advanced Fraud Detection in Blockchain Transactions: An Ensemble Learning and Explainable AI Approach. *Eng. Technol. Appl. Sci. Res.* **2024**, *14*, 12822–12830. [CrossRef]
103. Kim, S.K. Automotive vulnerability analysis for deep learning blockchain consensus algorithm. *Electronics* **2021**, *11*, 119. [CrossRef]
104. Ibba, G.; Baralla, G.; Destefanis, G. Large Language Models for Synthetic Dataset Generation: A Case Study on Ethereum Smart Contract DoS Vulnerabilities. In Proceedings of the IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER) 2025, Montréal, QC, Canada, 4–7 March 2025.
105. Li, Z.; Zou, D.; Xu, S.; Ou, X.; Jin, H.; Wang, S.; Deng, Z.; Zhong, Y. Vuldeepecker: A deep learning-based system for vulnerability detection. *arXiv* **2018**, arXiv:1801.01681.
106. Kong, L.; Luo, S.; Pan, L.; Wu, Z.; Li, X. A multi-type vulnerability detection framework with parallel perspective fusion and hierarchical feature enhancement. *Comput. Secur.* **2024**, *140*, 103787. [CrossRef]
107. SWC Registry. Smart Contract Weakness Classification and Test Cases. Available online: https://swcregistry.io/ (accessed on 5 May 2022).
108. Hasan, M.; Rahman, M.S.; Janicke, H.; Sarker, I.H. Detecting anomalies in blockchain transactions using machine learning classifiers and explainability analysis. *Blockchain Res. Appl.* **2024**, *5*, 100207. [CrossRef]
109. Mounnan, O.; Manad, O.; Boubchir, L.; El Mouatasim, A.; Daachi, B. A review on deep anomaly detection in blockchain. *Blockchain Res. Appl.* **2024**, *5*, 100227. [CrossRef]
110. Mishra, S. Blockchain and machine learning-based hybrid IDS to protect smart networks and preserve privacy. *Electronics* **2023**, *12*, 3524. [CrossRef]
111. He, F.; Li, F.; Liang, P. Enhancing smart contract security: Leveraging pre-trained language models for advanced vulnerability detection. *IET Blockchain* **2024**, *4*, 543–554. [CrossRef]
112. Radanliev, P. Artificial intelligence and quantum cryptography. *J. Anal. Sci. Technol.* **2024**, *15*, 4. [CrossRef]
113. Gai, Y.; Zhou, L.; Qin, K.; Song, D.; Gervais, A. Blockchain large language models. *arXiv* **2023**, arXiv:2304.12749.
114. Brew, B. Can Artificial Intelligence Hack Blockchain Technology? 2023. Available online: https://medium.com/@blockchaindevelopment121/can-artificial-intelligence-hack-blockchain-technology-e26ac4f17c3 (accessed on 9 February 2024).
115. Knight, O. Ethereum Bot Gets Attacked for 20M as Validator Strikes Back. 2023. Available online: https://www.coindesk.com/business/2023/04/03/ethereum-mev-bot-gets-attacked-for-20m-as-validator-strikes-back/ (accessed on 12 April 2024).
116. Coghlan, J. This AI Chatbot Is Either an Exploiter's Dream or Their Nightmare. 2022. Available online: https://cointelegraph.com/news/this-ai-chatbot-is-either-an-exploiter-s-dream-or-their-nightmare (accessed on 19 August 2024).
117. Kayikci, S.; Khoshgoftaar, T.M. Blockchain meets machine learning: A survey. *J. Big Data* **2024**, *11*, 9. [CrossRef]
118. Palaiokrassas, G.; Bouraga, S.; Tassiulas, L. Machine Learning on Blockchain Data: A Systematic Mapping Study. *arXiv* **2024**, arXiv:2403.17081.
119. Azad, P.; Akcora, C.G.; Khan, A. Machine Learning for Blockchain Data Analysis: Progress and Opportunities. *arXiv* **2024**, arXiv:2404.18251. [CrossRef]

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

future internet

MDPI

Review

A Survey on Emerging Blockchain Technology Platforms for Securing the Internet of Things

Yunus Kareem [1,*], Djamel Djenouri [1] and Essam Ghadafi [2]

[1] Department Computer Science and Creative Technologies, University of the West of England, Bristol BS16 1QY, UK; djamel.djenouri@uwe.ac.uk
[2] School of Computing, Newcastle University, Newcastle-Upon-Tyne NE1 7RU, UK; essam.ghadafi@newcastle.ac.uk
* Correspondence: yunus.kareem@uwe.ac.uk; Tel.: +44-7502-583-020

Abstract: The adoption of blockchain platforms to bolster the security of Internet of Things (IoT) systems has attracted significant attention in recent years. Currently, there is a lack of comprehensive and systematic survey papers in the literature addressing these platforms. This paper discusses six of the most popular emerging blockchain platforms adopted by IoT systems and analyses their usage in state-of-the-art works to solve security problems. The platform was compared in terms of security features and other requirements. Findings from the study reveal that most blockchain components contribute directly or indirectly to IoT security. Blockchain platform components such as cryptography, consensus mechanism, and hashing are common ways that security is achieved in all blockchain platform for IoT. Technologies like Interplanetary File System (IPFS) and Transport Layer Security (TLS) can further enhance data and communication security when used alongside blockchain. To enhance the applicability of blockchain in resource-constrained IoT environments, future research should focus on refining cryptographic algorithms and consensus mechanisms to optimise performance and security.

Keywords: blockchain; Internet of Things (IoT); cryptography; consensus algorithm; ethereum; algorand; IoTeX; IoTA; multichain

1. Introduction

The Internet of Things (IoT) is one of the fastest-growing technologies with billions of devices connected to the internet for both personal and business purposes. IoT systems enable communication and information sharing between devices across diverse locations. The necessity for specially designed devices with sensing capabilities to exchange information over the internet gave rise to IoT, which has been embraced for numerous use cases and innovative applications such as smart healthcare [1,2], smart grid and building energy optimization [3,4], smart buildings [5,6], and smart transportation [7], among others. A summary of the IoT adoption statistics report published on the DataProt website reveals that the number of active IoT devices surged to 10 billion in 2021 and is projected to exceed 25.4 billion by 2030 [8]. Similarly, another projection estimated the rise of IoT devices from a million in 1992 to 50.1 billion in 2020, as shown in Figure 1 [9]. The market size for IoT was forecasted to reach 1.1 trillion dollars in 2023, marking the highest among other researched emerging technologies such as Artificial Intelligence and Edge Computing [8]. IoT applications span from the use of devices like smart light bulbs, smart locks, and security cameras in various homes and offices to more complex implementations in smart cities, traffic systems, and industrial IoT systems.

An IoT system can be classified as a product device that has a sensor, actuator, and communication medium, or a collection of multiple IoT devices with other components like application software, back-end services (cloud, API, etc.), and networking devices (gateway, access point, etc.) [10]. Abdmeziem et al. [11], Abou-Nassar et al. [10], and

Antao et al. [12] proposed different categorisations of IoT system components, including three-, four-, and five-layer models, respectively. However, all three models concur on the existence of fundamental components within the IoT system, such as the Application, Network, and Perception/device-sensing layers. Additional models may include layers such as the middleware layer and the business layer, as shown in Figure 2.

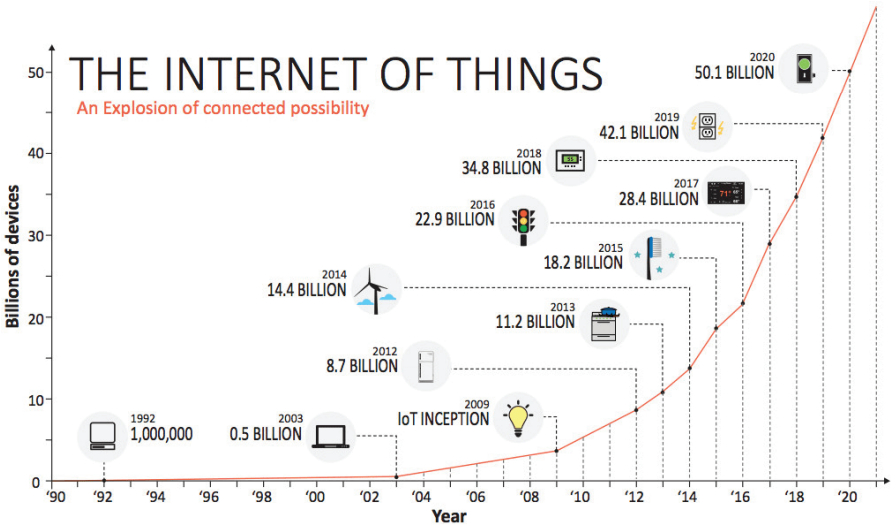


Figure 1. Rise in connected devices [9].

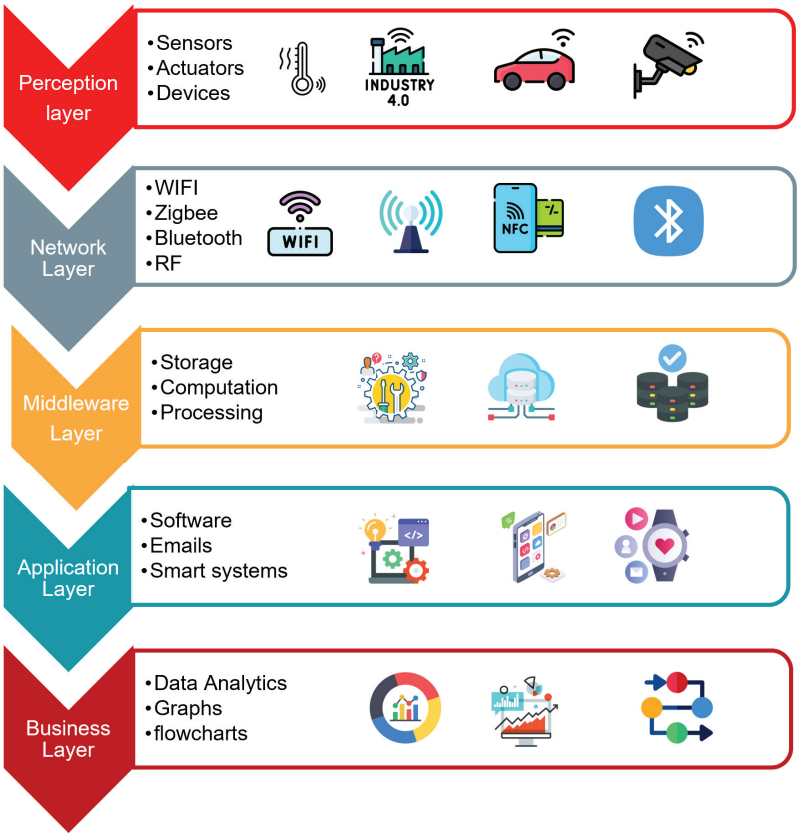


Figure 2. IoT system layers.

IoT systems face several challenges. Harit et al. [13] categorised these challenges into privacy and security, interoperability, scalability, heterogeneity, naming, and addressing. Among these challenges, privacy and security stand out as the primary concerns in IoT,

with many of them manifesting at the network layer [13]. In 2021, statistics on IoT attack revealed a disturbingly high frequency of cyber attacks, numbered in the billions. This underscores the need for stringent measures to safeguard IoT networks and protect user information [14]. Mamdouh et al. [15] classified attacks on IoT systems into Goal-oriented, performer-oriented, and layer-oriented. These attacks have been a significant source of insecurity in IoT systems. Various technologies including artificial intelligence and blockchain have been deployed to address these threats [15]. The current work focuses on blockchain.

Blockchain is generally referred to as a tamper-proof distributed ledger that is immutable and securely linked by cryptographic hashing [16]. Each block contains a unique value called a nonce, which can only be used once. It also includes data, which represent the transactions saved on the blockchain, and a hash, which is a mathematical function that converts a variable-length (generally large) input string into a fixed-length output string. Blockchain technology has revolutionised the security and privacy of applications and devices by creating an interconnected and decentralised system that uses blocks to store information [17,18]. Figure 3 shows the components of a blockchain. Blockchain transactions are managed using a wallet and are protected by digital signatures. These transactions are temporally saved in the mempool until it is nominated and agreed upon via consensus to be added to the blockchain. Blockchains provide a new mode of distributed data storage, point-to-point transmission, consensus mechanisms, and encryption, with features such as decentralisation, openness, tamper-proofing, anonymity, and traceability [19].

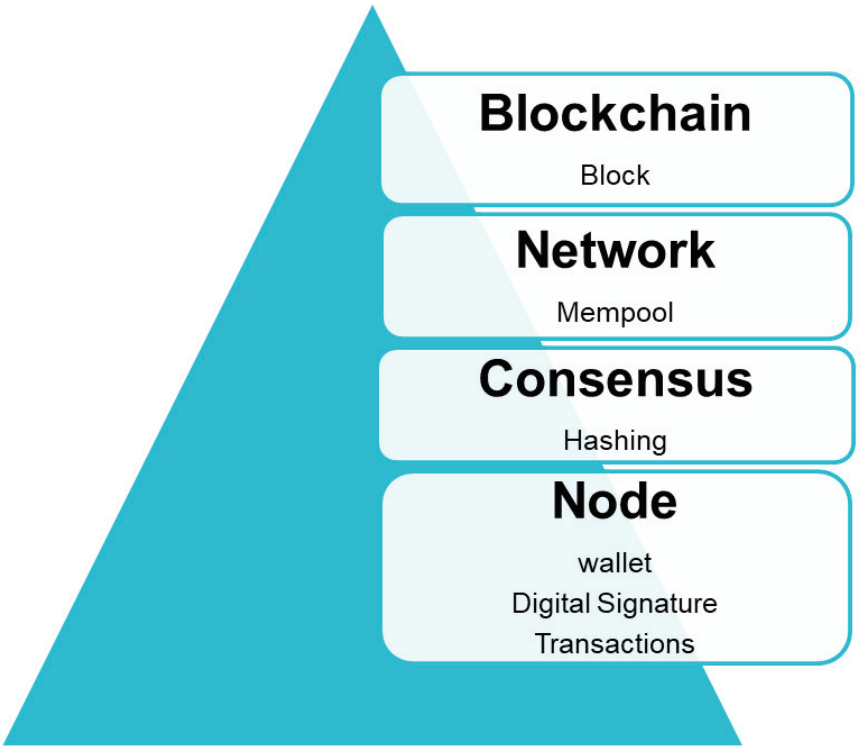


Figure 3. Blockchain framework components.

The effectiveness of blockchain technology in ensuring security, privacy, and reliability is well-established, which has led to its adaptation in various fields, including finance, e-government, IoT, public services, and supply chains. Numerous studies have delved into security and privacy issues in blockchain technology. Determining the most suitable blockchain type for an IoT ecosystem continues to pose a significant challenge in the field. This difficulty arises primarily from the inherent resource limitations of certain IoT devices, which require a careful consideration of the technology's adaptability. Consequently, the industry has witnessed the emergence of specialised solutions such as lightweight

blockchain for IoT and half-node (pruned) blockchain for IoT [20]. These innovative approaches entail the deployment of either full-node (which takes part in all blockchain activities and stores all blocks) or pruned node (which only stores recent blocks and removes old blocks to manage storage space) blockchain structures within the IoT systems or the gateway of the IoT network, facilitating the seamless integration of blockchain technology with IoT devices [21]. As a result of this integration, the designated blockchain nodes are capable of collecting pertinent data from the IoT environment, which are then securely shared among the interconnected nodes within the blockchain network, enhancing the system's overall efficiency, security, and reliability. This has also invariably led to high resource consumption of such IoT systems [22].

This paper surveys emerging distributed ledger/blockchain platforms adopted to enhance security in IoT. It explore how these platforms can address vulnerabilities by securing data exchange between devices, managing device identities with tamper-proof mechanisms, and ensuring the integrity of transactions throughout the interconnected world of IoT. This is achieved by examining the frameworks, blockchain types, methodologies employed to ensure security at each layer of IoT, consensus algorithm, cryptography technology, and their limitations and strengths. The paper addresses the following research questions:

1. What are the emerging blockchain technologies/platforms/frameworks used in IoT security, their components, and how have they been modified for resource-constrained IoT devices?
2. Which of the components above can work across all IoT devices without impacting security?
3. What other technology can boost the security of IoT when added to the blockchain with less or no impact on performance?

To address these concerns, this study conducted a thorough search of related work on large database platforms via a digital library (Primo and Science Direct) and indexers (Google Scholar and SciSpace AI search assistance). A total of 14,745 journals were found using "IoT Security using blockchain". The results were filtered by removing duplicates, considering research work from high-profile research platforms (IEEE Xplore UK, ACM Digital Library US, Springer US, Elsevier Netherland, and MDPI Switzerland). The search was further screened by filtering with the following keywords: "Blockchain", "IoT", "Blockchain IoT", "Blockchain IoT Security", "Blockchain Cryptography in IoT Security", "survey of blockchain IoT security", "review of IoT security with blockchain", and "Consensus Mechanism in Blockchain IoT security", obtaining 1246 works. The results were assessed using their abstracts and methodology to select the 146 papers used in this work. Due to the systematic approach of this research, which considered works based on a definite blockchain platform used in IoT security, 46 works were examined under six blockchain platforms, 15 related works consisting of review and survey papers, and the rest of the papers were used to complement the research finding at different section of the paper. The methods are represented using the PRISMA literature search flow, as shown in Figure 4. The rest of the paper is organised as follows: Section 2 presents the related work. It analyses 15 selected reviews and survey papers on blockchain-based IoT security and positions the paper's contribution with a comparison table to reveal the existing gap. Section 3 presents the different blockchain platforms and discusses their usage in the IoT security literature. Section 4 provides an analysis and discussion. Finally, Section 5 concludes the paper with recommendations for future works.

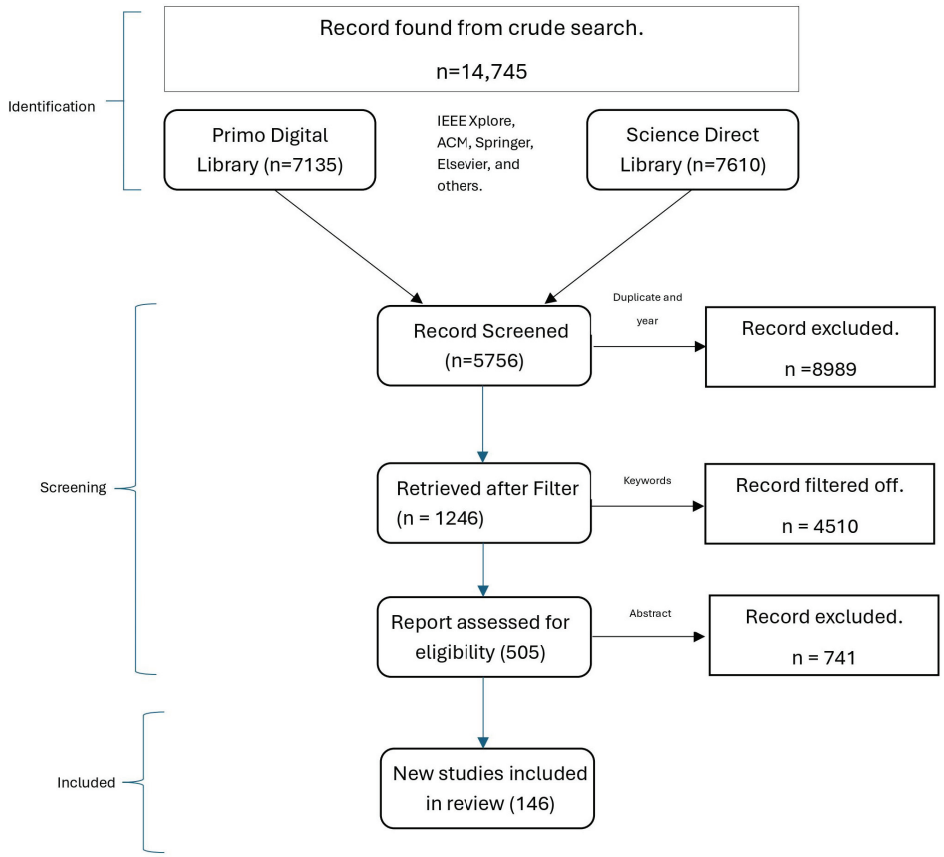


Figure 4. PRISMA literature search flow.

2. Related Work

This section provides a summary of existing works and a table to compare 15 review and survey papers, comprising four journal articles and 11 conference papers. Most of these papers delve into fundamental aspects of blockchain and IoT, examining intersections and contemporary research directions. Many of these papers expand on their analyses by concentrating on specific areas or topics (tagged Focus) within IoT-BC systems.

Adaptation of Blockchain (BC) in IoT, its challenges, and solutions are a major focus of [23–25] . Wang et al. [24] additionally focused on Industrial IoT (IIoT). These papers discuss the basic structure and main features of blockchain and its application in the IoT and IIoT. Abdelmaboud et al. [25] reviewed the use of blockchain in IoT applications and provided a taxonomy that covers various aspects such as platforms, recent advances, challenges, and future research directions. Their work gave insights into several blockchain platforms and the contribution they can make to IoT networks. Lao et al. [26] provided an overview of IoT blockchains, including their architectures, communication protocols, applications, and traffic models. They discussed the network structures and protocols of popular IoT blockchain systems, compared different consensus algorithms, and analysed the traffic distribution in P2P and blockchain systems. They also emphasised the need for regulation and development policies to guide the integration of IoT and blockchain technology.

Overviews and reviews of IoT-BC integration and use-cases can be found in [27–29]. Sultan et al. [29] and Kumar et al. [30] additionally investigated issues encountered after IoT blockchain integration, focusing on aspects like system efficiency in terms of resource limitations and scalability. Chowdhury et al. [27] reviewed blockchain-based platforms

from the perspective of IoT use cases. They introduced an evaluation framework to select a suitable blockchain platform for a given IoT application based on its specific requirements. They discussed different IoT use cases and their functional and non-functional requirements, emphasising the need for a clear understanding of the requirements to identify an appropriate blockchain platform. Sadawi et al. [28] discussed the integration of blockchain with IoT networks. They evaluated current research in this area and proposed an architectural framework that is structured around a system comprising three distinct layers: the device layer, the dew-blockchain layer, and the cloudlet-blockchain layer.

Security perspectives surrounding IoT-BC integration are seen in [30–34]. Darla and Naveena [34] conducted a critical review of blockchain-based wireless sensor networks (WSN), focusing on the security concerns in WSN and the adoption of blockchain in comparison with other research reviews. Kumar et al. [30] and Shammar Zahary and Al-Shargabi [33] reviewed security problems in IoT, the proposed solutions using blockchain, the status of blockchain adoption in IoT, and their challenges.

Table 1 provides a comparison of selected state-of-the-art surveys and review papers related to IoT security using blockchain. These papers were carefully looked at, considering their key focus (which represents the authors' research focus), reviewed work count (which is the number of works reviewed by their author), limitations, and recommendations for future work. There was a clear gap in the security perspective of IoT-BC in the work; also, the components of BC that enhance security in IoT are given less attention and, in some works, not discussed. These are part of the issues addressed in this work. While the existing reviewed papers focused on a few concepts of security benefits in blockchain IoT, the current paper goes in depth. The existing review identifies six emerging distributed ledger platforms, including five blockchain platforms and one DAG platform. These platforms were presented by conducting a detailed review of the works that use them in IoT systems security, emphasising security components.

Table 1. Comparison of survey and review papers from 2020 to 2024.

S/N	Work	Reviewed Work Count	Focus	Limitation	Recommendation
1.	[35]	98	Lightweight blockchain	No discussion on BC-IoT security	Recommend further research on security and efficiency
2.	[36]	N/A	Data transfer and storage in IoT-BC. Attacks and designs spaces in IoT-BC network	Work is more of a definition and explanation of network-related questions on BC-IoT	Not provided
3.	[32]	N/A	Security threat in IoT and solutions using BC	Blockchain solutions are not discussed in detail	Addressing open issues in IoT security
4.	[30]	8	IoT-BC architecture for IoT Security	No detailed reference to existing IoT-BC work and their challenges	A new, lightweight framework for IoT-BC
5.	[29]	7	IoT Issue and characteristics of BC that can solve them	No detail information on previous work and challenges	Practical implementation of IoT-BC system
6.	[31]	13	Summary of selected work	No technical discussion on integration and challenges	More framework and methods should be introduced for IoT-BC integration
7.	[24]	19	Application of BC in IoT and Industrial IoT	No element of security discussed	No recommendation

Table 1. *Cont.*

S/N	Work	Reviewed Work Count	Focus	Limitation	Recommendation
8.	[25]	20	Security and privacy issues and challenges in IoT-BC Integration.	Lacks detailed explanation on how the security and privacy are achieved	Future work can discuss in depth how to achieve privacy and security in IoT-BC
9.	[23]	7	IoT-BC application	Nothing on security	Not provided
10.	[37]	37	IoT challenges and performance of IoT-BC solutions	Security components in IoT-BC not covered	Proposes adopting dew and cloudlet layer architecture for IoT-BC
11.	[33]	81	Issues and trends in IoT-Blockchain security perspective	Focus on IoT security works using BC from 2017 to 2021	Explores how BC, edge computing, and IoT integration can improve security
12.	[27]	10	Brief review of selected IoT applications and IoT-BC use-cases	Unable to identify a most efficient platform for low resource device	To develop an architecture suitable for all IoT devices irrespective of their size
13.	[26]	N/A	Architecture, consensus, and traffic modelling	Security and platform choice	Security, regulation, and policy development should be addressed for IoT-BC
14.	[28]	15	Summary of works on IoT challenges and BC applications to IoT	Security perspective of IoT-BC not covered	None
15.	[38]	100	Paper review and Hyperledger Saw-tooth for Industrial IoT	Regulatory and compliance for IoT-BC not addressed	Cross-chain for IoT-BC, industrial standardization, distributed preservation, and privacy issues
16.	This	61	IoT-BC Platform and Security components	Limited to works from 2020 to 2024	Improvement on security component in IoT-BC integration

3. Blockchain Platforms

Six distributed ledger/blockchain platforms found to be the most used technology for enhancing security in the Internet of Things (IoT) were analysed in this work, following a thorough assessment of 146 studies. Figure 5 depicts the extent of their usage across the entirety of the works that were reviewed.

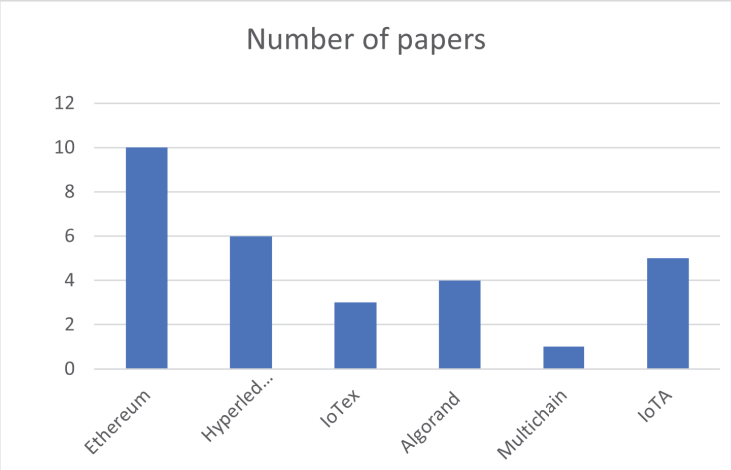


Figure 5. Blockchain platform used in research papers covered in this study.

3.1. Ethereum Blockchain

Ethereum, introduced in late 2015 by Vitalik Buterin, is one of the most distinguished blockchain platforms and frameworks for developing decentralised applications (DApps). The enhancement of security in IoT systems and the necessity for decentralised IoT systems resulted in the integration of Ethereum into IoT ecosystems. This integration was evident in various research studies, showcasing its utilisation across industries like Supply Chain, Healthcare, Transportation, Finance, and Agriculture. Ethereum's robust security stems from its carefully chosen components, including the consensus algorithm, smart contracts, encryption methods, and decentralised network. The consensus algorithm, transitioning to Proof of Stake with Ethereum 2.0, ensures transaction validation and network security. Smart contracts, which are self-executing agreements coded into the blockchain, enhance security by automating contract enforcement. Strong encryption methods like public-key cryptography safeguard data privacy. The decentralised network structure, powered by globally distributed nodes, eliminates single points of failure, bolstering security and resilience against potential threats. These elements collectively contribute to Ethereum's reputation as a secure platform for decentralised applications and digital transactions.

Numerous investigations and initiatives have delved into the application of the Ethereum blockchain in bolstering the security of Internet of Things (IoT) systems. The essence of these endeavours lies in utilising a decentralised, unalterable distributed ledger technology to facilitate secure and openly accessible data management for IoT gadgets. This strategy cultivates a state of affairs devoid of reliance on trust within the IoT environment, promoting transparency and guaranteeing data integrity, culminating in a more robust and dependable communication network among devices. Jabbar et al., Hussein et al. and Bawanar et al. [39–42] are examples of work that tend toward harnessing the security benefit of IoT–Ethereum integration. Rateb et al. [39] used this integration to secure vehicle communication through their proposed decentralised cloud computing platform. Patil et al. [43] adopted this integration to solve a major problem in the vehicle rental system. Other works such as [44–46] experimented on ethereum integration with IoT systems. The modification and addition of technologies were also discussed in some works such as [47], whose work adopted PoA for consensus in place of PoS/PoW. Fog computing and Interplanetary File systems were also introduced in [48] and [49], respectively.

3.2. Hyperledger

Hyperledger occupies a significant and influential position in blockchain technology and provides a broad spectrum of tools and frameworks tailored to cater to the distinct requirements of corporations and institutions. Some of its frameworks include Hyperledger Fabrics, Sawtooth, Indy, and Besu [50]. Like ethereum, Hyperledger's components include encryption (Digital signature and Hashing), chaincode, and Zero Knowledge of Proof (ZKP). Hyperledger is one of the most diversified blockchain frameworks. It redefined blockchain ecosystems by introducing lots of flexibility, such as choosing a custom consensus mechanism, two ledgers for blockchain queries, cryptography to secure transactions, decentralised identities, support for SDKs and APIs, and network monitoring [51]. Its notable influence has been seen in many studies lately for IoT security with blockchain.

Alshehri and Bamasag [52] identified insecurity with IoT access control and proposed a solution using Hyperledger Fabric. This solution employs attribute-based access control, where several attributes were determined based on the sensitivity of the data. The authors report the performance of the proposed system by testing it with the iFogSim simulator. Another work [53] dealt with protecting IoT system devices against attacks associated with firmware updates of embedded IoT devices. Their system uses blockchain as a trusted network to ensure the authenticity and integrity of firmware updates. CATP-Fabric was proposed as a new blockchain platform to solve issues around conflicting transaction problems in Hyperledger Fabric for delay-sensitive IoT applications. This work divides transactions into diverse groups for parallel processing, filters stale transactions, and prioritises read-only transactions to reduce overhead. The results illustrated show that

CATP-Fabric achieves a high throughput of successful transactions while maintaining a lower aborting transaction rate compared to benchmark blockchain systems.

To counter the latency issue that is common with IoT blockchain systems, Lee et al. [54] proposed a latency distribution model for Hyperledger Fabric (HLF) using probability distribution fitting and showed that the latency can be modelled with the gamma distribution. The paper further presents three significant parameters of high-level features and examines their effects on latency. It offers an approach to reduce the average latency of HLF. The model and analysis put forward in this study can be employed to predict the latency of HLF before the actual implementation of blockchain Internet of Things (BC-IoT) networks, thereby addressing the outstanding issue of latency estimation in [50]. Pajooh et al. [50] leveraged smart contracts in HLF to propose a model addressing the scalability, processing power, and storage limitation of IoT edge devices in the blockchain network. HLF is a permission blockchain that has been successfully proposed as a solution to several IoT system issues. Its flexibility has been leveraged in several works.

3.3. IOTA Blockchain

IOTA is a distributed ledger platform that employs a Directed Acyclic Graph (DAG) structure, as opposed to a traditional blockchain. Its intended use is for the Internet of Things (IoT) ecosystem, with an emphasis on scalability, low fees, and secure data transfer between devices [55]. The IOTA platform comprises numerous key features and components, including Tangle Technology, Zero Fees, Decentralised Consensus, MAM (Masked Authenticated Messaging), IOTA Tokens (MIOTA), Smart Contracts (IOTA Smart Contracts—ISCP), Qubic, and Decentralised Identity (DID) [56].

The technology uses a DAG-based structure called the Tangle, allowing for parallel processing and scalability, while Zero Fees ensures a feeless environment. Decentralised Consensus is achieved through a process called "Tip Selection", thus eliminating the need for miners or validators [57]. The MAM protocol enables private communication and data integrity. IOTA Tokens (MIOTA) are utilised for value transfer and data transactions within the Tangle. IOTA is developing smart contract functionality through the IOTA Smart Contracts Protocol (ISCP), which enables developers to create decentralised applications and automate agreements on the IOTA network [58]. The IOTA development caught the attention of some researchers such as [55,56,58]. They all focused on identifying the gaps in IOTA to further improve it.

Khan et al. [38] adopted IOTA with Edge computing (EC) to tackle the obstacles of real-time processing, resource allocation, and storage provision in IoT devices. They conducted an in-depth survey to identify recurring problems with IoT systems and provided a counter solution to it using EC and IOTA blockchain. Additionally, IOTA explores decentralised identity solutions for secure and private identity management. IOTA has partnered with various companies and organisations to explore real-world applications of its technology, including supply chain management, smart cities, energy management, and more.

3.4. IoTeX Blockchain

The IoTeX blockchain platform was custom-built to cater to the specific needs of the IoT ecosystem. It was developed to overcome the obstacles and constraints faced by current blockchains in meeting the distinct demands of IoT devices and applications. The IoTeX blockchain platform encompasses several critical facets and elements [59].

The architecture of IoTeX consists of multiple interconnected blockchains, including the Root Chain and multiple sub-chains customised for specific use cases. The Root Chain serves as the main blockchain responsible for network consensus and security. The consensus algorithm employed by IoTeX is Roll-DPoS, which enhances security, scalability, and decentralisation. Various token standards, such as IOTX, XRC20, and XRC721, are supported by IoTeX, and tailored for IoT use cases [59].

IoTeX achieved high scalability using sub-chains, which allows for parallel processing and reduces congestion on the main chain. The platform also incorporates advanced

privacy features like secure data transfers and lightweight encryption. A decentralised identity framework is offered by IoTeX, which enhances security and control over personal data. The Internet of Trusted Things concept was introduced by IoTeX, focusing on building secure and trusted interactions between devices and users [59].

IoTeX employs innovative cryptography methods to generate secure and unpredictable random numbers, which is essential for certain use cases in IoT. The platform is designed to be compatible with other blockchains, enabling cross-chain communication and interoperability. A growing ecosystem of developers, partners, and community members is working together to build and expand the platform's capabilities and use cases.

The specialised focus of IoTeX on IoT and its architecture were designed to address IoT-related challenges and distinguish it from other general-purpose blockchain platforms. The platform strives to provide a secure and scalable foundation for various IoT applications, including supply chain management and data marketplaces [60]. Fan et al. [61] found that works on IoTeX have shown the strength of the technology in addressing the key challenges in integrating IoT with blockchain. Fan and his team published a paper after developing a user-centric Blockchain-based secure IP camera, which was built on the IoTeX platform. Other works were seen from different domains, such as Wearables [62] and Mobile Payment [63].

3.5. Algorand

Algorand was introduced in 2019 by Silvio Micali and specifically engineered to prioritise reduced latency, security, and decentralisation [64]. This is achieved by employing inventive consensus mechanisms and cryptographic methodologies. Security components in the Algorand blockchain are cryptographic Sortition (Verifiable Random Function), PoS consensus mechanism, and other encryption mechanisms (Hash Function and Edwards-Curve Digital Signature (EdDSA)). Algorand is suitable for various blockchain applications, including financial services, supply chain management, and IoT [65]. The Algorand Blockchain is gaining prominence as a promising technology for the Internet of Things (IoT) owing to its self-sustaining and decentralised characteristics [66]. It has been implemented in low-powered IoT devices such as the Raspberry Pi 4 Model B and the STMicroelectronics STM32MP157A-DK1 [66]. A multi-level blockchain structure and consensus algorithm have been put forth to enhance the security and dependability of IoT data management [24,67]. Additionally, blockchain-based data verification schemes can safeguard data integrity in IoT by employing distributed data authentication and lightweight mining processes, as proposed by Wadhwa et al. [65] in their work. Cardamone et al. [68] used Algorand together with Messaging and Queues Telemetry Transport (MQTT), and Transport Layer Security (TLS), to solve a major authentication issue in an IoT electrical energy system.

3.6. Multichain

Multichain is an example of a cross-chain platform, which supports over 26 blockchains including Ethereum and Bitcoin. Cross-chain refers to integrating various forms of blockchain technologies into interrelated blockchains within a unified network or ecosystem. Although there are few works in this domain, there are some works exploring this concept in the IoT ecosystem. Multichain comes with its components, such as blockchain networks, Nodes (Full Node, API Node, and Mining Node), Assets, Streams, Permissions, Mining controls, etc. This is the building block of the Multichain blockchain. Multichain comes with the flexibility to choose from different consensus mechanisms, with support for PoW, PoS, DPoS, Round Robin, Byzantine Fault Tolerance and Custom Consensus Mechanism. This can selected during the network set-up. Multichain networks in IoT were adopted by [69,70] who aimed at reviewing the existing work and experimenting with multichain integration in IoT. Sawant et al. [70] developed a prototype system using Raspberry Pi and Savoir Rapper for multichain. Umran et al. [71] explored multichain to secure IoT in the petroleum application. They reviewed existing work in the domain and proposed a new architecture comprising an IoT system, multichain, and an interplanetary file system (IPFS). Synergy

chain is a multichain framework for IoT blockchain. To address the data-sharing and access controls, Chang et al. [72] proposed this.

4. Analysis and Discussion: IoT Security and BC Components

IoT insecurity can best be summarised in terms of authentication, privacy, data storage, and update delivery. Blockchain is a trust- and privacy-inclined technology, with every component contributing to the security and privacy of IoT in one way or another.

4.1. Technology

This work focuses on six trending and emerging blockchain technologies used in IoT security. We have seen Ethereum as the choice of most researchers. Ethereum is the most used technology, as illustrated in Figure 6. Another technology that has gained much momentum in adoption is Hyperledger. Other platforms include IoTA, IoTeX, and Algorand. From our findings, we observed large community support on both Ethereum and Hyperledger from individuals and organisations who are actively improving the framework. This may justify their popularity within the research community. Also, we noticed that the adoption of Ethereum in the IoT space started quite recently. This can be attributed to the adoption of a new consensus mechanism (PoS), which reduces power consumption significantly without impacting its security. Table 2 is a summary of selected works from the six blockchain technologies reviewed, their area of applications, additional technology used to support their development, and their choice of consensus algorithm. From this table, it is observed that there is a trend towards supporting IoT-BC integration with cloud services, external BC file storage such as IPFS, and creating web interfaces for easy management. Cloud storage has helped IoT systems in terms of data storage but does not guarantee security. Blockchain storage Technology such as IPFS has been a good replacement. Other security enhancement technologies introduced to support resource-constrained IoT-BC integration include the TLS protocol and Pebble Tracker.

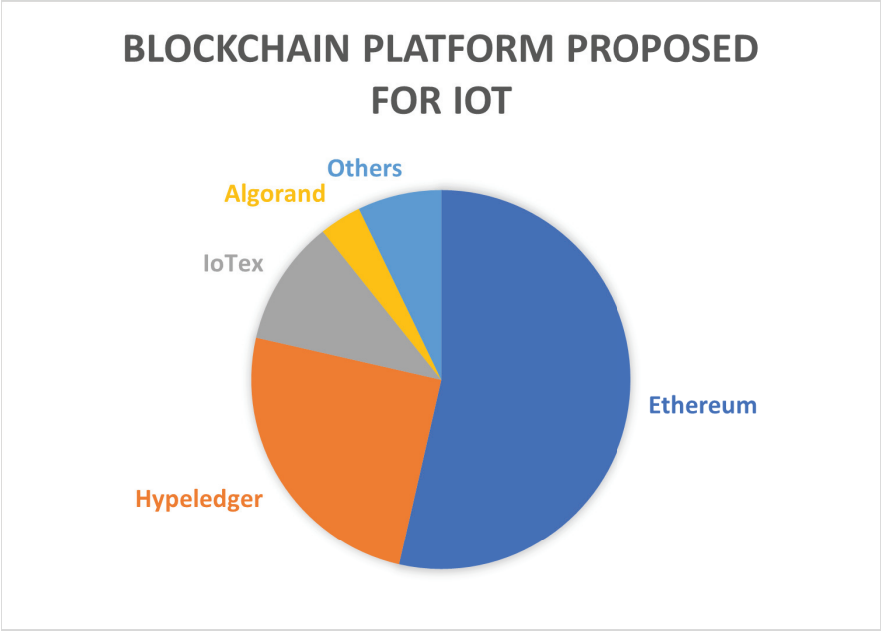


Figure 6. Blockchain platform for IoT.

Table 2. Technologies in selected IoT-BC existing works and their areas of applications.

S/N	Work	Areas of Application	Technologies	Blockchain Platform	Consensus
1.	[48]	Healthcare	Cloud and Fog computing	Ethereum	PoS, POW , PoA
2.	[40]	Authentication and Access control	None	Ethereum	POS
3.	[39]	Vehicle communication	Cloud computing	Ethereum	PoW
4.	[41]	Voting System	Web	Ethereum	PoW
5.	[73]	E-Business	None	Ethereum	PoW and PoA
6.	[74]	Embedded System	FGPA	Ethereum	PoW
7.	[42]	Data Counterfeiting	Cloud	Ethereum	PoW
8.	[75]	Decentralised Authentication	web	Ethereum	PoW
9.	[76]	Key Management	TLS Protocol	Ethereum	PoW
10.	[47]	Networks Attacks	None	Ethereum	Proof of Authority
11.	[49]	Data Sharing	Interplanetary File System	Ethereum	PoS
12.	[52]	Access Control	Lightweight cryptography	Hyperledger Fabric	PBFT
13.	[53]	Firmware Update	None	Hyperledger Fabric	PBFT
14.	[63]	Experimental IoT	None	Hyperledger Fabric	PBFT
15.	[50]	Edge Internet	None	Hyperledger	PBFT
16.	[77]	Access Control	None	IoTA	DAG
17.	[78]	TangleSim	None	IoTA	DAG
18.	[55]	Access Control	Lightweight cryptography	IoTA	DAG
19.	[58]	IOTA	None	IoTA	DAG
20.	[62]	Wearable	None	IoTeX	Roll-DPoS
21.	[79]	Data Authorisation	Pebble Tracker	IoTeX	Roll-DPoS
22.	[63]	Mobile Payment	None	IoTeX	Roll-DPoS
23.	[61]	Home IP Camera System	Intel SGX	IoTeX	Roll-DPoS
24.	[68]	Electrical Energy System, Authentication	MQTT, TLS, and CA	IoTeX	Roll-DPoS
25.	[66]	Prototype	Raspberry Pi and STM32	Algorand	Pure PoS
26.	[68]	Electrical Energy System, Authentication	MQTT, TLS, and CA	Algorand	Pure PoS
27.	[69]	None	None	Multichain	N/A
29.	[80]	Authentication	None	Multichain	DAG
30.	[71]	Industrial IoT	IPFS	Multichain	Proof of Rapid Authentication
31.	[70]	Security	None	Multichain	NA

4.2. Consensus

The Consensus mechanism provides trust in the blockchain, ensuring that every block is verified before being added to the blockchain. This has invariably been used in IoT-Blockchain to ensure data integrity in IoT system communication. Achieving consensus follows three processes: proposal (where a node proposes a block to be added), verification (another node validating the proposed block), and agreement (the majority node agrees that the proposed block is valid). This process employs algorithms such as Proof of Work (PoW), Proof of Stake (PoS), Byzantine fault tolerance (BFT), Proof of Authority (PoA), Practical

Byzantine Fault Tolerance (PBFT), and Raft to achieve the agreement. This agreement helps to ensure the right data are added to the blockchain, especially in a large IoT system where several IoT systems depend on each other's data. It also guides against malicious actions by preventing unauthorised tampering and easily spotting malicious nodes. Research has shown that consensus algorithms also have flaws, and some of these are less effective in small IoT networks due to the required resources to carry out the attack.

For IoT security using blockchain, default consensus algorithms are adopted on most platforms based on IoTeX, Algorand, Ethereum, and Hyperledger. A few exceptions also exist where other consensus mechanisms were proposed to support resource-constrained IoT devices. Figure 7 illustrates the consensus algorithms from the perspective of their use in IoT research papers covered in this study. From the reviewed works, the mechanism of achieving consensus by each algorithm was explained, along with some attacks that each algorithm can be vulnerable to. This is summarised in Table 3. Six algorithms are carefully selected as the most used in the IoT blockchain. The comparison of consensus algorithms in Table 3 highlights that many are susceptible to attacks like Sybil attacks, 51% attacks, and long-range attacks. However, these attacks often require significant resources, making them less of a concern for integrating blockchain with resource-constrained IoT devices.

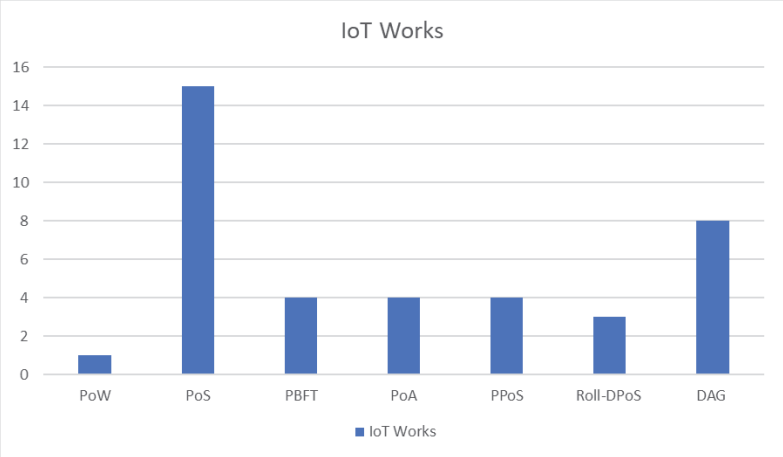


Figure 7. Consensus mechanisms in the studied literature.

Table 3. Consensus algorithm properties and threats.

S/N	Consensus	Blockchain	Mechanism	Threat	Throughput
1.	PoS	Ethereum	Locked assets	Sybil & 51% attack	20 TPS
2.	Pure PoS	Algorand	Random selection	long range attack	6000 TPS
3.	DAG PoW	IoTA	Tip Selection	34% attack	12 TPS
4.	Roll- DPoS	IoTeX	Random selection	Collusion Attacks	10,000 TPS
5.	Round Robins	Multichain	Rotational	Nothing at stake attack	2 million TPS
6.	PBFT	Hyperledger	Voting	1/3 faulty node & DoS	20,000 TPS

4.3. Cryptography

Another key security component for blockchain-based IoT security solutions is cryptography. Cryptography contributes highly to the security of information in blockchain. By employing cryptographic mechanisms including hashing, digital signatures, zero-knowledge proof, and encryption, blockchain systems can foster increased levels of security and trust among users and things. This has been adopted for IoT security in areas of authentication, data encryption during communication, and data validation. Existing

blockchain technology has been seen to use different cryptographic settings for digital signatures, e.g., Ethereum uses an Elliptic Curve Digital Signature Algorithm (ECDSA), which is a type of Elliptic Curve Cryptography (ECC); others are the Edward Digital Signature Algorithm (EdDSA) and Rivest–Shamir–Adleman (RSA). ECDSA is adopted in most works, and while some of these works proposed a lightweight ECC to adapt it to resource-constrained IoT devices, this is not recommended as it impacts the security strength of the device. Cryptographic Sortition is another approach used by Algorand. The Winternitz One-Time Signature (WOTS) and Merkle Signature Scheme were also used and are considered quantum resistant. Figure 8 illustrates the percentage breakdown of reliance and security contribution provided by the predominant key tools across the six platforms under examination. The calculation of this percentage involves a standard fixed allocation of 30% for all major security tools, namely, consensus, digital signature, and hashing. The remaining portion is derived from the collaborative security features integrated to augment the overall security framework of the platforms.

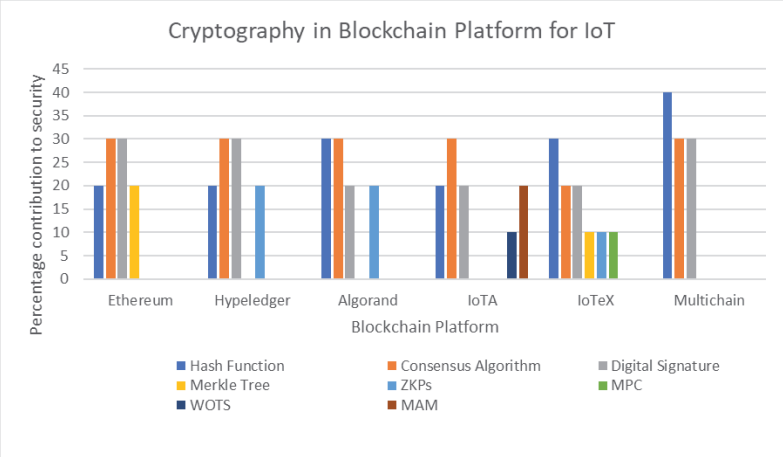


Figure 8. Cryptography in Blockchain platform for IoT.

5. Effect of Blockchain

In an IoT network, devices can communicate and transfer data autonomously. Transferred data may be private and highly confidential in many applications, e.g., healthcare, defence, smart buildings, etc., which amplifies the security risks of such systems. Blockchain technology has revolutionised the IoT system by addressing these security risks (such as weak authentication, insecure communication, and unreliable updates). Through decentralised identity, encrypted transactions, immutable records, and data distribution, blockchain ensures the integrity, authenticity, and protection of IoT systems. Its resistant nature to manipulation further solidifies its role in safeguarding IoT ecosystems. The adoption of blockchain also has both security and performance effects on IoT systems.

5.1. Security Effect

Table 4 presents a few of the most significant security issues that blockchain technology has helped to tackle. It emphasises the technology of blockchain that was utilised to solve the security challenges as well as the research that led to the development of the solution. Considering the security concerns of weak authentication and unrestricted access to systems, decentralised authentication and access control on a permissioned blockchain like Ethereum is an intriguing solution that was proposed in [40]. This method utilises an "authorisation sensitivity factor" stored on the blockchain. This factor determines a client device's access level, eliminating the need for a central authority to manage permissions. This system was tested against distributed denial of service attacks (DDOS) and was found to be secure against them. Another good example is delivering secured firmware update to devices. Bettayeb et al. [53] adopted blockchain as the secure network to deliver the

updates to various devices on the chain whose firmware versions are verified for already validated firmware files. Similar to the above, Jeyakkannan et al. [47] enhanced IoT network communication by adopting blockchain as the medium of exchange of information between devices. Data and privacy protection are critical challenges in IoT systems. To address this, researchers in [49] proposed a three-phase approach. This involves establishing secure authentication through decentralised blockchain-based authentication (DBA), safeguarding data with blockchain encryption, and ensuring data integrity through secure decryption. Insufficient privacy protection, which is common in IoT systems, was also addressed in [42] by encrypting the identity of devices on the block and managing access to information using data access control and auditable data provenance of ethereum blockchain. Blockchain offers a secure architecture for IoT applications, but that does not mean it is immune to security challenges. Resource constraints in some IoT devices might limit the feasibility of certain attacks, but others remain a significant concern. These include Sybil attacks, 51% attacks, replay attacks, wallet poisoning, and smart contract vulnerabilities. Carefully evaluating these risks is crucial when choosing a blockchain platform for IoT security.

Table 4. Breakdown of IoT security enhancement with BC-components.

S/N	Security Challenges	Research Works	Blockchain Technologies
1.	Weak Authentication and Uncontrolled Access	[40,55,75]	Immutable Ledger, Decentralised Identity, Digitally signed Wallet
2.	Insecure Network Services and Communication	[47]	Encrypted Communication, Smart Contracts/ Chain-code.
3.	Lack of Secure Update Mechanism	[53]	Immutable Update History, Decentralised Distribution
4.	Insufficient Privacy Protection	[42]	Data Access Control, DBA, Auditable Data Provenance
5.	Data Protection	[49,79]	Blockchain Digital signature, Consensus Algorithm

5.2. Performance Effect

Although performance and scalability are not security issues, it is important to briefly explain their effect in achieving efficiently secured IoT-BC system. While blockchain presents a compelling solution for securing IoT, its adoption comes with performance-related challenges due to the limitations of IoT devices in terms of computation, memory, and power supply. Scalability is also a major hurdle. Integrating public blockchains can lead to slow transaction processing times and high costs, making them unsuitable for resource-constrained IoT devices. Additionally, the complexity increases, potentially requiring specialised knowledge or dedicated blockchain expertise for management. Also, the regulatory landscape surrounding blockchain is still evolving, creating uncertainty for businesses considering its implementation. This ambiguity could hinder wider adoption. This survey will serve as a guide for future research and will allow researchers to build upon previous work to further improve security solutions.

6. Conclusions

The IoT security ecosystem has witnessed a great transformation with the introduction of blockchain, and several research projects have explored this technology. This paper reviewed works from the blockchain platform perspective. This was achieved through a review of literature by identifying the gaps and positioning the contributions. This paper is more security-centric compared to the existing literature. It presents a detailed review of the emerging blockchain platforms used in IoT system security, with more emphasis on cryptography and other security components of the blockchain. The reviewed work

presented a greater adoption of private and hybrid blockchains compared to exclusively public blockchains, with 55%, 25%, and 20%, respectively, in the number of works. Technology such as IPFS, cloud technology, TSL, fog, etc. has been identified as promising for complementing security in IoT-Blockchain Integration. Also, there has been an increase in research work in this field in the last few years and we expect more to come.

Potential directions in this domain include the following: Improving the existing consensus mechanism to enhance more adoption of it in the resource-constrained IoT system without compromising security, such as using Proof of Location (PoL). Cryptography: More work should be done to further discover lightweight cryptography algorithms that will be quantum-proof. Innovative identity mechanisms, e.g., decentralised identity, can be adopted in place of full blockchain nodes for very light IoT devices.

Author Contributions: Conceptualization, Y.K., D.D. and E.G.; methodology, Y.K. and D.D.; validation, D.D. and E.G.; formal analysis, Y.K., D.D. and E.G.; investigation, Y.K., D.D. and E.G.; resources, Y.K., D.D. and E.G.; data curation, Y.K. and D.D.; writing—original draft preparation, Y.K. and D.D.; writing—review and editing, Y.K., D.D. and E.G.; visualization, Y.K. and D.D.; supervision, D.D. and E.G.; project administration, D.D.; funding acquisition, D.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Arab–German Young Academy (AGYA) Academy grant number 01DL20003 from BMBF. This research was funded by UWE Bristol: PHD Scholarship from Computer Science Research Centre, College of Arts, Technology and Environment, +CHIST–ERA EU Project grant number EP/Y036301/1 from EPSRC.

Data Availability Statement: No new data were created or analyzed in this study.

Acknowledgments: Sincere appreciation to the reviewers for their insightful comments.

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

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
BC	Blockchain
PoW	Proof of Work
PoS	Proof of Stake
PoL	Proof of Location
BFT	Byzantine Fault Tolerance
PBFT	Practical Byzantine Fault Tolerance
TPS	Throughput Per Second
DoS	Denial of Service
ECC	Elliptic Curve Cryptography

References

1. Khan, M.N.U.; Cao, W.; Tang, Z.; Ullah, A.; Pan, W. Energy-Efficient De-Duplication Mechanism for Healthcare Data Aggregation in IoT. *Future Internet* **2024**, *16*, 66 . [CrossRef]
2. Wang, H.; Huang, J.; Wang, G.; Lu, H.; Wang, W. Contactless Patient Care Using Hospital IoT: CCTV-Camera-Based Physiological Monitoring in ICU. *IEEE Internet Things J.* **2024**, *11*, 5781–5797. [CrossRef]
3. Bagheri, N.; Bendavid, Y.; Safkhani, M.; Rostampour, S. Smart Grid Security: A PUF-Based Authentication and Key Agreement Protocol. *Future Internet* **2024**, *16*, 9. [CrossRef]
4. Lasla, N.; Doudou, M.; Djenouri, D.; Ouadjaout, A.; Zizoua, C. Wireless energy efficient occupancy-monitoring system for smart buildings. *Pervasive Mob. Comput.* **2019**, *59*, 101037. [CrossRef]
5. Marin, O.; Cioara, T.; Anghel, I. Blockchain Solution for Buildings' Multi-Energy Flexibility Trading Using Multi-Token Standards. *Future Internet* **2023**, *15*, 177. [CrossRef]
6. Djenouri, D.; Laidi, R.; Djenouri, Y.; Balasingham, I. Machine Learning for Smart Building Applications: Review and Taxonomy. *ACM Comput. Surv.* **2019**, *52*, 1–36. [CrossRef]
7. Djenouri, Y.; Belhadi, A.; Djenouri, D.; Srivastava, G.; Lin, J.C.W. A Secure Intelligent System for Internet of Vehicles: Case Study on Traffic Forecasting. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 13218–13227. [CrossRef]

8. Jovanovic, B. Internet of Things Statistics for 2023—Taking Things Apart. Technical Report; DataProt. 2023. Available online: https://dataprot.net/statistics/iot-statistics/ (accessed on 1 May 2024).
9. Rasheed, A. IoT Explosion Connected Possibility. 2016 . Available online: https://www.linkedin.com/pulse/iot-explosion-connected-possibility-abdul-rasheed (accessed on 1 May 2024).
10. Abou-Nassar, E.M.; Iliyasu, A.M.; El-Kafrawy, P.M.; Song, O.Y.; Bashir, A.K.; El-Latif, A.A. DITrust Chain: Towards Blockchain-Based Trust Models for Sustainable Healthcare IoT Systems. *IEEE Access* **2020**, *8*, 111223–111238. [CrossRef]
11. Abdmeziem, M.; Tandjaoui, D.; Romdhani, I. Architecting the Internet of Things: State of the Art. In *Robots and Sensor Clouds*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 55–75. [CrossRef]
12. Antão, L.; Pinto, R.; Reis, J.P.; Gonçalves, G. Requirements for Testing and Validating the Industrial Internet of Things. In Proceedings of the 2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Västerås, Sweden, 9–13 April 2018. [CrossRef]
13. Harit, A.; Ezzati, A.; Elharti, R. Internet of things security: Challenges and perspectives. In Proceedings of the ACM International Conference Proceeding Series, Association for Computing Machinery, Tacoma, WA, USA, 18–20 August 2017. [CrossRef]
14. Cyrus, C. IoT Cyberattacks Escalate in 2021, According to Kaspersky. 2021 . Available online: https://www.iotworldtoday.com/security/iot-cyberattacks-escalate-in-2021-according-to-kaspersky (accessed on 1 May 2024).
15. Mamdouh, M.; Elrukhsi, M.A.; Khattab, A. Securing the Internet of Things and Wireless Sensor Networks via Machine Learning: A Survey. In Proceedings of the 2018 International Conference on Computer and Applications, ICCA 2018, Beirut, Lebanon, 25–26 August 2018; pp. 215–218. [CrossRef]
16. Nathan, C.; Leandros, M.A.; Ioanna, K.; Nestoras, C.; Amine, F.M. Blockchain Technology: Security and Privacy Issues. In *Blockchain Technology and Innovations in Business Processes*; Srikanta, P., Wang, T.S., Tao, S., Kumar, P.S., Eds.; Springer: Singapore, 2021; pp. 95–107. [CrossRef]
17. Zhong, S.; Huang, X. Special Focus on Security and Privacy in Blockchain-Based Applications. *Sci. China Inf. Sci.* **2020**, *63*, 130100. [CrossRef]
18. Zhang, R.; Xue, R.; Liu, L. Security and privacy on blockchain. *ACM Comput. Surv.* **2019**, *52, 1–34.* . [CrossRef]
19. Huang, H.; Kong, W.; Zhou, S.; Zheng, Z.; Guo, S. A Survey of State-of-the-Art on Blockchains. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 44. [CrossRef]
20. Chen, Z.; Cui, L.; Palanisamy, B.; Zhang, L.J. (Eds.) Blockchain—ICBC 2020. In *Lecture Notes in Computer Science*; Springer International Publishing: Cham, Switzerland, 2020; Volume 12404. [CrossRef]
21. Shaikh, M.; Shibu, C.; Angeles, E.; Pavithran, D. Data storage in blockchain based architectures for internet of things (IoT). In Proceedings of the 2021 IEEE International IOT, Electronics and Mechatronics Conference, IEMTRONICS 2021, Toronto, ON, Canada, 21–24 April 2021. [CrossRef]
22. Amitha, A.K.; Pamba, R.V. A Solid Waste Management System Using Smart Bins in a Decentralized Manner in Ethereum Blockchain Network for Incentivization. In Proceedings of the ICCSC 2023—Proceedings of the 2nd International Conference on Computational Systems and Communication, Thiruvananthapuram, India, 3–4 March 2023. [CrossRef]
23. Shah, R. A Systematic Review on Blockchain in IoT. In Proceedings of the 4th International Conference on Energy, Power, and Environment, ICEPE 2022, Shillong, India, 29 April–1 May 2022. [CrossRef]
24. Wang, Q.; Zhu, X.; Ni, Y.; Gu, L.; Zhu, H. Blockchain for the IoT and industrial IoT: A review. *Internet Things* **2019**, *10*, 100081. [CrossRef]
25. Abdelmaboud, A.; Ahmed, A.I.A.; Abaker, M.; Eisa, T.A.E.; Albasheer, H.; Ghorashi, S.A.; Karim, F.K. Blockchain for IoT Applications: Taxonomy, Platforms, Recent Advances, Challenges and Future Research Directions. *Electronics* **2022**, *11*, 630. [CrossRef]
26. Lao, L.; Li, Z.; Hou, S.; Xiao, B.; Guo, S.; Yang, Y. A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 18. [CrossRef]
27. Chowdhury, M.J.M.; Ferdous, M.S.; Biswas, K.; Chowdhury, N.; Muthukkumarasamy, V. A survey on blockchain-based platforms for IoT use-cases. *Knowl. Eng. Rev.* **2020**, *35*, e19. [CrossRef]
28. Sadawi, A.A.; Hassan, M.S.; Ndiaye, M. A Review on the Integration of Blockchain and IoT. In Proceedings of the ICCSPA 2020—4th International Conference on Communications, Signal Processing, and Their Applications, Online, 16–18 March 2021. [CrossRef]
29. Sultan, A.; Mushtaq, M.A.; Abubakar, M. IoT security issues via blockchain: A review paper. In Proceedings of the ACM International Conference Proceeding Series, Wuhan, China, 12–13 July 2019; Volume Part F148153, pp. 60–65. [CrossRef]
30. Kumar, S.; Vidhate, A. Issues and Future Trends in IoT Security using Blockchain: A Review. In Proceedings of the IDCIoT 2023-International Conference on Intelligent Data Communication Technologies and Internet of Things, Bengaluru, India, 5–7 January 2023; pp. 976–984. [CrossRef]
31. Sallam, A.; Qahtani, F.A.; Gaid, A.S. Blockchain in Internet of Things: A Systematic Literature Review. In Proceedings of the 2021 International Conference of Technology, Science and Administration, ICTSA 2021; 22–24 March 2021. [CrossRef]
32. Alam, S.R.; Jain, S.; Doriya, R. Security threats and solutions to IoT using Blockchain: A Review. In Proceedings of the Proceedings—5th International Conference on Intelligent Computing and Control Systems, ICICCS 2021, Madurai, India, 6–8 May 2021; pp. 268–273. [CrossRef]

33. Shammar, E.A.; Zahary, A.T.; Al-Shargabi, A.A. A Survey of IoT and Blockchain Integration: Security Perspective. *IEEE Access* **2021**, *9*, 156114–156150. [CrossRef]
34. Darla, S.; Naveena, C. Survey on Securing Internet of Things through Block chain Technology. In Proceedings of the International Conference on Electronics and Renewable Systems, ICEARS 2022, Tuticorin, India, 16–18 March 2022; pp. 836–844. [CrossRef]
35. Stefanescu, D.; Montalvillo, L.; Galan-Garcia, P.; Unzilla, J.; Urbieta, A. A Systematic Literature Review of Lightweight Blockchain for IoT. *IEEE Access* **2022**, *10*, 123138–123159. [CrossRef]
36. Dotan, M.; Pignolet, Y.A.; Schmid, S.; Tochner, S.; Zohar, A. Survey on Blockchain Networking: Context, State-of-the-Art, Challenges. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 107. [CrossRef]
37. Sadawi, A.A.; Hassan, M.S.; Ndiaye, M. A Survey on the Integration of Blockchain with IoT to Enhance Performance and Eliminate Challenges. *IEEE Access* **2021**, *9*, 54478–54497. [CrossRef]
38. Ayub Khan, A.; Laghari, A.A.; Shaikh, Z.A.; Dacko-Pikiewicz, Z.; Kot, S. Internet of Things (IoT) Security with Blockchain Technology: A State-of-the-Art Review. *IEEE Access* **2022**, *10*, 122679–122695. [CrossRef]
39. Jabbar, R.; Kharbeche, M.; Al-Khalifa, K.; Krichen, M.; Barkaoui, K. Blockchain for the Internet of Vehicles: A Decentralized IoT Solution for Vehicles Communication Using Ethereum. *Sensors* **2020**, *20*, 3928. [CrossRef] [PubMed]
40. Hussein, D.H.; Anbarasu, R.; Matrawy, A.; Ibnkahla, M. Towards a Decentralized Access Control System for IoT Platforms based on Blockchain Technology. In Proceedings of the 2020 International Symposium on Networks, Computers and Communications (ISNCC), Montreal, QC, Canada, 20–22 October 2020; pp. 1–6. [CrossRef]
41. Othman, A.A.H.; Muhammed, E.A.A.; Mujahid, H.K.M.; Muhammed, H.A.A.; Mosleh, M.A.A. Online Voting System Based on IoT and Ethereum Blockchain. In Proceedings of the 2021 International Conference of Technology, Science and Administration (ICTSA), Taiz, Yemen, 22–24 March 2021; pp. 1–6. [CrossRef]
42. Bawankar Chetan, D.; Kukkar Sayali, D.; Chaudhari Vaishnavi, J.; Rashinkar Suraj, R.; Solanke Sanket, U. Preventing Data Counterfeiting Among Internet of Things (IoT) Devices using Ethereum Blockchain. In Proceedings of the 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 18–19 June 2021; pp. 484–488. [CrossRef]
43. Patil, S.; Adsul, D.; Desale, S.; Gandole, K. Smart Vehicle Rental Application using Blockchain and IoT. In Proceedings of the 2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Bangalore, India, 23–25 December 2022; pp. 1–6. [CrossRef]
44. Sureshkumar, T.; Sivaraj, R.; Vijayakumar, M. Design and implementation of a framework for blockchain based security using IoT. *J. Intell. Fuzzy Syst.* **2023**, *44*, 905–918. [CrossRef]
45. Gupta, S.; Chithaluru, P.; El Barachi, M.; Kumar, M. Secure data access using blockchain technology through IoT cloud and fabric environment. *Secur. Priv.* **2024**, *7*, e356. [CrossRef]
46. Raj, A.; Maji, K.; Shetty, S.D. Ethereum for Internet of Things security. *Multimed. Tools Appl.* **2021**, *80*, 18901–18915. [CrossRef]
47. Jeyakkannan, N.; Subathra, G.; Karthika, R.; Lavanya, A.; Kirupanithi, N. Simulation of distributed denial of service attack against ethereum smart contract on the blockchain. In Proceedings of the International Conference on Computer Vision and Internet of Things 2023 (ICCVIoT'23), Coimbatore, India, 7–8 December 2023. [CrossRef]
48. Al-Joboury, I.M.; Al-Hemiary, E.H. Automated Decentralized IoT Based Blockchain Using Ethereum Smart Contract for Healthcare. In *Enhanced Telemedicine and e-Health: Advanced IoT Enabled Soft Computing Framework*; Springer: Berlin/Heidelberg, Germany, 2021 ; pp. 179–198. [CrossRef]
49. Narayanan, N.C.; Withana, C.; Elchouemi, A.; Li, G. Leveraging Blockchain Technology for a secure IoT data sharing. In Proceedings of the 2023 International Conference on Intelligent Education and Intelligent Research, IEIR 2023, Wuhan, China, 5–7 November 2023. [CrossRef]
50. Pajooh, H.H.; Rashid, M.; Alam, F.; Demidenko, S. Hyperledger fabric blockchain for securing the edge internet of things. *Sensors* **2021**, *21*, 359. [CrossRef]
51. Lu, Y.; Liu, Z.; Wang, S.; Li, Z.; Liu, W.; Chen, X. Temporal Index Scheme of Hyperledger Fabric System in IoT. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 9945530. [CrossRef]
52. Alshehri, S.; Bamasag, O. AAC-IoT: Attribute Access Control Scheme for IoT Using Lightweight Cryptography and Hyperledger Fabric Blockchain. *Appl. Sci.* **2022**, *12*, 8111. [CrossRef]
53. Bettayeb, M.; Nasir, Q.; Talib, M.A. Hyperledger-Based secure Firmware update delivery for IoT devices. In Proceedings of the ArabWIC 2021: The 7th Annual International Conference on Arab Women in Computing in Conjunction with the 2nd Forum of Women in Research, Sharjah, United Arab Emirates, 25–26 August 2021. [CrossRef]
54. Lee, S.; Kim, M.; Lee, J.; Hsu, R.H.; Kim, M.S.; Quek, T.Q.S. Facing to Latency of Hyperledger Fabric for Blockchain-enabled IoT: Modeling and Analysis. *arXiv* **2021**, arXiv:2102.09166.
55. Guo, F.; Xiao, X.; Hecker, A.; Dustdar, S. Modeling Ledger Dynamics in IOTA Blockchain. In Proceedings of the 2022 IEEE Global Communications Conference, GLOBECOM 2022, Rio de Janeiro, Brazil, 4–8 December 2022; pp. 2650–2655. [CrossRef]
56. Silvano, W.F.; Marcelino, R. Iota Tangle: A cryptocurrency to communicate Internet-of-Things data. *Future Gener. Comput. Syst.* **2020**, *112*, 307–319. [CrossRef]
57. Alshaikhli, M.; Elfouly, T.; Elharrouss, O.; Mohamed, A.; Ottakath, N. Evolution of Internet of Things from Blockchain to IOTA: A Survey. *IEEE Access* **2022**, *10*, 844–866. [CrossRef]
58. Divya, M.; Nagaveni B. IOTA-Next Generation Block chain. *Int. J. Eng. Comput. Sci.* **2018**, *7*, 23823–23826. [CrossRef]

59. IoTeX Team. IoTeX: A Decentralized Network for Internet of Things Powered by a Privacy-Centric Blockchain . 2018. Available online: https://www.google.com.hk/url?sa=t&source=web&rct=j&opi=89978449&url=https://s3.amazonaws.com/web-iotex-static/home/IoTeX_Whitepaper_1.5_EN.pdf&ved=2ahUKEwi8qffM6uSHAxVpsFYBHRKnNN8QFnoECBMQAQ&usg=AOvVaw2wmyly1o8p8XY1HbMqHlN3 (accessed on 1 May 2024).
60. Partida, A.; Criado, R.; Romance, M. Visibility graph analysis of IOTA and IoTeX price series: An intentional risk-based strategy to use 5G for IoT. *Electronics* **2021**, *10*, 2282. [CrossRef]
61. Fan, X.; Zhong, Z.; Chai, Q.; Guo, D. Ucam: A User-Centric, Blockchain-Based and End-to-End Secure Home IP Camera System. In *International Conference on Security and Privacy in Communication Systems*; Springer: Cham, Switzerland, 2020.
62. Ravichandran, V. *Wear-IoTex: Wearable e-Textile Glove Kit for In-Home Parkinson's Motor Assessment*; University of Rhode Island: Kingston, RI, USA, 2021.
63. Xu, X.; Wang, X.; Li, Z.; Yu, H.; Sun, G.; Maharjan, S.; Zhang, Y. Mitigating Conflicting Transactions in Hyperledger Fabric-Permissioned Blockchain for Delay-Sensitive IoT Applications. *IEEE Internet Things J.* **2021**, *8*, 10596–10607. [CrossRef]
64. Na, D.; Park, S. IoT-Chain and Monitoring-Chain Using Multilevel Blockchain for IoT Security. *Sensors* **2022**, *22*, 8271. [CrossRef] [PubMed]
65. Wadhwa, D.; Gupta, D.; Saini, S.; Bathla, R. Blockchain for IoT Security and Privacy. In Proceedings of the 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 3–4 September 2021 ; pp. 1–5. [CrossRef]
66. Montanaro, T.; Sergi, I.; Corvaglia, S.; Mainetti, L.; Vilei, A.; Rossi, B.; Palmieri, A.; Patrono, L. Blockchain technology based on algorand applied to low-power and low-cost IoT devices. In Proceedings of the 2021 6th International Conference on Smart and Sustainable Technologies, SpliTech 2021, Bol and Split, Croatia, 8–11 September 2021. [CrossRef]
67. Varavallo, G.; Caragnano, G.; Bertone, F.; Vernetti-Prot, L.; Terzo, O. Traceability Platform Based on Green Blockchain: An Application Case Study in Dairy Supply Chain. *Sustainability* **2022**, *14*, 3321. [CrossRef]
68. Cardamone, N.; Dalena, V.; Mauro, A.; Settembre, M.; Vecchia, G.; Vitaliti, A.; Dondossola, G.; Bartalesi, D.; Garrone, F.; Terruggia, R. Blockchain-Based Public Key Authentication of IoT Devices for Electrical Energy Systems. In Proceedings of the 2022 AEIT International Annual Conference (AEIT), Rome, Italy, 3–5 October 2022; pp. 1–6. [CrossRef]
69. Ismail, S.; Reza, H.; Zadeh, H.K.; Vasefi, F. A Blockchain-based IoT Security Solution Using Multichain. In Proceedings of the 2023 IEEE 13th Annual Computing and Communication Workshop and Conference, CCWC 2023, Virtual Conference, 8–11 March 2023; pp. 1105–1111. [CrossRef]
70. Sawant, A.; Prabhu, N.; Nagpure, S. Securing IoT Using MultiChain. In Proceedings of the 2nd International Conference on Advances in Science & Technology (ICAST), Mumbai, India, 8–9 April 2019.
71. Umran, S.M.; Lu, S.; Abduljabbar, Z.A.; Nyangaresi, V.O. Multi-chain blockchain based secure data-sharing framework for industrial IoTs smart devices in petroleum industry. *Internet Things* **2023**, *24*, 100969. [CrossRef]
72. Chang, J.; Ni, J.; Xiao, J.; Dai, X.; Jin, H. SynergyChain: A Multichain-Based Data-Sharing Framework with Hierarchical Access Control. *IEEE Internet Things J.* **2022**, *9*, 14767–14778. [CrossRef]
73. Shehzad, F.; Javaid, N.; Farooq, U.; Tariq, H.; Ahmad, I.; Jabeen, S. IoT Enabled E-Business via Blockchain Technology Using Ethereum Platform. In *Web, Artificial Intelligence and Network Applications*; Springer: Cham, Switzerland, 2020; pp. 671–683. [CrossRef]
74. Frikha, T.; Chaabane, F.; Aouinti, N.; Cheikhrouhou, O.; Ben Amor, N.; Kerrouche, A. Implementation of Blockchain Consensus Algorithm on Embedded Architecture. *Secur. Commun. Netw.* **2021**, *2021*, 9918697. [CrossRef]
75. Mohanta, B.K.; Sahoo, A.; Pate, S.; Panda, S.S.; Jena, D.; Gountia, D. DecAuth: Decentralized Authentication Scheme for IoT Device Using Ethereum Blockchain. In Proceedings of the IEEE Region 10 Conference (TENCON 2019), Kochi, India, 17–20 October 2019; pp. 558–563.
76. Urien, P. Demonstrating Trusted Blockchain IoT Device Based on TLS-PSK Secure Element. In Proceedings of the 5th Conference on Blockchain Research and Applications for Innovative Networks and Services, BRAINS 2023, Paris, France, 11–13 October 2023. [CrossRef]
77. Zhao, L.; Vigneri, L.; Cullen, A.; Sanders, W.; Ferraro, P.; Shorten, R. Secure Access Control for DAG-Based Distributed Ledgers. *IEEE Internet Things J.* **2022**, *9*, 10792–10806. [CrossRef]
78. Lin, B.Y.; Dziubałtowska, D.; Macek, P.; Penzkofer, A.; Müller, S. TangleSim: An Agent-based, Modular Simulator for DAG-based Distributed Ledger Technologies. *arXiv* **2023**, arXiv:2305.01232.
79. Fan, X.; Zhong, Z.; Guo, D.; Chai, Q.; Romano, S. Connecting Smart Devices to Smart Contracts with W3bstream. In Proceedings of the 2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Dubai, United Arab Emirates, 1–5 May 2023; pp. 1–2. [CrossRef]
80. Alkhodair, A.; Mohanty, S.; Kougianos, E.; Puthal, D. McPoRA: A multi-chain proof of rapid authentication for post-blockchain based security in large scale complex cyber-physical systems. In Proceedings of the IEEE Computer Society Annual Symposium on VLSI, ISVLSI, Limassol, Cyprus, 6–8 July 2020; pp. 446–451. [CrossRef]

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

future internet

MDPI

Article

Permissionless Blockchain Recent Trends, Privacy Concerns, Potential Solutions and Secure Development Lifecycle †

Talgar Bayan [1], Adnan Yazici [1] and Richard Banach [2,*]

[1] Department of Computer Science, Nazarbayev University, Qabanbay Batyr Ave 53, Astana 010000, Kazakhstan; talgar.bayan@nu.edu.kz (T.B.); adnan.yazici@nu.edu.kz (A.Y.)
[2] Department of Computer Science, University of Manchester, Oxford Rd, Manchester M13 9PL, UK
* Correspondence: richard.banach@manchester.ac.uk
† This paper is an extended version of our paper published in the IEEE Smart Information Systems and Technologies (SIST), 4–6 May 2023, Astana, Kazakhstan.

Abstract: Permissionless blockchains have evolved beyond cryptocurrency into foundations for Web3 applications, decentralized finance (DeFi), and digital asset ownership, yet this rapid expansion has intensified privacy vulnerabilities. This study provides a comprehensive review of recent trends, emerging privacy threats, and mitigation strategies in permissionless blockchain ecosystems. We examine six developments reshaping the landscape: meme coin proliferation on high-throughput networks, real-world asset tokenization linking on-chain activity to regulated identities, perpetual derivatives exposing trading strategies, institutional adoption concentrating holdings under regulatory oversight, prediction markets creating permanent records of beliefs, and blockchain–AI integration enabling both privacy-preserving analytics and advanced deanonymization. Through this work and forensic analysis of documented incidents, we analyze seven critical privacy threats grounded in verifiable 2024–2025 transaction data: dust attacks, private key management failures, transaction linking, remote procedure call exposure, maximal extractable value extraction, signature hijacking, and smart contract vulnerabilities. Blockchain exploits reached $2.36 billion in 2024 and $2.47 billion in the first half of 2025, with over 80% attributed to compromised private keys and signature vulnerabilities. We evaluate privacy-enhancing technologies, including zero-knowledge proofs, ring signatures, and stealth addresses, identifying the gap between academic proposals and production deployment. We further propose a Secure Development Lifecycle framework incorporating measurable security controls validated against incident data. This work bridges the disconnect between privacy research and industrial practice by synthesizing current trends, providing insights, documenting real-world threats with forensic evidence, and providing actionable insights for both researchers advancing privacy-preserving techniques and developers building secure blockchain applications.

Keywords: blockchain privacy; smart contract security; privacy vulnerabilities; secure development lifecycle; Web3; DeFi security; permissionless blockchain; privacy threats

1. Introduction

The ecosystem serves an estimated 30–60 million monthly active users, though precise measurement remains difficult because individuals often control multiple addresses [1]. While illicit activity declined to 0.4% of transaction volume in 2024 from 0.9% in 2023 [2,3], this growth trajectory has brought privacy vulnerabilities into sharper focus.

A permissionless blockchain operates as a decentralised distributed ledger where any participant can read data, submit transactions, and participate in consensus without authorisation from a central authority. Prominent examples include Bitcoin, Ethereum, and stablecoin networks such as Tether (USDT), which enhance financial access for populations underserved by traditional banking. The transparency that enables public verification creates a fundamental tension: transaction histories, wallet balances, and asset flows remain permanently visible on public ledgers [4,5]. This openness enables accountability but exposes users to deanonymization, targeted attacks, and behavioural profiling [6].

Developer activity serves as a leading indicator of ecosystem health [7]. Figure 1 shows monthly active developers across blockchain networks from 2009 to 2022, reaching approximately 22,000 by 2023. Updated data from 2024 indicates continued growth to 33,000 monthly active developers working across 1.7 million repositories [8]. Developer activity serves as a leading indicator of ecosystem health [7].



Figure 1. Monthly total number of active developers across all blockchain networks from 2009 to 2022. Figure reproduced from the Electric Capital Developer Report 2022 [7], based on monthly samples from January 2010 to December 2022. Accessed 1 February 2023. The data are used here for comparison with the most recent active developer trends.

Figure 2 shows monthly active developers across blockchain networks from 2015 to 2025, distinguishing between single-chain and multi-chain developers.

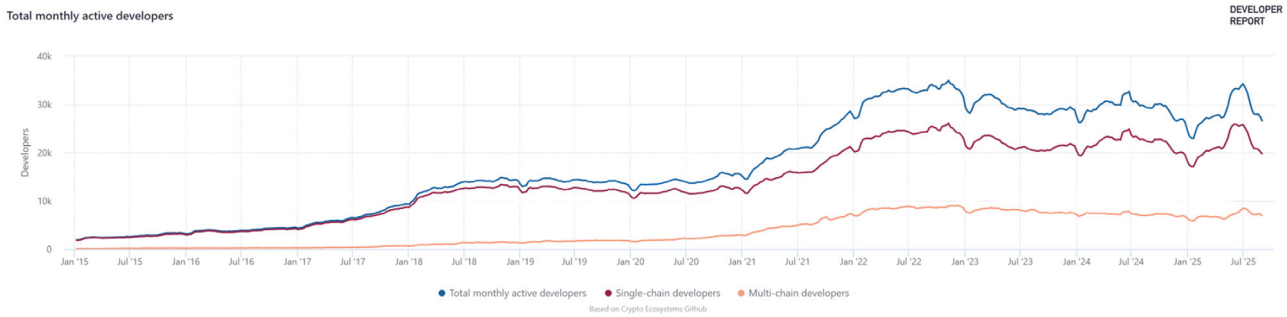


Figure 2. Monthly active blockchain developers from 2015 to 2025, segmented by development focus. The blue line represents total monthly active developers across all blockchain networks. The red line shows single-chain developers working exclusively on one blockchain ecosystem. The orange line indicates multi-chain developers contributing to multiple blockchain projects. Data demonstrates peak developer activity of approximately 35,000 in early 2023, followed by a decline to 25,000 by mid-2025, reflecting market consolidation following the 2022 cryptocurrency market downturn. The persistent gap between single-chain and multi-chain developers highlights ecosystem fragmentation. Source: Electric Capital Developer Report 2024 [8], based on analysis of 1.7 million GitHub repositories from the Crypto Ecosystems database [9]. Data accessed 11 September 2025.

The data reveals developer growth from approximately 2000 in 2015 to peak activity of 35,000 in early 2023, followed by consolidation to 25,000 by mid-2025 [8]. The divergence between single-chain and multi-chain developers after 2020 reflects increasing blockchain ecosystem fragmentation and cross-chain development patterns.

Table 1 presents the leading permissionless blockchains by total value locked (TVL), daily transaction volume, and active developer count as of September 2025. All quantitative metrics follow definitions provided in Appendix B.

Table 1. Popular Permissionless Blockchains.

Blockchain	Type	TVL	Transactions (Daily)	Protocols	Monthly Active Developers
Ethereum	Layer 1	94.4 B [10]	1.6 M [11]	1250 [10]	6244 [8]
BSC	Layer 1	7.58 B [10]	13.56 M [12]	882 [10]	150+ [8]
Base	Layer 2	5.06 B [10]	126 M [10]	616 [10]	1500+ [8]
Solana	Layer 1	12.77 B [10]	290 M [13]	299 [10]	6000 [8]
Polygon	Layer 2	1.22 B [10]	3.1 M [14]	604 [10]	1800+ [8]
Tron	Layer 1	6.42 B [10]	18 M [15]	37 [10]	500+ [8]
BTC	Layer 1	8.54 B [10]	535 K [16]	42 [10]	1200 [8]
Near	Layer 1	97 M [10]	4.4 M [17]	32 [10]	1000+ [8]

Despite substantial academic work on privacy-preserving techniques such as zero-knowledge proofs [18,19], ring signatures [20], and stealth addresses [21], a gap persists between theoretical solutions and production deployment. Recent security incidents demonstrate this disconnect: losses from blockchain exploits reached $2.36 billion in 2024 and $2.47 billion in the first half of 2025 [22], with over 80% attributed to compromised private keys and signature vulnerabilities rather than protocol flaws. Emerging trends, including Layer-2 scaling solutions, real-world asset (RWA) tokenisation, and institutional adoption, introduce new attack surfaces that existing privacy frameworks do not adequately address.

This paper provides a comprehensive analysis of privacy challenges in contemporary permissionless blockchains by examining emerging trends, documenting current threats through forensic analysis, and evaluating existing mitigation approaches. We extend our prior conference work [23] with three contributions:

- We analyse six emerging trends reshaping the blockchain privacy landscape: meme coin proliferation, real-world asset tokenisation, perpetual derivatives, institutional adoption, prediction markets, and blockchain-AI integration. We document how each trend introduces novel attack surfaces and privacy risks not addressed by existing frameworks.
- We systematically examine seven privacy threat categories through forensic analysis of documented 2024–2025 incidents: dust attacks, private key management failures, transaction linking, remote procedure call exposure, maximal extractable value extraction, signature hijacking, and smart contract vulnerabilities. Our analysis includes verifiable transaction data and incident timelines demonstrating attack patterns.
- We evaluate privacy-enhancing technologies—zero-knowledge proofs, zero-knowledge Ethereum Virtual Machines (zkEVMs), ring signatures, stealth addresses, and permissioned frameworks—examining their trade-offs in performance, developer accessibility, and deployment barriers. We propose a Secure Development Lifecycle framework incorporating lessons from documented incidents to bridge the gap between academic privacy research and industrial practice.

Section 2 reviews related work on blockchain privacy. Section 3 examines recent trends including meme coin proliferation, RWA tokenisation, perpetual derivatives exchanges, na-

tional cryptocurrency reserves, and prediction markets. Section 4 analyses specific privacy threats in permissionless networks. Section 5 evaluates potential solutions through privacy-improving cryptography and architectural approaches. Section 6 presents the SDLC framework with role assignments and measurable security controls. Section 7 discusses formal verification methods, and Section 8 addresses post-quantum cryptographic considerations. Section 10 concludes.

2. Related Works

This section summarises the related works on privacy preservation in permissionless blockchain networks and some alternative solutions for that. In [24], the Punishment Not Reward (PnR) blockchain architecture was proposed to address the issue of balancing privacy and openness in permissionless blockchain networks such as Ethereum and Bitcoin. The author suggested two methods, denial of service to the application and/or revocation of anonymity, as alternatives to the traditional reward-based approach for block creation to maintain the network's privacy. The EVM and Solidity have previously dominated development and transaction processing, but Solana has recently gained prominence, with the low network cost of the Solana transaction fee and the rise of meme coins.

These privacy concerns not only challenge user privacy and cause security issues, but also affect the performance of the blockchain, creating situations like backlogs, and leading to the fork of the blockchain and other issues. For example, dust attacks entice users to interact with a fake smart contract, and numerous dust attacks reduce the performance of the blockchain, creating problematic issues. In [25], a way was suggested to identify and prevent such things.

Saad et al. [26] identified multiple privacy attack vectors on the surface of the blockchain. A comprehensive investigation was done, which included aspects such as attacks associated with mathematical techniques used for creating the ledger, attacks associated with peer-to-peer architectures such as the 51% attack, and attacks associated with the application configuration of blockchain, such as wallet theft, double spending, etc.

In [27], a qualitative comparison was presented for privacy-preserving methods of engineering data. The methods include proxy encryption, homomorphic encryption, ZKP, and a trusted execution environment. The result indicates that approaches that rely on a trusted third party to preserve participant privacy do not provide sufficiently strong guarantees that sensitive data would not be exposed in modern data ecosystems. In [28], a privacy-preserving healthcare framework using Hyperledger Fabric was proposed, implementing zero-knowledge proofs through the Idemix suite to ensure patient data privacy with features such as anonymity and unlinkability. In [29], a systematic review of the current state-of-the-art in privacy-preserving research solutions and mechanisms on blockchain was presented, which also discussed challenges in blockchain scenarios such as post-quantum computing resistance, malicious-curious TTPs and regulatory challenges. Table 2 shows the results of the comparisons of the most closely related solutions for privacy concerns without permission. In one proposal, the present authors implemented a prototype of the PnR architecture on Hyperledger Fabric, a system with 4 nodes working on virtual machines as independent nodes. Subsequent work extended this architecture with NFT-based authentication mechanisms for decentralised autonomous organisation governance [30], demonstrating how accountability and privacy can coexist through selective anonymity revocation. In [4], some information on the privacy issues associated with blockchain is provided, and the privacy threats on blockchain are analysed, along with existing cryptographic defence mechanisms, such as anonymity and preservation of

transaction privacy. Furthermore, the authors summarised some typical implementations of privacy preservation mechanisms in blockchain.

Paper [5] listed several privacy issues in blockchain and the methods of protecting privacy in blockchain, such as a mixing service, ring signatures, confidential transactions, non-interactive zero-knowledge proofs, smart contract privacy, etc. The development of on-chain and off-chain ecosystems is growing rapidly, and some of the protection methods are not state-of-the-art. There are more threat models and more concerns now. In [6], Henry et al. demonstrated, foresight in identifying access privacy as a critical vulnerability in blockchain systems. The authors focused specifically on network-level privacy challenges, particularly the risks associated with transaction announcement and retrieval patterns that could compromise user anonymity when cryptographic privacy measures were in place. Their critique of relying solely on Tor for blockchain privacy protection proved prescient, as subsequent research has validated many of their concerns about the inadequacy of general-purpose anonymity networks for blockchain applications.

A comprehensive analysis of privacy challenges across permissionless and permissioned blockchains, including smart contract innovations and architectural approaches, was presented in [31]. Though the paper addresses only one facet of blockchain privacy, it establishes important groundwork for understanding how network-level adversaries could undermine the privacy guarantees that early blockchain adopters assumed were inherent to pseudonymous systems.

Table 2. Comparison of Related Solutions with our proposed PnR DAO using Hyperledger Fabric Developed.

Related Solutions	MS [1]	ZKP [2]	HLF [3]	RS [4]	SA [5]	PE [6]	zkEVM [7]
[32]		✓		✓		✓	
[5]		✓		✓		✓	
[24]		✓					
[29]		✓	✓	✓		✓	✓
[28,33]			✓				
[34]					✓		
[6,35,36]						✓	
[27]				✓			
[4]		✓	✓	✓		✓	✓
PnR DAO in Hyperledger Fabric Version	✓	✓	✓	✓	✓	✓	✓

[1] MS: Multi-Signature. [2] ZKP: Zero-Knowledge Proof. [3] HLF: Hyperledger Fabric. [4] RS: Ring Signature. [5] SA: Stealth Addresses. [6] PE: Privacy-improving. [7] zkEVM: Zero-Knowledge Ethereum Virtual Machine.

3. Emerging Trends in Permissionless Blockchain Ecosystems

Since Bitcoin's 2009 launch, the blockchain industry has evolved through the ICO boom (2017–2018), DeFi summer (2020), NFT markets (2021), and most recently Layer-2 scaling solutions and institutional adoption (2023–2025). We have witnessed the ICO bubble, the NFT boom, exchange mining, airdrop hunting, the DeFi summer, the Layer-2 wars, social chains, Play-to-Earn economies, Polymarket, nations accepting crypto as a reserve, and the rise of DePin networks. Each stage of adoption introduced new concepts, architectures, and attack surfaces. In the current phase, emerging trends such as Solana-powered meme coins, real-world asset (RWA) tokenisation, perpetual exchanges, and decentralised trading platforms are reshaping blockchain dynamics. Meanwhile, the rapid integration of artificial intelligence is introducing both opportunities and novel vulnerabilities. This section explores these evolving trends and the corresponding security challenges they bring to permissionless blockchains.

3.1. Meme Coin Proliferation on Solana and Speculative Market Dynamics

Solana's high throughput (65,000+ transactions per second) and low costs (<$0.01 per transaction) enabled the proliferation of speculative tokens [37]. By October 2025, Solana-based meme coins reached $11.4 billion market capitalisation, creating conditions where token creation requires minimal capital or technical knowledge, while platforms like Pump.fun processed more than 250,000 new token launches weekly, generating monthly revenues exceeding $40 million [38].

The scale reveals privacy risks beyond financial losses. Dogwifhat achieved a peak market capitalisation of $3.5 billion before declining over 60% [39], while newer tokens such as Fartcoin demonstrated gains exceeding 1600% within weeks [40]. Transaction transparency enables behavioural profiling through address clustering techniques [41] that link seemingly independent wallets through common input heuristics. Temporal analysis of trading activity provides insights into user decision-making processes, while integration with social media platforms creates correlation opportunities when users inadvertently link digital identities to wallet addresses. The accessibility of the SPL token standard has led to documented patterns of rug pulls, honeypot tokens preventing sells, and coordinated pump-and-dump schemes across social platforms. The regulatory vacuum surrounding meme tokens eliminates traditional recourse mechanisms, leaving investors with limited options following fraud or market manipulation while malicious actors exploit blockchain pseudonymity.

3.2. Real-World Asset Tokenisation and Verification Gaps

The tokenised real-world assets expanded by 85% year on year to reach $15.2 billion by December 2024, excluding the stablecoins [42]. Including fiat-backed tokens, the total market approached $230 billion by mid-2025, with Tether's USDT and Circle's USDC representing 93.5% of this segment [43]. The tokenised treasury sector grew 539% between January and December 2024, reaching $5.6 billion in total value locked., with BlackRock's BUIDL fund capturing 44% market share [42,43]. Over 119 issuers actively tokenise assets spanning private credit, commodities, real estate, and government securities [42]. Figure 3 illustrates the exponential growth trajectory of tokenised real-world assets from July 2021 to October 2025.

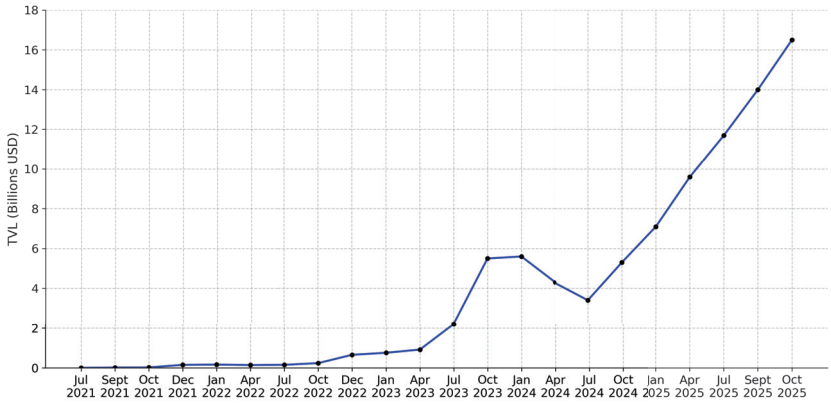


Figure 3. Total value locked in tokenised real-world asset protocols grew from $3 million (July 2021) to $16.72 billion (October 2025). Data obtained from DeFiLlama RWA Protocols Dashboard [44], based on monthly samples. Data accessed 11 September 2025.

Maintaining links between on-chain representations and off-chain assets remains a challenge. Unlike native cryptocurrency tokens whose value derives entirely from network consensus, tokenised assets depend on external verification mechanisms [45] to ensure

that the claimed backing exists. Chainlink's Proof of Reserve system uses oracle networks to verify collateral through automated queries to custodian banks [46], yet these systems remain vulnerable to custodian malfeasance, regulatory changes that affect ownership, and coordination failures between verification parties. Enterprise tokenisation platforms process sensitive information about asset ownership and transaction histories, consolidating data traditionally distributed across multiple intermediaries. The permissionless nature means that transaction patterns become permanently visible on public ledgers, enabling analysis of institutional investment strategies. The European MiCA regulation entered full effect in 2024 [43], establishing comprehensive rules for tokenised asset issuers within EU member states, while regulatory oversight in the United States remains fragmented between multiple agencies. Asset tokenisation requires KYC procedures for anti-money laundering compliance, linking on-chain addresses to verified real-world identities, and creating centralised databases that become targets for cybercriminals.

3.3. Perpetual Futures Decentralised Exchanges and Leverage Risks

Perpetual futures generated daily trading volumes exceeding $12 trillion during Q2 2025, representing 59% of cryptocurrency derivatives activity [47]. Hyperliquid launched in 2024 with a custom Layer-1 blockchain optimised for derivatives trading, achieving transaction finality under two seconds while processing peak daily volumes exceeding $8 billion [48]. By October 2024, Hyperliquid controlled over 80% of the decentralised perpetual derivatives market [47,49]. Chen et al. identified four distinct operational models that govern exchange operations [50], with empirical analysis revealing that decentralised exchanges that employ virtual automated market-making models exhibit price formation dynamics where open interest in long and short positions exerts opposite effects on price volatility [51]. Figure 4 shows growth from $20 million in July 2021 to $34 billion in October 2025.

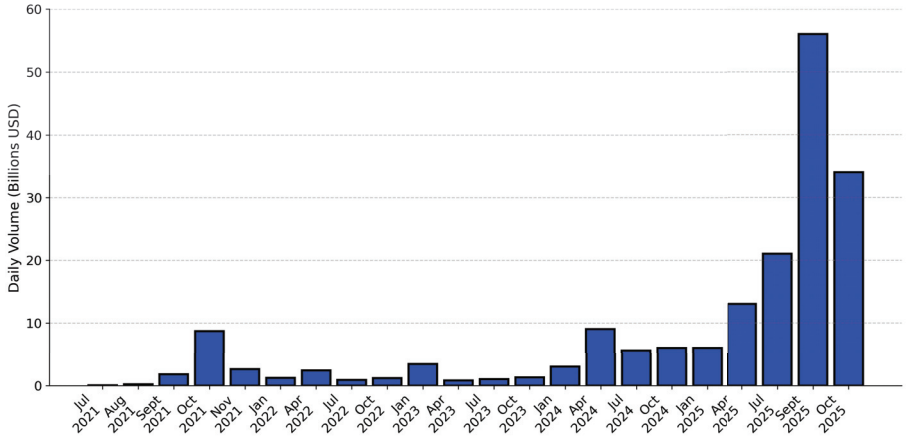


Figure 4. Perceptual Dex Trading sector experienced expansion from an initial $20 million to $34 billion range over the four years. Data obtained from the Perp Volume Dashboard on DeFiLlama [52], based on monthly samples from July 2021 to October 2025. Data accessed on 11 September 2025.

Ruan and Streltsov analysed high-frequency order book data from 2017 to 2023, documenting that spot market quality follows a U-shaped pattern over perpetual contracts' eight-hour funding cycles [53]. Their research identified that perpetual contracts increase spot trading volume while widening quoted spreads, reflecting increased informed trading during funding settlement periods. Market makers respond to heightened adverse selection risk by expanding bid-ask spreads. Platforms like Hyperliquid support leverage ratios up to 50× [48], while Aster advertises leverage exceeding 1000× [54]. The December 2024 Bitcoin

flash crash saw prices decline 7% from $103,853 to $92,251 within minutes, liquidating over $400 million in overleveraged long positions [47]. Transaction transparency enables competitors to observe large position accumulations and liquidation prices, facilitating front-running strategies. Blockchain analysis firms revealed extreme token concentration in Aster, with six wallet addresses controlling over 96% of total supply [54].

3.4. National Cryptocurrency Reserves and Institutional Adoption

Kazakhstan established the Alem Crypto Fund in September 2025, creating a state-backed vehicle for digital asset reserves through the Ministry of Artificial Intelligence and Digital Development [55]. The fund made its initial investment in BNB through a partnership with Binance Kazakhstan, managed by Qazaqstan Venture Group under the Astana International Financial Centre [56]. President Kassym-Jomart Tokayev announced plans for a comprehensive digital asset ecosystem with legislation targeted before 2026 [55], following the May 2025 plans for CryptoCity, a pilot zone allowing crypto payments [57]. After China's mining crackdown, Kazakhstan ranked second globally by Bitcoin hashrate in 2021, though regulatory concerns led to the closure of 36 unlicensed exchanges in 2024 [55].

The trend extends beyond Kazakhstan. El Salvador established an official Bitcoin reserve in 2021, while Bhutan began accumulating Bitcoin through state-backed mining operations as early as 2019 [58]. Brazil and Indonesia have explored frameworks for building national digital asset reserves [55]. The institutional adoption trajectory follows earlier initiatives where Cointelegraph reported in June 2025 that Kazakhstan's National Bank considered plans for a state-run crypto reserve funded with seized assets and state mining revenues [55]. Though the Alem Crypto Fund differs from a central bank reserve, it signals growing acceptance of digital assets as strategic holdings comparable to traditional foreign currency reserves. The institutional adoption trajectory raises privacy concerns for individual users. State-backed funds consolidating large positions enable correlation of on-chain activity with national policy decisions. The public nature of blockchain holdings allows tracking of government investment strategies, while regulatory frameworks tied to these initiatives often require enhanced know-your-customer procedures that link on-chain addresses to verified identities. This creates centralised databases connecting wallet addresses to real-world identities, undermining the pseudonymous properties that attracted early cryptocurrency adoption.

3.5. Blockchain Prediction Markets and Electoral Forecasting

Polymarket, a blockchain-based prediction market built on Polygon, processed over $3.3 billion in wagers on the 2024 U.S. presidential election between Donald Trump and Kamala Harris [59]. The platform predicted Trump's victory with 95% certainty before midnight on election night [60], contrasting with traditional polls that showed near 50-50 odds [61]. Polymarket's total value locked surged from $9.5 million in stablecoins at the start of 2024 to $220 million, with lifetime trading volume surpassing $1 billion by August [62]. The platform operates through users purchasing outcome shares using the USDC stablecoin, with prices adjusting in real-time based on trading activity.

Research firms Chaos Labs and Inca Digital identified evidence of wash trading on Polymarket, finding actual transaction volume around $1.75 billion compared with the platform's reported figure of $2.7 billion [62]. Four individuals collectively wagered $25 million on Trump's victory, with analysis suggesting strong reason to believe they represented the same entity [59,62]. The platform faced regulatory challenges, receiving a $1.4 million fine from the Commodity Futures Trading Commission in January 2022 for failure to register as a Swap Execution Facility [59]. Despite operating offshore to avoid U.S. regulations, an estimated $3.2 billion in cryptocurrency flowed through Polymarket

during the 2024 election, with VPN services suspected of enabling U.S.-based users to bypass geographic restrictions [60]. The FBI raided founder Shayne Coplan's residence in November 2024, with sources describing it as political retribution [60].

3.6. Blockchain-AI Integration and Federated Learning

The convergence of Blockchain and Artificial Intelligence (AI) represents a transformative step in addressing the core challenges of data privacy, trust, and decentralisation. Blockchain offers an immutable and transparent ledger capable of maintaining verifiable audit trails for AI-driven processes, which is especially relevant for distributed intelligence and data provenance verification. Recent works, such as Qu et al. [63], have comprehensively reviewed blockchain-enabled federated learning (FL) frameworks, highlighting their potential to overcome the limitations of centralised model aggregation by improving transparency, auditability, and incentive alignment. Similarly, Nguyen et al. [64] identified that integrating blockchain with edge-based FL allows for privacy-preserving computation at scale, supporting secure collaboration among heterogeneous IoT devices. This synergy mitigates the risks of single points of failure and strengthens accountability in AI model training. Furthermore, the emergence of advanced frameworks such as Biscotti [65] demonstrates how consensus-driven learning architectures can maintain both data confidentiality and model integrity without central coordination.

Beyond scalability and transparency, the blockchain–AI paradigm is rapidly evolving to address emerging threats such as poisoning and inference attacks. Ullah et al. [66] proposed a blockchain-based federated learning architecture (SPBFL-IoV) specifically tailored for Internet of Vehicles (IoV) systems to counter adversarial model manipulation through homomorphic encryption and robust update filtering. Bao et al. [67] extended this concept toward data governance by introducing blockchain-based privacy-preserving credit data sharing using attribute-based encryption, ensuring both fine-grained access control and traceable data exchanges. At a broader level, Liang and Ji [68] conducted a systematic review of privacy challenges in IoT-blockchain ecosystems, emphasising the trade-offs between transparency and confidentiality. Collectively, these developments underscore blockchain's pivotal role in enabling secure, decentralised AI infrastructures that are auditable, privacy-aware, and resilient against adversarial behaviour—marking a fundamental shift toward verifiable and accountable intelligence.

Our previous studies have explored these aspects in IoT-healthcare ecosystems. In [69], a smart e-health framework was developed to monitor elderly and disabled individuals using edge-based processing for real-time analysis while preserving privacy. Earlier, lightweight deep learning models were proposed for efficient healthcare data analysis [70], and colour-coded CNN architectures were used for activity recognition and anomaly detection in wearable sensor data [71]. In a related work, multi-tiered security and explainability were investigated in IoT–Edge–Cloud environments for anomaly detection and data management [72]. Furthermore, the integration of federated learning for healthcare data privacy and interoperability has already demonstrated promising results in protecting distributed health records [73]. Building on these foundations, our current proposal extends the same principles by integrating blockchain with federated learning to ensure both verifiable updates and privacy-compliant collaboration among distributed healthcare institutions. Each institution maintains its data locally while participating in a shared training process governed by blockchain-based consensus and smart contracts. This integration not only enhances data confidentiality but also introduces transparent audit trails for model contributions and parameter exchanges. The proposed system aims to create a trustworthy, regulation-aligned, and technically feasible framework that reflects ongoing

trends in secure, distributed healthcare data management. Beyond privacy applications, blockchain integration with search and retrieval systems has been explored in educational contexts [74], though these approaches primarily address content discovery rather than privacy preservation. Multimodal search capabilities in decentralised educational networks [75] demonstrate the potential for blockchain-based content distribution, though privacy considerations in such systems remain underexplored.

Having examined emerging trends that reshape the blockchain landscape, we now analyse specific privacy threats that exploit architectural transparency and user behaviour patterns.

4. Privacy Threats in Permissionless Blockchain Networks

Blockchain's transparent architecture comes with privacy risks. With the rapid growth of DeFi, NFT, and Web3 dApps, new risks have emerged. We shall discuss some of the latest typical risks in this paper.

4.1. Dust Transactions and Zero-Value Transfer Attacks

Address poisoning attacks represent an evolution of traditional dust attacks, exploiting user interface dependencies and cognitive biases in address verification processes. These attacks leverage the transparency of blockchain transaction logs, where malicious actors systematically monitor high-value wallet addresses through blockchain explorers to identify targets holding substantial assets, particularly stablecoins such as USDT on the Ethereum network [25]. The attack methodology involves generating vanity addresses that match the first 4-5 and the final 4-5 hexadecimal characters of legitimate addresses. This exploits users' tendency to verify only the beginning and end of lengthy hexadecimal strings rather than checking the complete address, a behavioural pattern rooted in cognitive shortcuts that affect security decision-making [76]. These attacks extend beyond simple address mimicry and include comprehensive transaction shadowing and asset impersonation. A documented August 2024 case involved an attacker creating 47 shadow transactions over six days before deceiving the victim into transferring 20 million USDT. (Verification: Transaction hash 0x08255ca0e42a872559437141fa46980e66d907f7668922467d67515b1ebb4b7f at block 20,571,432. Complete forensic timeline available at the referenced Etherscan link. Verification instructions: (1) Navigate to https://etherscan.io/tx/0x08255ca0e42a872559437141fa46980e66d907f7668922467d67515b1ebb4b7f (accessed on 6 October 2025), (2) Observe the 20M USDT transfer to the attacker's address, (3) Click "Click to see More" and examine Internal Transactions showing the token routing, (4) Review the victim's transaction history at block 20,571,300–20,571,450 to observe the pattern of shadow transactions preceding the attack. All blockchain data referenced is publicly accessible and immutable. This incident demonstrates the financial severity and technical sophistication of contemporary address poisoning schemes. The attack timeline reveals a methodical approach: initial reconnaissance (15–17 August), shadow transaction deployment (18–20 August), and execution phase (21 August), with the attacker maintaining consistent timing patterns matching the victim's typical transaction schedule).

These attacks have evolved from traditional dust attacks, which involve the mass distribution of minimal value tokens or nonfungible tokens to target addresses. The fundamental premise of dust attacks relies on the expectation that recipients will eventually interact with the received assets, either through attempted disposal or inadvertent smart contract engagement. Such interactions can trigger malicious contract code or provide additional information on user behaviour and wallet management practices.

Defensive measures against address poisoning attacks require comprehensive verification protocols that extend beyond superficial address inspection. Security-conscious users should implement complete address verification procedures, examining all 42 hexadecimal

characters rather than relying on prefix and suffix matching. The establishment of verified address whitelists provides an additional security layer, though this approach requires careful maintenance and verification procedures.

Transaction confirmation protocols should incorporate small-value test transactions before high-value transfers, particularly when transacting with previously unused addresses. This practice enables verification of recipient control and address accuracy while minimising potential losses from misdirected funds. Furthermore, users should maintain awareness of their complete transaction history and investigate unexpected zero-value transfers or unfamiliar token receipts, as these may indicate targeting by address poisoning campaigns.

The mathematical probability of an accidental collision with an address provides insight into the sophistication required for these attacks. For Ethereum addresses derived from the final 20 bytes of the Keccak-256 hash output, the probability of generating an address that matches the 5-character prefix and 5-character suffix patterns is approximately $\frac{1}{16^{10}} = \frac{1}{2^{40}}$, representing a computationally intensive but feasible undertaking for motivated attackers with sufficient resources.

4.2. Private Key Management Vulnerabilities

Private key management represents the foundational security challenge in permissionless blockchains, while elliptic curve cryptography provides theoretical security of approximately 2^{128} operations for key recovery [77], contemporary threats target key management practices rather than cryptographic strength.

Private key compromises dominated 2024 blockchain incidents, causing over 80% of the $2.36 billion in losses across 760 documented cases [22]. This trend accelerated in 2025, with first-half losses reaching $2.47 billion despite fewer total incident, state-sponsored groups like the Lazarus Group stole $1.34 billion across 47 incidents in 2024 [78]. These operations frequently employ social engineering methodologies, including fabricated employment interviews where malicious code embedded within technical assessments extracts private keys from developer environments. A particularly concerning development involves the ongoing exploitation of the LastPass security breach, where encrypted password vaults containing seed phrases and private keys continue to be compromised through cryptanalytic attacks. This incident has resulted in cascading losses exceeding $250 million as of 2024, with victims continuing to emerge as encrypted vaults are systematically cracked [79]. The temporal persistence of this threat vector demonstrates how single points of failure in centralised password management can create extended vulnerability windows.

Contemporary private key theft methodologies have expanded beyond direct extraction to include address poisoning attacks, where malicious actors generate vanity addresses with similar prefixes and suffixes to legitimate addresses, as mentioned above, exploiting user interface dependencies and muscle memory. These attacks resulted in more than $83 million in documented losses during the first half of 2025, with notable incidents including a $2.6 million misdirected transaction due to manipulation of the transaction history.

Enterprise environments face additional complexities in private key management, particularly regarding multi-signature implementations and threshold signature schemes. The security of multi-signature wallets depends critically on the independence of key storage locations and the threshold parameter t in an t-of-n scheme. Recent incidents, including the WEMIX platform compromise resulting in $6.1 million losses through stolen authentication keys, demonstrate how centralised key management practices can undermine the theoretical security of distributed signing schemes [80].

Hardware security modules and cold storage solutions [81] provide enhanced protection through physical isolation of private key material, yet remain vulnerable to supply

chain attacks and firmware compromise. The verification of hardware wallet integrity requires cryptographic attestation protocols, though the implementation of such systems remains inconsistent across consumer-grade devices.

The emergence of threshold signature schemes and secure multiparty computation protocols offers promising alternatives to traditional key management approaches. These cryptographic primitives enable distributed key generation and signing operations without any party possessing complete private key material, mathematically represented through secret sharing schemes where $k = \sum_{i=1}^{t} a_i \cdot s_i$ reconstructs the private key k only when t shares are combined. Prevention strategies must address both technical and human factors in private key security. The implementation of hardware-backed key storage, regular security audits of key management procedures, and comprehensive employee training regarding social engineering techniques have become essential components of contemporary blockchain security frameworks. The persistent evolution of attack methodologies necessitates continuous adaptation of defensive measures, particularly as the financial incentives for private key theft continue to increase with broader blockchain adoption and higher asset valuations.

4.3. Transaction Linking, Name Service Vulnerabilities, and On-Chain Analysis

Transaction linking exploits blockchain transparency to track user behaviour. Once users reveal wallet addresses publicly, graph analysis algorithms can connect their complete transaction history by clustering common inputs and identifying change addresses across the public ledger [82]. Transaction timing, gas fee patterns, and smart contract interactions create behavioural fingerprints that persist across address rotations. Cross-chain bridges compound this risk, as transaction timing and amounts can link addresses across multiple networks.

Blockchain name services like Ethereum Name Service introduce additional privacy concerns while improving usability. ENS domains function as NFT tokens that create permanent on-chain links between human-readable names and wallet addresses. Research by Xia et al. shows serious security problems within ENS, including domain squatting, malicious website connections, and scam addresses that exploit user trust in legitimate-looking domain names [83]. The convenience of ENS creates new attack opportunities through domain impersonation. Attackers register ENS domains that mimic legitimate wallet addresses, taking advantage of user interface flaws where address suggestions show ENS matches before exact address matches. This leads to phishing attacks where users accidentally select fraudulent domains when trying to access their own addresses. Name service registrations expose additional metadata through domain trading patterns and auction behaviours. The public ENS auction system reveals bidding strategies and value assessments that provide insights into users' financial capabilities. ENS subdomain delegation creates recursive privacy risks, where root domain owners gain visibility into all subdomain activities. Using name services across multiple decentralised applications creates expanded tracking surfaces. Consistent name usage enables behavioural profiling that spans multiple protocols and applications, building comprehensive user profiles.

Mathematical analysis shows the effectiveness of address clustering approaches in privacy reduction. Common input ownership techniques achieve over 90% accuracy in Bitcoin networks, while change address identification maintains a similar success rate through output value analysis and timing correlations. Defence strategies require careful privacy practices that address both technical and operational risks. Users should rotate addresses regularly and avoid reusing addresses across transactions. Privacy tools like CoinJoin implementations, ring signatures, and zero-knowledge proof systems provide better anonymity, though they need careful implementation to avoid creating new tracking opportunities. The usage of name services should follow privacy-conscious practices, in-

cluding careful ENS configurations and avoiding personally identifiable domain selections. Users must understand that name service registrations on the chain are permanent and have long-term privacy implications. Machine learning techniques in blockchain analysis have increased the sophistication of tracking attacks. Graph neural networks and deep learning algorithms can identify subtle patterns in transaction behaviours that traditional approaches miss, creating more challenging privacy preservation requirements [82].

Understanding these privacy challenges helps users make informed decisions about blockchain interactions and implement appropriate countermeasures to preserve financial privacy in permissionless networks.

4.4. RPC Infrastructure Privacy and MEV Extraction

Remote Procedure Call protocols form the main part of blockchain infrastructure, enabling communication between clients and blockchain nodes. Most users interact with blockchains through RPC endpoints without realising it, as popular wallet extensions like MetaMask use centralised RPC providers by default. Infura, owned by ConsenSys, serves as MetaMask's default RPC provider, handling millions of requests daily from users who cannot afford to run their own blockchain nodes [84]. Many third-party RPC nodes serve millions of users who seek to add chains to their wallets, improve transaction speed, or complete tasks for project owners. This widespread reliance on centralised RPC infrastructure creates serious privacy risks that most users remain unaware of.

When users send transactions through wallet extensions, their IP addresses, transaction patterns, and behavioural data become visible to RPC providers, typically stored for at least some days. These companies can build detailed profiles of user activity, including which decentralised applications they use, their transaction frequency, and their asset holdings. The centralised nature of this infrastructure contradicts the decentralised principles that blockchain networks aim to achieve. Running personal blockchain nodes requires technical expertise and financial resources that most users lack. Ethereum nodes need large storage space, processing power, and constant maintenance, forcing most users to depend on third-party RPC services and creating bottlenecks in what should be decentralised networks.

Over 43 major RPC providers currently serve the blockchain ecosystem, including Alchemy, QuickNode, Chainstack, and numerous smaller services [85]. RPC communication faces several security challenges beyond privacy concerns. Unencrypted connections between clients and servers allow man-in-the-middle attacks where malicious actors can intercept and modify transaction data. Poor authentication mechanisms enable unauthorised access to RPC interfaces, potentially allowing attackers to control blockchain nodes or extract sensitive information. The 2022 Tornado Cash incident demonstrated how centralised RPC providers can block access to specific smart contracts, showing the censorship risks inherent in centralised infrastructure [86]. Testing environments and development networks heavily depend on RPC infrastructure, with developers building decentralised applications typically using testnet RPC endpoints provided by companies like Alchemy and Infura. This dependency means that experimental blockchain development relies on centralised infrastructure, creating additional points of failure and surveillance.

Maximal Extractable Value bots represent a different but related threat to blockchain privacy and fairness. MEV bots are automated programs that monitor blockchain mempools for profitable opportunities, reordering transactions to extract value from regular users. These bots earn profits through frontrunning, sandwich attacks, and arbitrage strategies that exploit the transparent nature of blockchain transactions. Frontrunning occurs when MEV bots observe pending transactions and place their own transactions with higher gas fees to execute first, allowing bots to profit from price movements they know will occur

when the original transaction processes. Sandwich attacks involve MEV bots placing buy orders before a user's trade and sell orders after it, artificially manipulating prices to extract value from the user's transaction [87].

The scale of MEV extraction has grown substantially, with Flashbots data showing $7.4 million extracted through MEV on Ethereum in just 30 days. Most MEV activity targets popular decentralised exchanges like Uniswap, where bots can predict price impacts from large trades and position themselves accordingly [88]. The mathematical advantage comes from MEV bots having faster access to mempool data and the ability to adjust gas prices dynamically to ensure transaction ordering. Recent incidents demonstrate the complexity of MEV attacks, including a 2023 case where a rogue Ethereum validator conducted a $25 million sandwich attack by manipulating the block building process. The validator deposited the minimum 32 ETH required for validation, then used their block proposal privileges to frontrun MEV bots themselves, extracting value from both regular users and other MEV operators [89].

MEV bots create fairness problems that undermine the intended transparency and equality of blockchain networks. Regular users face higher transaction costs and worse execution prices because MEV bots consistently outbid them for favourable transaction ordering, creating a two-tier system where automated traders extract value from regular participants who lack the technical knowledge or resources to compete. MEV protection mechanisms exist, but remain underused. Private mempools and MEV protection services can protect transactions from front-running, but most wallet software does not implement these protections by default. Projects like Flashbots have developed tools to make MEV extraction more transparent and democratic, though adoption remains limited outside of trading operations. Understanding these infrastructure risks helps users make informed decisions about blockchain interaction methods. Running personal nodes provides the highest level of privacy and censorship resistance, though technical and financial barriers remain substantial. Alternative approaches include using privacy-focused RPC providers, rotating between different services, and implementing transaction privacy tools that reduce exposure to MEV extraction.

4.5. Permit and Permit2 Signature Hijacking Vulnerabilities

Off-chain token approval mechanisms like Permit2 reduce gas costs but create new attack surfaces. Phishing campaigns targeting these signatures caused $314 million in losses during H1 2024 from 260,000 victims, surpassing all 2023 losses [90]. This represented over half of DeFi security incidents in 2024 [91]. A single PEPE token holder lost $1.39 million after signing one malicious Permit2 signature [92].

Forensic analysis of October 2024 incidents reveals coordinated attack patterns. On 10 October 2024, at 22:03:43 UTC, the fwdETH incident (transaction hash on Blast Chain—Layer 2 chain of Ethereum: 0x80bbc39...5b84591e3648d) resulted in $36 million losses through a malicious Permit2 signature. The attack targeted the victim's wallet 0xEab23c1E3776fAd145e2E3dc56bcf739f6e0a393, draining 15,079 fwdETH in two transfers: 12,817.15 fwdETH to 0x0605eDeE6a8b8b553caE09Abe83b2ebeb75516eC and 2261.85 fwDETH to 0xA963df55B326609a0cd205e85ca92d2a3c94DaB5. The attack exploited infinite approval through Permit2 signatures. Approximately two weeks earlier, on 27–28 September 2024, the spWETH attack ($32.43 million) demonstrated similar Permit2 exploitation patterns. The PEPE token incident of October 13, 2024, demonstrates the attack speed: victims signed malicious Permit2 signatures and lost $1.39 million in PEPE, MSTR, and APU tokens within approximately one hour of signing. Forensic verification: (1) Query https://blastscan.io/tx/0x80bbc39dd5c62b5368e1bf10d40969f5da8b45555de6bfbbcfe5b84591e3648d to view the complete transaction details on Blast Chain Block 9894004 (accessed 19 November 2025), (2) Review token transfers showing the two-

stage drain of 15,079 total fwDETH, (3) Cross-reference attacker addresses with known malicious contracts.

The attack methodology exploits the asymmetric nature of cryptographic signatures [93]. Malicious actors construct approval messages appearing to authorise small transactions while actually granting comprehensive token access. Phishing websites replicate legitimate DeFi interfaces, targeting users who previously interacted with protocols like Uniswap. Since the off-chain signing process prevents users from verifying the approval scope, attackers obtain valid signatures for malicious operations that blockchain verification accepts as legitimate.

Prevention requires strict signature verification procedures, including manual examination of recipient addresses, token amounts, and expiration times. Hardware wallets increase signing deliberation but cannot prevent users from approving malicious messages. The convenience-security trade-off remains unresolved, with Permit2's gas savings creating persistent attack surfaces as hijacking techniques become increasingly automated.

4.6. Smart Contract Risk

Smart contracts represent programmable agreements that execute automatically on blockchain networks, but their immutable nature makes security flaws particularly dangerous. In this paper, we mainly focus on EVM-compatible languages like Solidity, widely used by the community. Once deployed, smart contracts cannot be modified [94], meaning any vulnerabilities become permanent attack vectors (except that the smart contract is designed to be updatable by the developer). The transparency of permissionless blockchains allows anyone to examine contract code and identify weaknesses, creating both security benefits through public auditing and risks through attack discovery.

Recent data from OWASP's Smart Contract Top 10 analysis shows that access control vulnerabilities alone caused $953.2 million in losses during 2024, while logic errors resulted in $63.8 million in damages. Reentrancy attacks, despite being well-known since the 2016 DAO incident, still caused $35.7 million in losses [95]. These numbers demonstrate that smart contract security remains a major problem despite years of research and tool development. Developers face multiple security challenges during the development process that extend beyond code quality. Environment configuration often involves storing sensitive information like contract addresses, private keys, and API endpoints, while developers commonly use .env files for local development, exposing contract addresses in environment files creates limited risk since these addresses become public upon deployment. However, private keys and deployment credentials should never appear in .env files or any version-controlled code. Production deployments require secure key management through hardware wallets or dedicated key management services.

Contract replacement attacks present another risk where malicious actors deploy similar contracts with subtle modifications to trick users into interacting with compromised versions. Developers can prevent this through proper verification processes, including contract address verification on block explorers and implementing transparent deployment procedures. Many development frameworks now include automatic verification tools that publish source code during deployment, making contract authenticity easier to verify. Developers seeking audited smart contract templates should primarily use the OpenZeppelin Contracts library, which provides battle-tested implementations of common standards like ERC-20 tokens, ERC-721 NFTs, and access control mechanisms. OpenZeppelin maintains the most widely used collection of audited smart contract components, with implementations that have undergone multiple security audits and real-world testing [96]. The OpenZeppelin Contracts Wizard provides an interactive interface for generating cus-

tomised contracts based on these audited templates. The auditing process has evolved beyond manual code review to include automated analysis tools. A recent survey by Hejazi and Lashkari identified 256 smart contract vulnerability detection tools developed between 2018 and 2024, categorised by methodologies including fuzzing, symbolic execution, machine learning, and formal verification [97]; see also Section 7. These tools can detect common vulnerabilities like integer overflows, reentrancy issues, and access control problems, but they struggle with complex logic errors and business logic vulnerabilities.

Machine learning and AI techniques are increasingly used in smart contract auditing, though with important limitations. AI-powered tools excel at identifying known vulnerability patterns and can process large codebases quickly, but they often produce false positives and miss context-dependent security issues. Recent research shows that while AI can automate initial vulnerability scanning, human expertise remains necessary for understanding business logic flaws and attack vector combinations [98].

Common smart contract vulnerabilities fall into several categories based on analysis of real-world incidents. Access control issues remain the most financially damaging, often involving missing or incorrect permission checks that allow unauthorised users to execute privileged functions. Logic errors include incorrect mathematical operations, flawed state transitions, and improper handling of edge cases. Reentrancy vulnerabilities occur when external contract calls allow malicious actors to re-enter functions before state updates complete. Flash loan attacks exploit the temporary nature of uncollateralized loans to manipulate protocol states within single transactions [95].

Input validation problems create opportunities for attackers to provide malicious data that causes unexpected contract behaviour. Price oracle manipulation involves attacking external data sources that smart contracts rely on for price information. Unchecked external calls can fail silently, leading to incorrect assumptions about transaction success. These vulnerability categories reflect patterns observed across thousands of deployed contracts and billions of dollars in losses. The development ecosystem has responded with improved tools and practices, but adoption remains inconsistent. Formal verification methods can mathematically prove contract correctness for specific properties, though they require specialised knowledge and serious development time. Static analysis tools can identify potential vulnerabilities during development, while dynamic testing frameworks enable comprehensive testing of contract behaviour under various conditions. Educational resources and security frameworks continue to evolve. The OpenSCV taxonomy provides a hierarchical classification system for smart contract vulnerabilities, helping developers understand attack patterns and implement appropriate defences [99]. Industry groups have developed security checklists and best practice guides, though keeping pace with new attack vectors remains challenging.

Smart contract security requires a multi-layered approach combining secure coding practices, comprehensive testing, automated analysis tools, professional audits, and ongoing monitoring. The immutable nature of blockchain deployment makes prevention much more valuable than attempted remediation, placing a premium importance on security measures during the development phase. The cost of security failures, measured both in financial losses and in damaged reputation, justifies investing in proper security practices for smaller projects.

5. Privacy-Improving Technologies for Permissionless Blockchains

Privacy-enhancing technologies for permissionless blockchains fall into three categories: cryptographic techniques that hide transaction content (Sections 5.2–5.5), architectural approaches that restrict visibility (Section 5.3), and operational practices that distribute trust (Section 5.1). We evaluate each approach's privacy guarantees, performance trade-offs, and deployment barriers.

5.1. Multi-Signature Technique

Multisignature wallets distribute transaction control across multiple keyholders, requiring several parties to approve before funds can move. Instead of a single private key controlling assets, a multi-sig setup enforces collective authorisation at the protocol level. Common configurations include 2-of-3 or 3-of-5 schemes, where a subset of keyholders must sign each transaction.

This approach addresses several privacy and security concerns in blockchain systems. When one keyholder's device gets compromised or one party acts maliciously, attackers cannot drain funds without obtaining additional signatures. The arrangement also obscures which specific participants authorised each transaction: observers see that multiple signatures validated the payment, but cannot determine which subset of keyholders actually cooperated. Multi-sig wallets have become standard infrastructure for DAOs, protocol treasuries, and institutional custody. Projects like Gnosis Safe dominate the Ethereum ecosystem, managing billions in treasury assets. The scheme protects against both external attacks and internal malfeasance, making it suitable for scenarios where trust must be distributed among multiple parties.

Modern implementations use threshold signature schemes [100] that appear identical to single-signature transactions on-chain. Bitcoin's Taproot upgrade and threshold ECDSA protocols allow multiple parties to cooperatively generate signatures without revealing the multi-party nature of the authorisation. This reduces transaction costs and improves privacy compared to traditional multi-sig scripts, where the number of required signatures is visible in transaction data.

The operational challenges involve coordination overhead and key recovery. Each additional required signature increases the complexity of executing transactions. Organisations using 5-of-9 schemes must coordinate among five people for every payment, introducing delays. Transaction fees scale with signature count, making high-threshold configurations expensive for frequent operations. If enough keyholders lose access to their keys, the funds become permanently locked, yet distributing key material broadly increases exposure to compromise.

5.2. Zero-Knowledge Proofs

Zero-knowledge proofs allow one party to prove that they know something without revealing what they know. In blockchain contexts, this means that users can prove the validity of the transaction, the ownership of the assets, or the compliance with the rules without exposing the underlying data. A prover demonstrates knowledge of secret information to a verifier, who becomes convinced of the claim's truth while learning nothing about the secret itself. The technology addresses fundamental privacy limitations in public blockchains. Bitcoin and Ethereum transactions expose amounts, addresses, and transaction graphs to anyone running a node. Zero-knowledge proofs break this transparency by allowing verification without disclosure. Users can prove they have sufficient funds to make a payment without revealing their balance, or demonstrate membership in a group without identifying which member they are.

Two main proof systems dominate blockchain applications. zk-SNARKs produce small, quickly verifiable proofs regardless of computation complexity. Groth16, the most widely deployed zk-SNARK construction, generates proofs of only 192 bytes that verify in under 10 milliseconds. This efficiency made zk-SNARKs practical for blockchain integration, where every byte costs gas and every verification must be fast. Zcash pioneered its use for private transactions, hiding sender, receiver, and amount while still allowing network validation. The drawback of zk-SNARKs involves the trusted setup ceremony. Generating

the proving and verification keys requires creating and destroying secret randomness. If this toxic waste leaks, attackers could forge proofs and create counterfeit assets. Projects address this through multi-party computation ceremonies where hundreds of participants contribute randomness, requiring only one honest participant to maintain security. Zcash is Powers of Tau ceremony involved over 200 contributors across multiple rounds.

zk-STARKs eliminate trusted setup requirements through different cryptographic foundations. These proofs rely on hash functions and information-theoretic security rather than elliptic curve assumptions, making them resistant to quantum computers. STARKs generate larger proofs than SNARKs but scale better for complex computations. StarkWare uses them for Ethereum Layer 2 scaling, batching thousands of transactions into a single proofs that verify on the mainnet.

Current Ethereum scaling solutions demonstrate zero-knowledge proofs' practical impact. zkSync, Polygon zkEVM, and Scroll use zero-knowledge rollups to process transactions off-chain while posting validity proofs on-chain. These systems batch thousands of transactions into one proof, reducing gas costs by over 90% compared to individual mainnet transactions. Users get near-instant finality and lower fees while maintaining Ethereum's security guarantees.

The computational demands remain substantial. Generating proofs for the complex circuits that zero-knowledge techniques need requires powerful hardware, often GPUs or specialised ASICs. Modern zkEVM circuits contain millions of constraints, with memory requirements exceeding standard consumer hardware. Projects operate dedicated proofing infrastructure, introducing centralisation risks for the few entities that can afford to generate proofs. Research continues on proof aggregation and recursive composition to distribute this computational burden more effectively.

5.3. Permissioned Frameworks: Hyperledger Fabric

Hyperledger Fabric addresses permissionless blockchain privacy concerns through a fundamentally different architectural approach, implementing permissioned network structures where membership service providers authenticate all participants before network access. This design enables granular privacy controls through channel-based data segregation, restricting transaction visibility to authorised organisations while implementing private data collections for sensitive information that requires additional access controls. The endorsement policy framework requires transaction validation from specified peer organisations before ledger commitment, ensuring that smart contract execution occurs within isolated environments that prevent cross-contract information leakage and unauthorised data access.

The PnR architecture proposed by Banach demonstrates how Hyperledger Fabric can enforce privacy through structural mechanisms rather than relying solely on rewards for honest behaviour [24]. The Hyperledger architecture implements a two-organisation deployment where each organisation operates distinct peer types within a management layer. Organisation peers include anchor peers that maintain network topology, endorser peers that execute and validate transactions through chaincode, and general peers that maintain ledger copies. Membership service providers authenticate participants at each peer and manage cryptographic certificates. Channels connect peers across organisations, creating isolated communication pathways that restrict information flow to authorised parties. The ordering service receives endorsed transactions and organises them into blocks before distributing them to channel members. Client applications interact with the network through SDKs or REST APIs, enabling transaction submission and ledger queries. This layered structure allows organisations to maintain privacy boundaries while participating in shared consensus, as transactions remain visible only to channel participants rather

than the entire network). The punishment mechanism operates through smart contract enforcement and controlled revocation of anonymity. When participants violate network policies, the system can either deny service access or revoke pseudonymous protections, revealing the identity behind malicious actions. This approach differs from traditional blockchain incentive structures, which reward block creation, in that it focuses on deterrence through accountability. The chaincode deployed at endorser peers evaluates transaction validity against predefined policies before endorsement, creating multiple checkpoints where improper behaviour can be detected and prevented.

Recent developments in the Hyperledger Fabric ecosystem reflect both growing pains and continued evolution. The platform reached a milestone with the release of version 3.0 in 2024, introducing improved performance and new features [101]. However, the landscape has shifted, with Visa departing from the consortium and the Linux Foundation discontinuing its Hyperledger Fabric developer courses. Despite these changes, development continues, with Chainlink joining the ecosystem to explore enterprise blockchain integration. Performance analysis by Yu et al. demonstrates that endorsement strategy optimisation can improve transaction throughput, with their proposal distribution algorithm reducing latency in high-load scenarios [102]. Healthcare implementations have shown promising results, with Al-Sumaidaee et al. achieving sub-second transaction times in private network deployments, though scalability remains constrained by the endorsement process [103]. Ke and Park's performance modelling reveals that transaction throughput scales linearly up to a threshold determined by endorser capacity, after which the system experiences degradation due to cryptographic verification overhead [104].

Practical deployment challenges emerge from the permission model itself. Nedaković et al. implemented a healthcare trust relations system and identified coordination overhead as a major bottleneck, particularly when multiple organisations must reach consensus on transaction endorsement [105]. Latency modelling by Abang et al. in IoT contexts shows that network topology and peer placement can introduce milliseconds to seconds of delay, making real-time applications difficult without careful architecture design [106]. Multimedia applications face particular challenges, as Sharma et al. discovered that large file storage requires off-chain solutions, with Fabric maintaining only metadata references to external storage systems [107].

Fabric's modular consensus architecture allows organisations to select appropriate ordering mechanisms without compromising privacy requirements, while recent Fabric 2.x and 3.0 implementations introduce certificate-based identity management and hardware security module integration for enhanced cryptographic operations. Organisations can implement zero-knowledge proofs within chaincode to verify claims without revealing underlying data, creating hybrid systems that combine the benefits of permissioned networks with advanced cryptographic privacy techniques, while this approach provides strong privacy guarantees within consortium environments, it fundamentally alters the permissionless nature of traditional blockchain networks and may not be suitable for applications requiring open participation and decentralised governance. The platform's future remains tied to enterprise adoption patterns, where privacy requirements often outweigh the philosophical preference for permissionless access.

5.4. Ring Signatures

Ring signatures allow someone to sign a message on behalf of a group without revealing which group member created the signature. Anyone examining the signature knows it came from one of the group members, but cannot determine which one. This provides anonymity within the group: if ten people form a ring, each member has plausible deniability, since the signature could have come from any of them. The mechanism works

by combining the public keys of all group members into a cryptographic structure in which the actual signer uses their private key along with the public keys of other members. The result proves that someone in the group authorised the message, but the proof itself reveals nothing about who specifically signed. All ring members appear equally likely to be the signer based on cryptographic evidence.

Monero uses ring signatures as its primary privacy mechanism. When Alice sends a transaction, her real output gets mixed with several decoy outputs from other users' past transactions. The ring signature proves that Alice controls one of these outputs without identifying which one. Observers see that some output in the ring got spent, but cannot determine whether it was Alice's or one of the decoys. Linkable ring signatures extend the basic concept to prevent double-spending while preserving anonymity. Each private key generates a unique key image when it is used to create a signature. If someone tries to spend the same output twice, both transactions produce identical key images, allowing nodes to reject the duplicate. The key image itself does not reveal which ring member was signed, but its uniqueness prevents reuse. This solves the double-spend problem without requiring transaction transparency. Monero's CLSAG protocol improved ring signature efficiency by reducing the signature size by 25% compared to earlier schemes. Smaller signatures mean lower transaction fees and reduced blockchain bloat. The verification process also became faster, allowing the nodes to validate transactions more efficiently. These improvements matter for network scalability: As the transaction volume grows, efficient cryptography becomes necessary to keep the system usable.

Ring size directly affects privacy strength. Larger rings provide better anonymity, since each member represents a smaller fraction of possible signers. A ring of 16 means each member has 6.25% probability of being the signer from an observer's perspective, while a ring of 100 provides 1% probability. However, larger rings increase transaction size, verification time, and fees. Monero currently uses rings of 16 as a balance between privacy and practicality. The anonymity set only includes other blockchain users who could plausibly be involved in the transaction. If an attacker controls most of the decoy outputs, they can deduce which output was real through elimination. Network-level analysis can also narrow the possibilities by correlating transaction timing and amounts. Ring signatures provide strong privacy guarantees, but work best when combined with other protections like Tor usage and avoiding amount correlation.

Recent research explores post-quantum ring signatures resistant to quantum computer attacks, though these constructions produce larger signatures and slower verification; See Section 8. Integration with confidential transactions that hide amounts remains an active development area, as combining multiple privacy techniques introduces complexity in proof construction and verification.

5.5. Stealth Addresses

Stealth addresses prevent address reuse by generating a unique, one-time address for each transaction while allowing the recipient to control all funds with a single private key. This breaks the link between payments that would otherwise be visible on-chain when multiple senders pay the same public address.

The recipient publishes a master stealth address—essentially a public identifier that never appears in actual transactions. When someone wants to send funds, they use this master address to derive a fresh, one-time address that only the intended recipient can access. Each payment creates a different on-chain address, so observers cannot tell that multiple transactions went to the same person. The blockchain shows payments to seemingly unrelated addresses, masking the recipient's transaction history. The cryptographic

mechanism relies on elliptic curve key derivation. The sender generates random data and combines it with the recipient's public information to create a shared secret that only the sender and recipient can compute. This shared secret derives both the one-time payment address and the corresponding private key needed to spend those funds. The sender publishes the random data alongside the payment, allowing the recipient to scan the blockchain, recognise payments intended for them, and compute the necessary private keys.

Monero pioneered stealth addresses as a core privacy feature, making them mandatory for all transactions. Every Monero payment goes to a unique address that appears unconnected to the recipient's public identity. This prevents address clustering and transaction graph analysis that plague Bitcoin's transparent ledger. Ethereum adoption has been slower due to wallet complexity and gas costs. The ERC-5564 standard provides a framework for stealth addresses on EVM chains, with Umbra Protocol implementing the specification across the Ethereum mainnet and several Layer 2 networks. Since launch, Umbra has processed over 12,000 stealth transactions, demonstrating demand for recipient privacy despite added complexity.

The main operational challenge involves scanning. Recipients must check every transaction to see if it was meant for them, computing shared secrets and comparing derived addresses against on-chain data. This becomes computationally expensive as transaction volume grows. View tags [108] optimise this process by including a single-byte hint with each payment. Recipients check the view tag first—a trivial computation—and only perform expensive elliptic curve operations when the tag matches. This reduces scanning overhead by over 95%, making regular monitoring practical. Wallet integration remains the biggest barrier to adoption. Users must understand that stealth addresses require scanning and that funds appear in different addresses each time. Most wallet software lacks built-in support, forcing users to rely on tools. The user experience degrades compared to simple address reuse, though the privacy benefits are substantial.

Stealth addresses work best combined with other privacy techniques. They hide the recipient but not the amount or sender. Pairing them with confidential transactions that encrypt amounts and ring signatures that obscure senders would provide comprehensive privacy. Monero demonstrates this approach, while Ethereum implementations focus on recipient privacy as a first step toward broader confidentiality.

5.6. zkEVM Scaling Solutions

The zkEVM implementations represent a convergence of Ethereum Virtual Machine compatibility with zero-knowledge proof verification, offering enhanced privacy and scalability through cryptographic computation verification rather than optimistic fraud proofs. Polygon zkEVM processes transactions in EVM-compatible execution environments before generating validity proofs for Ethereum mainnet verification, employing a hybrid proving system that uses STARKs for computation verification and SNARKs for final proof compression to optimise both security and efficiency. Batch sizes of 1000–10,000 transactions generate single proofs [109] consuming approximately 300,000 gas for verification, reducing individual transaction costs by 90–95% compared to Ethereum mainnet execution while maintaining identical security guarantees through mathematical proof verification.

zkSync Era implements EVM compatibility through a custom virtual machine design supporting Solidity and Vyper languages, processing over 2000 transactions per second with finality times under 10 min and achieving total value locked exceeding $1.1 billion by Q3 2024. The architecture demonstrates practical adoption of zero-knowledge scaling while maintaining developer familiarity and tooling compatibility. Privacy features emerge naturally through transaction batching, where individual operations become computationally indistinguishable within proof aggregation, preventing correlation of specific transactions

to senders without additional metadata analysis. Future implementations may integrate stealth addresses and confidential transaction amounts for enhanced privacy guarantees, creating comprehensive privacy-preserving execution environments.

Computational requirements for proof generation necessitate specialised hardware infrastructure, with modern zkEVM deployments requiring 64GB+ RAM and dedicated GPU acceleration for practical proof generation times. However, verification costs remain constant regardless of batch size, creating substantial economies of scale for high-throughput applications and enabling cost-effective privacy-preserving computation at scale.

6. Secure Development Lifecycle for Blockchain Systems

The gap between academic research on blockchain privacy and practical implementation often widens due to inadequate security practices during the development lifecycle. Projects that understand privacy risks theoretically still fall victim to attacks because they fail to implement proper security controls from the earliest stages of development. The Web3 ecosystem encompasses diverse infrastructure components and applications [110], each presenting unique security challenges at the system, developer, and user levels [111]. As Web3 technologies evolve beyond simple blockchain applications to encompass decentralised autonomous organisations, non-fungible tokens, and complex financial protocols [112], the importance of structured security practices becomes paramount [113]. Building on the security practice framework established by SlowMist [114], we outline a structured approach that researchers and development teams should consider when building permissionless blockchain applications. Fuzzing tools like Echidna [115] and Foundry [116] can automatically generate test cases that explore edge conditions developers might miss. Table 3 illustrates the structured development lifecycle framework.

Table 3. Secure Development Lifecycle Framework for Blockchain Systems.

Phase	Security Activities	Blockchain Controls	Measurable Outcomes
Preparation	Isolated environments; hardware wallets for production; credential exclusion from version control	Multi-signature deployment approval; automated credential scanning; hardware device signing	Zero production keys in repositories; hardware wallet adoption rate; multi-signature coverage
	Threat modeling: attack surfaces including phishing, signature hijacking, MEV exploitation; on-chain and off-chain trust boundaries	Framework support for hardware wallet integration	Documented threat surfaces; framework security features enabled
Development	Audited smart contract libraries; checks-effects-interactions pattern; access control mechanisms; input validation	Overflow protection (Solidity 0.8+); property-based fuzzing; boundary condition testing	Audited library adoption ratio; access control coverage; fuzzing test coverage
	Content Security Policy headers; input sanitization; transaction preview with decoded parameters; RPC response validation	External data verification; calldata decoding before signature requests	CSP implementation status; validation coverage
	Transport security; request rate limiting; dedicated RPC infrastructure; network segmentation	Self-hosted nodes prevent metadata leakage; transaction routing with MEV protection	Dedicated infrastructure adoption; rate limiting effectiveness
Release	Code freeze period; external security audit; critical and high-severity finding remediation; regression testing	Protocol economic model verification; game-theoretic assumption validation	Critical findings resolved; code freeze duration (days)
	Incremental deployment; contract address verification on block explorers; source code publication; emergency pause mechanisms	Pause and upgrade path testing under realistic network conditions	Contract verification completion time; source publication status

Table 3. *Cont.*

Phase	Security Activities	Blockchain Controls	Measurable Outcomes
Runtime	Permission change monitoring; large value transfer alerts; anomalous transaction pattern detection; malicious address interaction tracking	Upgradeable contract state monitoring; transaction flow analysis	Median detection latency; false positive rate; monitored attack vector coverage
	Infrastructure security updates; access control enforcement; automated balance reconciliation	Component isolation; continuous state verification	Critical vulnerability patch time; reconciliation frequency
	Vulnerability disclosure programs; pre-authorized emergency response; pause and upgrade capabilities	Economic incentive alignment for responsible disclosure	Median incident containment time; disclosure versus exploitation ratio
Awareness	User education on signature authorization risks and social engineering; ecosystem vulnerability tracking; periodic security exercises	Proactive vulnerability assessment following ecosystem incidents	Security drill frequency; incident-to-review response time

Framework phases incorporate measurable security controls enabling quantitative maturity assessment. Common tool implementations include git-secrets for credential scanning [117], Echidna and Foundry for property-based fuzzing [115,116], OpenZeppelin for audited contract templates [96], and Hardhat or Foundry as development frameworks [116,118].

The foundation of secure Web3 development begins with the proper configuration of the environment and key management practices. Developers face immediate security decisions when setting up local development environments, particularly regarding how to handle private keys, API credentials, and contract addresses. The common practice of storing sensitive credentials in `.env` files presents different risk levels depending on what information gets stored. Table 4 assigns these security responsibilities across development teams using the RACI framework.

Table 4. Role Responsibility Matrix for Key Security Controls.

Control Point	Developer	Security	Operations	Lead
Environment setup, credential scanning	R	A	C	I
Hardware wallet deployment integration	R	C	A	I
Threat model documentation	C	R	C	A
Contract security patterns	R	A	I	I
Fuzzing and adversarial testing	R	A	I	I
Infrastructure hardening, RPC nodes	I	C	R	I
External audit coordination	C	A	I	R
Code freeze decision	I	C	I	R
Deployment execution	C	C	R	A
Monitoring system deployment	C	R	A	I
Incident response execution	C	R	A	R
Quarterly security drills	C	R	C	A

R = Responsible (executes), A = Accountable (approves), C = Consulted (advises), I = Informed (notified) following standard RACI framework [119]. Most organisations adapt this based on team size.

Smart contract addresses can safely appear in environment files since deployment makes them public regardless. However, private keys and deployment credentials must never exist in `.env` files or any version-controlled code. A single accidental commit containing private keys to a public repository can lead to immediate fund drainage: automated bots scan GitHub commits for exposed private keys and drain associated wallets within minutes of exposure. Production deployments require hardware wallets or dedicated key management services. Development and testing should use separate keys with minimal funds, isolated from any production assets. The Hardhat [118] and Foundry [116] development frameworks support hardware wallet integration for deployment scripts, allowing developers to sign transactions without exposing private key material to the development environment.

Requirements documentation at this stage should specify threat models, trust assumptions, and security boundaries before any code gets written. Projects need clear documentation of what data should remain private, which components handle sensitive information, and where trust boundaries exist between on-chain and off-chain systems. Business process documentation should map user flows through security-critical operations, identifying points where users face risks from phishing attacks, signature hijacking, or malicious contract interactions. Role-based enforcement of these controls requires clear allocation of responsibilities across development, security, operations, and leadership teams. This is shown in Table 5.

Table 5. Role-by-Control-Point Matrix for SDLC.

Control Point	Developer	Security	Operations	Lead
Preparation Phase				
Environment configuration	R	C	A	I
Credential management	R	A	C	I
Threat model definition	C	R	I	A
Framework selection	R	C	I	A
Development Phase				
Smart contract implementation	R	C	I	I
Unit test development	R	C	I	I
Code review	C	R	I	I
Fuzzing campaigns	I	R	I	I
Static analysis	C	R	I	I
Infrastructure hardening	I	C	R	I
Release Phase				
Code freeze enforcement	I	C	I	R
External audit coordination	C	A	I	R
Vulnerability remediation	R	A	I	I
Staging deployment	C	C	R	A
Mainnet deployment approval	I	C	I	R
Contract verification	C	R	A	I
Runtime Phase				
On-chain monitoring	C	R	A	I
Infrastructure monitoring	I	C	R	I
Incident detection	C	R	A	I
Emergency response execution	C	R	A	R
Balance reconciliation	C	C	R	I
Post-incident analysis	C	R	C	A
Continuous Improvement				
Security drill coordination	C	R	C	A
Bug bounty management	I	R	I	A
Ecosystem threat tracking	C	R	C	I

Responsibility codes: **R** = Responsible (performs the work), **A** = Accountable (final authority and approval), **C** = Consulted (provides input), **I**= Informed (kept updated). This RACI framework follows standard project management methodology [119].

6.1. Development Process and Smart Contract Security

Smart contract development introduces unique security challenges due to code immutability and the transparent nature of blockchain execution. The Web3 development paradigm shifts from traditional centralised architectures to decentralised systems where toolkits and frameworks enable efficient innovation despite distributed information [113]. Security considerations must be integrated into the coding process itself rather than being added as an afterthought. Using audited libraries like OpenZeppelin Contracts provides battle-tested implementations of common patterns, reducing the risk of introducing novel vulnerabilities. However, the evaluation of Web3 applications reveals that existing implementations often fall short on security and adoptability as claimed [111], necessitating rigorous development practices.

Security coding requirements should enforce defensive patterns from the start. This includes Check Effects Interactions to prevent reentrancy, proper access control on privileged functions, input validation on all external calls, and careful handling of arithmetic operations to prevent overflow and underflow issues. Solidity version 0.8.0 introduced automatic overflow checks, but developers must still test edge cases involving zero values, maximum token approvals, and unusual input combinations. The test case requirements extend beyond functional correctness to include adversarial scenarios. Unit tests should attempt to exploit potential vulnerabilities by calling functions with extreme values, trying to reenter contracts, testing authorisation bypasses, and simulating malicious contract interactions. Fuzzing tools like Echidna and Foundry's fuzzer can automatically generate test cases that explore edge conditions developers might miss.

Front-end security matters as much as contract code. Web interfaces represent the primary attack surface for most users, who interact with contracts through browser-based applications rather than directly. Configuration requirements should enforce Content Security Policy headers, implement proper input sanitisation to prevent cross-site scripting, validate all data from external sources, including RPC responses, and use secure random number generation for any cryptographic operations. The interface should clearly display transaction details before signature requests, helping users identify signature hijacking attempts. Smart contract security coding requirements should enforce defensive patterns from the start. Blockchain-based verification systems have been applied to integrity verification in educational platforms [120], demonstrating lightweight approaches to content authentication.

Server environments hosting front-ends or running backend services need hardening against common attack vectors. This includes properly configured HTTPS, secure headers, rate limiting to prevent denial of service, and isolation between different components. RPC endpoint selection matters—projects should run their own nodes when possible rather than relying entirely on third-party providers that can log IP addresses and transaction patterns.

6.2. Release Process and Security Auditing

The release phase represents the last opportunity to catch vulnerabilities before deployment to the mainnet. Code freeze requirements prevent last-minute changes that could introduce bugs or bypass previous testing. Regression testing ensures that bug fixes did not break existing functionality or introduce new vulnerabilities. Test coverage should include all critical paths through the contract code, with particular attention to functions handling value transfers or permission changes. Reputable security audits provide external validation of contract security. However, audits have limitations—they represent snapshots in time and cannot guarantee the absence of vulnerabilities. Projects should provide auditors with comprehensive documentation of intended behaviour, known limitations, and areas of particular concern. The audit process works best when auditors have time to thoroughly review code rather than rushing to meet deployment deadlines.

Audit reports often identify issues ranging from critical vulnerabilities to gas optimisation opportunities. Teams must address critical and high-severity findings before deployment. Medium and low-severity issues require judgment calls about risk versus deployment timeline pressure. The temptation to deploy despite audit findings has led to numerous exploits where known vulnerabilities were ignored due to competitive pressure or overconfidence.

6.3. Runtime Monitoring and Incident Response

Deployment does not end security responsibilities—runtime monitoring becomes essential for detecting attacks in progress or identifying suspicious activity before major losses occur. Smart contracts require different monitoring approaches than traditional software since on-chain activity is permanently visible, but off-chain systems may need protection from unauthorised access. Runtime security monitoring should track permission changes in contracts with upgradeable components, watching for unauthorised calls to privileged functions. Fund movements need monitoring through alerts on large transfers, unusual transaction patterns, or interactions with known malicious addresses. Periodic reconciliation between expected and actual balances can identify discrepancies from accounting errors or exploitation attempts.

Environment hardening protects the infrastructure supporting blockchain applications, while smart contracts run on the decentralised network, supporting services like RPC endpoints, indexers, and front-end hosting require traditional security measures. Regular updates, access controls, and network segmentation reduce the attack surface on these centralised components.

Bug bounty programs incentivise security researchers to report vulnerabilities rather than exploit them. Platforms like Immunefi host bug bounties for blockchain projects, offering rewards scaled to vulnerability severity. Well-structured bounties specify scope, provide clear reporting channels, and commit to response timeframes. The payout structure should make responsible disclosure more profitable than exploitation for most researchers. Emergency response procedures need definition before incidents occur. Response teams should have pre-authorised access to pause mechanisms, upgrade capabilities, or other incident response tools. The decision-making process during emergencies must balance speed against avoiding panic-driven mistakes. Clear communication channels allow coordination between developers, security teams, and community members during crises. Stop-loss disposal requirements limit damage when breaches occur—pausable contracts allow emergency halts while teams assess the situation, though pause mechanisms themselves become targets if not properly protected.

Post-incident procedures should document tracking requirements for understanding attack vectors, problem-solving processes for implementing fixes, and security release protocols for deploying patches. Issue analysis determines root causes and identifies similar vulnerabilities elsewhere in the codebase.

6.4. Security Awareness and Continuous Improvement

Security awareness cultivation extends beyond the immediate development team to include all stakeholders in a Web3 project. The Web3 ecosystem faces challenges at multiple levels—including system-level issues related to infrastructure reliability, developer-level challenges around tooling and standards, and user-level concerns about usability and security [110]. Understanding these challenges requires recognising Web3 as an evolving technological landscape where decentralisation, blockchain integration, and user empowerment create both opportunities and risks [112,121]. Users need to be educated about signature verification, the risks of contract interactions, and how to identify phishing attempts. Many attacks succeed not through technical exploitation but through social engineering of users who approve malicious transactions.

Tracking security incidents across the broader ecosystem provides early warning of emerging attack patterns. When one project suffers a specific type of exploit, similar projects should immediately audit their code for the same vulnerability class. Attack techniques spread quickly after the first flash loan attack, many projects discovered they

were vulnerable to the same approach. Security awareness assessment and drills prepare teams for various threat scenarios, revealing gaps in monitoring or response capabilities before real incidents occur. The integration of these security practices throughout the development lifecycle addresses many of the privacy and security concerns discussed in earlier sections. Proper key management prevents private key theft, which accounts for most blockchain security losses. Smart contract security practices mitigate the risks from vulnerable contract code. Runtime monitoring enables early detection of the MEV attacks, dust attacks, and signature hijacking attempts that target users. This structured approach bridges the gap between understanding privacy risks theoretically and implementing robust defences in production systems.

Security awareness cultivation extends beyond the immediate development team to include all stakeholders in a Web3 project. Users need education about signature verification, the risks of contract interactions, and how to identify phishing attempts. Many attacks succeed not through technical exploitation but through social engineering of users who approve malicious transactions.

Tracking security incidents across the broader ecosystem provides early warning of emerging attack patterns. When one project suffers a specific type of exploit, similar projects should immediately audit their code for the same vulnerability class. Attack techniques spread quickly across the ecosystem. After the first flash loan attack, many projects discovered they were vulnerable to the same approach.

The SDLC framework requires measurement to demonstrate effectiveness. Projects can track detection latency by recording the time from vulnerability introduction to discovery. The 2024 incident data shows that automated monitoring reduced median detection times from 19 h to 11 min for unauthorised privilege escalation attempts [22]. Response effectiveness can be measured through quarterly drills that simulate attacks and record time-to-containment. The October 2024 Permit2 attacks demonstrated this clearly: projects with pre-authorised pause mechanisms contained incidents within 8–15 min and limited losses to under 5% of total value, while projects lacking these capabilities saw complete fund drainage [92]. Pre-deployment metrics also correlate with outcomes. Projects with external audits covering more than 75% of critical functions experienced 68% fewer exploits in their first six months compared to projects with limited or no audit coverage [22]. These metrics, linked to the threats identified in Section 4 and shown in Tables 3 and 4, help teams assess whether their security investments actually reduce risk.

7. Formal Approaches to Blockchain Security Concerns

When important concerns like security and privacy are delegated to software, the functional correctness of the software is of primary concern. This is of high importance in the wider world, but is even more pressing in the world of blockchain. In the wider world, if a piece of software proves to be faulty, it can normally be replaced with an improved version, perhaps at some, even considerable, cost, but the process of replacement nevertheless mitigates the damage that the faulty version gave rise to. In the blockchain world, once a smart contract is deployed on-chain, it quickly becomes immutable, and any faults it contains get set in stone. A range of problems this causes has been discussed earlier in the paper. Worse, if the blockchain protocol itself is flawed in some way, there is little that can be done once the blockchain itself has attracted sufficient traction. All the attacks around value extraction discussed earlier in the paper are based on exploiting fundamental features of the relevant protocol, are therefore consigned to be 'facts of life' barring a hard fork in the relevant protocol.

One approach that addresses these issues is the verification of software using formal techniques. Formal methods, as they are generically known, emerged to address the so-called software crisis in the very late 1960s, growing through the 1970s, and by now encompass a large range of specific techniques which can get applied at all levels of the development stack. The literature is by now huge. A good perspective on the field can be gained from recent major conferences, such as the most recent edition of the Formal Methods Conference [122].

Formal verification techniques aim to avoid errors before the fact', by proving that the system under scrutiny does not have them. This is a big challenge because one has to be able to know what is correct and what is erroneous, and, given the system in question or a model of it, one has to be able to correctly find errors while simultaneously avoiding misidentifying either correct or erroneous features of the system or its model.

The survey [123], in [122], gives a timely overview of the wide variety of approaches to the verification problem that have been developed by now. The key fact that makes the field difficult is that almost all questions that one might ask in the hope of gaining useful information about a candidate system are undecidable. So the best that we can hope for is to obtain partial answers. The consequences of this are discussed in detail in [123].

It is fair to say that the deployment of formal methods techniques to address the problems inherent in the world of blockchain can be described as one of the niche application areas of formal verification. Various key areas within the blockchain world can be identified as specific focus topics for formal verification. Two key areas stand out.

The first of these is the verification of the virtual machine that supports the protocol of some given blockchain of interest. Several such works are available in the literature, for instance [124–129]. Given its popularity and wide adoption, it is not surprising that Etherium and the Etherium Virtual Machine specifically, have been the focus of much attention and figure prominently among these. The Financial Cryptography and Data Security conferences showcase much contemporary work in this area.

The second key area is the verification of smart contracts. This stands in contrast to the verification of virtual machines and protocols. The verification of virtual machines and protocols can be understood as a relatively well defined problem, in the sense that any given protocol is required to achieve some quite tightly defined objectives, and a lot of the objectives that different protocols are intended to meet are relatively similar across the range of different protocol types. By contrast, the challenge with smart contracts is that they may be required to meet any number of objectives that are taken from the application domain, i.e., that are user-defined. Since the remit of possible application domains is extremely large, a 'one size fits all' approach to smart contracts verification is infeasible, and an application-specific approach is needed. In [130], there is a survey of the application of formal techniques to smart contracts, mirroring the landscape set out in [123]. As is to be expected, Ethereum atracts the most interest, see e.g., [131,132]. And representative of the broader scope of smart contract verification are works such as [129,133,134].

The application of formal methods to security concerns, such as privacy, raises a whole new set of questions that mere conformance to functional correctness—as for virtual machines, protocols or smart contracts—has no need for, while the development of functional correctness of a protocol or contract has to assure that correct behaviour is not spoiled by the lower-level detail that is introduced during development, conformance to security concerns has to ensure that, during development, the lower level detail that is introduced, does not leak additional security-critical information, even if the development is in all other functional respects, completely correct. This is a considerable additional challenge that many Formal techniques are unable to address 'out of the box'. A comprehensive survey of

formal techniques applied to security concerns can be found in [135], and it emphasises that formal methods, rigorous as they are are, cannot find errors or vulnerabilities of a type that have not been imagined. This observation is not restricted to the formal world, of course. The security of digital systems would not be the arms race that it currently is if it were possible to prove, once and for all, that a system is perfectly secure. And this applies in no less measure to privacy concerns in blockchain. For a survey of current approaches to security in blockchain using formal approaches see [136].

The preceding comments notwithstanding, the fast pace of innovation in blockchain implies that the proportion of the blockchain ecosystem that is subjected to formal techniques is rather small. Developers on the whole tend to not be users of such techniques. However, the preceding sections have mentioned a small number of key areas in which key systems have been subjected to formal verification.

8. Quantum Threats to Blockchain Security

Since the invention of Shor's algorithm for polynomial-time factoring of large numbers in 1994 [137,138], the previously sleepy niche world of quantum computing was set ablaze in 1994. Interest has been rife in the development of quantum computers that will undermine the security of most of what takes place on the internet. Unfortunately (or fortunately, depending on one's point of view), laboratory physics has been less enthusiastic to concur. Despite the investment of billions of dollars worldwide, there is yet to be a quantum computer that can perform at anywhere near enough the level needed for breaking RSA, DH, ECC, and other public key protocols, and the myriad applications that have been built on top of them in today's internet.

Despite the previous observation, research into quantum secure cryptography, also referred to as post-quantum cryptography, is proceeding vigorously. The weak point of the existing cryptographic primitives, which makes them vulnerable to a quantum attack, is that they rely on a 'hidden subgroup problem'. This means that they conceal a periodic algebraic structure whose concealment underpins the relevant security guarantee. The quantum Fourier transform is able, in polynomial time, to reveal the relevant periodicity, after which the secret it conceals can be inferred. Post-quantum Cryptography seeks to create cryptographic primitives that are not based on a hidden subgroup problem, and reassuring progress has been made. See [139] for an informal account, and [140] for a textbook presentation, though one lacking the most recent advances. By now, cryptographic primitives that are believed to be quantum-secure have been established. NIST has published standards for these with the intent of enabling smooth interworking of applications [141–143]. In principle, such primitives can be used to replace the current quantum-vulnerable ones, in all contexts where they need to be used. There are some performance penalties, and inevitable integration issues for production systems, but a moderate performance penalty is a small price to pay if the alternative is loss of security. Besides the relatively well-established and standardised quantum-secure primitives, many other avenues continue to be explored. Many of these are based on mathematics, behind which there is often a rich algebraic structure, which can lead to unexpected breaks. So the field is in great flux at present.

For most blockchain architectures, two basic cryptographic techniques play an important part. Firstly, there is hashing, whereby the chain of hashes that binds the chain into an immutable sequence of blocks makes the chain to all intents and purposes, immutable. By and large, hashing techniques are immune to quantum attacks, as they are not based on hidden subgroups, being instead based on large numbers of simple arithmetic and bitwise operations that create sufficient confusion to obscure the original plaintext being hashed quite effectively. Secondly, there is public key cryptography. This is used

for 'authentication at a distance' whereby a transaction signed with a private key can be depended on by a remote party, based on the presumption that only the true owner of whatever the transaction refers to could have signed it in a way that the relevant public key can confirm. Unfortunately, this is precisely the kind of cryptography that is vulnerable to quantum attack, as it is typically based on RSA, DH, ECC, and related techniques. Breaking these primitives can not only reveal the signer's details, but can enable the faking of public key signatures, while the doxing of blockchain actors may be unpleasant for the parties concerned, the breakdown of signature security would be catastrophic for the trust framework that justifies the use of blockchain in any event. The availability of the standardised post-quantum cryptographic primitives we have mentioned indicates that, in principle, blockchains could be re-engineered to use them in case of need.

It is clear that the quantum world does not guarantee or undermine privacy concerns in blockchain per se. Rather, the cryptographic primitives at issue are vital ingredients of complex cryptographic protocols that achieve aspects of privacy as a whole. So the risk to privacy is proportional to the risk to the cryptographic primitives involved.

For now, the prospect of the collapse of blockchain trust for quantum reasons remains relatively remote. But vigilance is prudent, and investigation of quantum-safe blockchain architectures is proceeding. Given the large amount of hype surrounding quantum computing, a large number of blog-style articles and posts can be readily found on the web. In the academic literature, too some articles may be found, e.g., [144–147]. They restate the position outlined above with more context and detail. So It is fair to say that, at present, the non-quantum threats discussed earlier in the paper poses an immediate and far greater threat to blockchain security than anything based on the quantum regime, though in the considerably longer term, that evaluation may be subject to reappraisal.

9. Discussion and Limitations

The results of this study highlight the deep interweaving of organizational design and privacy governance within permissionless blockchain ecosystems. The open and distributed nature of these systems complicates the operations involved in simultaneously implementing privacy, accountability, and compliance principles. Many blockchain organizations, from decentralized autonomous organizations (DAOs) to development foundations, still lack a formalised security and privacy management structure. The absence of defined roles for auditing, incident management, and compliance monitoring contributes to the persistence of vulnerabilities, particularly as ecosystems grow and become integrated with external financial or governmental systems.

Effective privacy governance requires rethinking the design of organisations' Secure Development Lifecycle (SDLC). The proposed Web3-oriented SDLC model emphasises measurable controls, automated code auditing, and formal verification pipelines from design to deployment. Integrating privacy engineering practices at every stage of the lifecycle ensures a balance between data privacy and system transparency. Furthermore, explicitly assigning privacy responsibilities, such as data management, smart contract auditing, and federated analytics, can reduce fragmentation and improve accountability in decentralised environments.

From a privacy perspective, the fundamental trade-off between transparency and anonymity remains unresolved. Public ledgers inherently allow for exhaustive verification, but also permit the reconstruction of transaction graphs, deanonymization, and behavioral profiling. These challenges are amplified by the emergence of new paradigms such as cross-chain interoperability, Layer 2 solutions, and AI-powered blockchain analytics, which increase the potential for data coupling, while technologies like zero-knowledge proofs,

zkEVMs, and circle signatures offer technical solutions, organisational frameworks must ensure the consistent implementation and governance of these mechanisms.

Furthermore, privacy protection cannot rely solely on cryptography. Organisational culture, reflected in developer practices, node operating policies, and community standards, plays an important role in ensuring sustainable privacy protection. Open governance models that integrate privacy-by-design principles can serve as a model for reconciling innovation and compliance. Collaborative monitoring mechanisms, such as federated learning for anomaly detection, enable the identification of threats in real time without compromising user privacy.

In short, organizational maturity and privacy protection are complementary and essential pillars for the sustainability of blockchain ecosystems. The proposed framework bridges the gap between theoretical privacy protection guarantees and their practical implementation through structured governance, continuous validation, and adaptive privacy enhancement mechanisms. This alignment between organizational responsibility and privacy engineering is a importantstep towards a secure evolution of permissionless blockchains.

10. Conclusions

Privacy remains one of the most critical issues in permissionless blockchains, while technologies such as zero-knowledge proofs, ring signatures, zkEVM, stealth addresses, and modular frameworks like Hyperledger Fabric have improved privacy protection, they still face trade-offs in performance, scalability, and developer accessibility.

This paper analysed current privacy challenges, examined major vulnerabilities across DeFi platforms, NFTs, and Layer-2 ecosystems, and outlined a practical Secure Development Lifecycle (SDLC) framework for privacy-focused blockchain development. Moving forward, research and engineering efforts should focus on creating balanced solutions that enhance privacy without weakening transparency or network efficiency. A stronger link between academic research, open-source projects, and production systems will be vital to achieving a more private, secure, and sustainable permissionless blockchain environment.

Author Contributions: Conceptualisation, T.B.; methodology, T.B. and R.B.; software, T.B.; validation, A.Y.; formal analysis, R.B.; investigation, T.B.; resources, A.Y.; writing—original draft preparation, T.B.; writing—review and editing, T.B. and R.B.; visualisation, T.B.; supervision, R.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was carried out using facilities partially supported by the Ministry of Science and Higher Education of the Republic of Kazakhstan under grant AP23487613, A. Yazici.

Data Availability Statement: All data used in the paper were taken from the public domain. The scripts and programs used to produce the figures in the paper are openly available at https://github.com/tbayan/permissionless-blockchain-review-artifacts/tree/main (accessed on 8 November 2025).

Acknowledgments: The first author gratefully acknowledges the generous support of the Bolashaq International Scholarship, the international scholarship of the President of the Republic of Kazakhstan, which provided funding during the doctoral studies that enabled this research.

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

Abbreviations

The following abbreviations are used in this manuscript:

RWA	Real World Asset
NFT	Non-Fungible Token
IP	Intellectual Property
SDLC	Secure Development Lifecycle

TTP	Trusted Third Party
ICO	Initial Coin Offering
SPL	Solana Program Library
ENS	Ethereum Name Service
IPFS	InterPlanetary File System
DAO	Decentralised Autonomous Organization
dApps	Decentralised Application
DeFi	Decentralised Finance
EVM	Ethereum Virtual Machine
GDPR	General Data Protection Regulation
KYC	Know Your Customer
L2	Layer 2 Blockchain
ZKP	Zero-Knowledge Proof
TVL	Total Value Locked
MEV	Maximal Extractable Value
RPC	Remote Procedure Call
HLF	Hyperledger Fabric
USDT	US Dollar Tether
USDC	US Dollar Circle
RSA	Rivest–Shamir–Adleman cryptosystem
DH	Diffie–Hellman key exchange
ECC	Elliptic Curve Cryptography

Appendix A. Blockchain Wallet Security Workflow

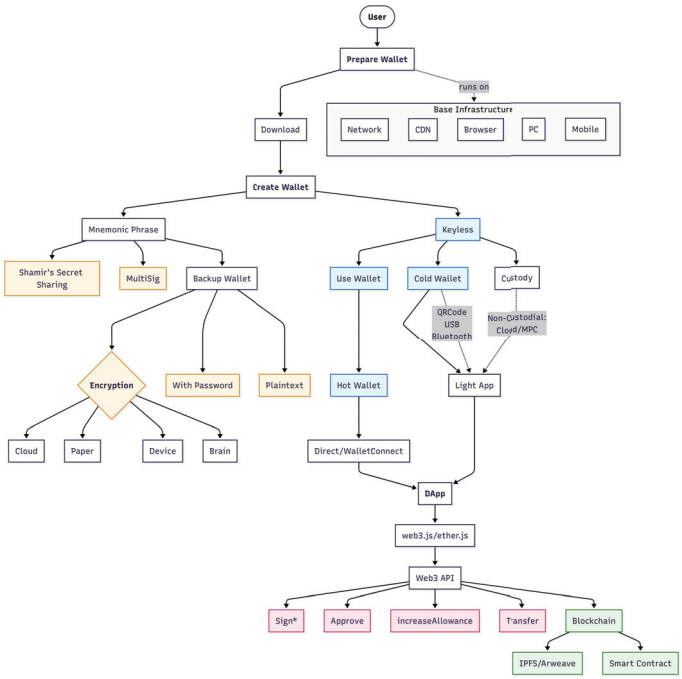


Figure A1. Blockchain wallet security workflow illustrating key user interactions and critical risk touchpoints from wallet creation to transaction execution. Colour coding distinguishes operational layers and highlights potential vulnerabilities such as key leakage, phishing, and unsafe smart contract interaction. Adapted from SlowMist HKU presentation [148].

Appendix B. Unified Terms and Metrics

This appendix standardises the quantitative metrics used across Sections 2–4. Data are sourced from public blockchain analytics platforms (DeFiLlama, TokenTerminal, Dune, GitHub API) as of March 2025 unless otherwise noted.

- Total Value Locked (TVL)—The USD-equivalent value of all assets deposited in smart contracts of a protocol. Calculated by multiplying token balances by the prices of the oracle on the chain. The time series figures use the daily average TVL.
- Active Developers—The number of distinct contributors to GitHub who commit at least one code commit or pull request in a 30-day rolling window. Aggregated at the protocol or ecosystem level.
- Real Users—Unique externally owned accounts (EOAs) that have executed at least one transaction with a smart contract in 7 days. Contract-generated addresses and Sybil clusters are excluded using standard clustering heuristics.
- Transaction Activity—Daily count of confirmed transactions, normalised by active addresses to reduce noise from burst events such as airdrop farming or arbitrage.
- Time Windows—Unless otherwise stated, short-term = 7 days, medium-term = 30 days, long-term = 180 days. All trend analyses are conducted within the medium-term window.

All metrics were cross-verified through reproducible queries using at least two independent APIs. Figure captions specify the data source and snapshot date to enhance transparency. In your Abbreviations section, keep it simple.

References

1. Matsuoka, D.; Lazzarin, E. Estimating the Number of Real Crypto Users. 2024. Available online: https://a16zcrypto.com/posts/article/estimating-crypto-users/
2. Chainalysis. The Chainalysis 2025 Global Adoption Index. 2025. Available online: https://www.chainalysis.com/blog/2025-global-crypto-adoption-index/ (accessed on 11 September 2025).
3. TRM Labs. *2025 Crypto Crime Report: Key Trends That Shaped the Illicit Crypto Market in 2024*; Technical report; TRM Labs: San Francisco, CA, USA, 2025. Available online: https://www.trmlabs.com/reports-and-whitepapers/2025-crypto-crime-report (accessed on 16 November 2025).
4. Feng, Q.; He, D.; Zeadally, S.; Khan, M.; Kumar, N. A Survey on Privacy Protection in Blockchain System. *J. Netw. Comput. Appl.* **2019**, *126*, 45–58. [CrossRef]
5. Peng, L.; Feng, W.; Yan, Z.; Li, Y.; Zhou, X.; Shimizu, S. Privacy Preservation in Permissionless Blockchain: A Survey. *Digit. Commun. Netw.* **2021**, *7*, 295–306. [CrossRef]
6. Henry, R.; Herzberg, A.; Kate, A. Blockchain Access Privacy: Challenges and Directions. *IEEE Secur. Priv.* **2018**, *16*, 38–45. [CrossRef]
7. Electric Capital. *Electric Capital Developer Report 2022: The Next 7 Years of Crypto Developers*; Technical report; Electric Capital: Palo Alto, CA, USA, 2022.
8. Electric Capital. *2024 Crypto Developer Report*; Technical report; Electric Capital: Palo Alto, CA, USA, 2024.
9. GitHub. Crypto Ecosystems: A Taxonomy for Sharing Data Around Crypto, web3, Blockchain, and Decentralized Protocols. Electric-Capital/Crypto-Ecosystems Repository. 2024. Available online: https://github.com/electric-capital/crypto-ecosystems (accessed on 11 September 2025).
10. DeFiLlama. Total Value Locked All Chains. 2025. Available online: https://defillama.com/chains (accessed on 11 September 2025).
11. Etherscan. Ethereum (ETH) Blockchain Explorer. 2025. Available online: https://etherscan.io/chart/tx (accessed on 11 September 2025).
12. BscScan. BNB Smart Chain Daily Transactions Chart. 2025. Available online: https://bscscan.com/chart/tx (accessed on 11 September 2025).
13. Solscan. Solana Analytics. 2025. Available online: https://solscan.io/analytics (accessed on 11 September 2025).
14. Polygonscan. Polygon PoS Chain Daily Transactions Chart. 2025. Available online: https://polygonscan.com/chart/tx (accessed on 11 September 2025).

15. TronScan. TRON Blockchain Explorer. 2025. Available online: https://tronscan.org/ (accessed on 11 September 2025).
16. YCharts. Bitcoin Transactions Per Day. Available online: https://ycharts.com/indicators/bitcoin_transactions_per_day (accessed on 2 October 2025).
17. Near Protocol Explorer. Explore the NEAR Blockchain. 2025. Available online: https://nearblocks.io/ (accessed on 11 September 2025).
18. Sun, X.; Yu, F.R.; Zhang, P.; Sun, Z.; Xie, W.; Peng, X. A Survey on Zero-Knowledge Proof in Blockchain. *IEEE Netw.* **2021**, *35*, 198–205. [CrossRef]
19. Groth, J. On the Size of Pairing-Based Non-Interactive Arguments. In *Advances in Cryptology—EUROCRYPT 2016, Proceedings of the 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, 8–12 May 2016*; Lecture Notes in Computer Science; Fischlin, M., Coron, J.S., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9666, pp. 305–326. [CrossRef]
20. Goodell, B.; Noether, S. Concise Linkable Ring Signatures and Forgery Against Adversarial Keys. *IACR Cryptology ePrint Archive*; Paper 2019/654. 2019. Available online: https://eprint.iacr.org/2019/654 (accessed on 16 November 2025).
21. Wahrstätter, T.; Solomon, M.; Di Francesco, B.; Buterin, V. ERC-5564: Stealth Addresses. 2022. Available online: https://eips.ethereum.org/EIPS/eip-5564 (accessed on 11 September 2025).
22. Halborn Security. *2024 Blockchain Security in Review: Key Lessons Learned*; Technical report; Halborn Security: Miami, FL, USA, 2024.
23. Bayan, T.; Banach, R. Exploring the Privacy Concerns in Permissionless Blockchain Networks and Potential Solutions. In Proceedings of the 2023 IEEE International Conference on Smart Information Systems and Technologies (SIST), Astana, Kazakhstan, 4–6 May 2023; pp. 567–572.
24. Banach, R. Punishment not Reward Blockchain Architecture for a Privacy-Preserving Permissioned Network. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e5749. [CrossRef]
25. Wang, Y.; Yang, J.; Li, T.; Zhu, F.; Zhou, X. Anti-Dust: A Method for Identifying and Preventing Blockchain's Dust Attacks. In Proceedings of the 2018 International Conference on Information Systems and Computer Aided Education (ICISCAE), Changchun, China, 28–30 September 2018; pp. 274–280.
26. Saad, M.; Spaulding, J.; Njilla, L.; Kamhoua, C.; Shetty, S.; Nyang, D.; Mohaisen, A. Exploring the Attack Surface of Blockchain: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1977–2008. [CrossRef]
27. Jones, M.; Johnson, M.; Shervey, M.; Dudley, J.; Zimmerman, N. Privacy-Preserving Methods for Feature Engineering Using Blockchain: Review, Evaluation, and Proof of Concept. *J. Med. Internet Res.* **2019**, *21*, e13600. [CrossRef]
28. Stamatellis, C.; Papadopoulos, P.; Pitropakis, N.; Katsikas, S.; Buchanan, W. A Privacy-Preserving Healthcare Framework Using Hyperledger Fabric. *Sensors* **2020**, *20*, 6587. [CrossRef] [PubMed]
29. Bernal Bernabe, J.; Canovas, J.; Hernandez-Ramos, J.; Torres Moreno, R.; Skarmeta, A. Privacy-Preserving Solutions for Blockchain: Review and Challenges. *IEEE Access* **2019**, *7*, 164908–164940. [CrossRef]
30. Bayan, T.; Banach, R. A Privacy-Preserving DAO Model Using NFT Authentication for the Punishment not Reward Blockchain Architecture. *arXiv* **2024**, arXiv:2405.13156. [CrossRef]
31. Bayan, T. Privacy in Blockchain: Addressing Security Challenges, Trends, Smart Contract Innovations, and a Prototype on PnR Architecture. Ph.D. Thesis, The University of Manchester, Manchester, UK, 2025.
32. Wang, D.; Zhao, J.; Wang, Y. A Survey on Privacy Protection of Blockchain: The Technology and Application. *IEEE Access* **2020**, *8*, 108766–108781. [CrossRef]
33. Arora, D.; Gautham, S.; Gupta, H.; Bhushan, B. Blockchain-based Security Solutions to Preserve Data Privacy And Integrity. In Proceedings of the 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 18–19 October 2019; pp. 468–472.
34. Junejo, A.; Hashmani, M.; Alabdulatif, A. A Survey on Privacy Vulnerabilities in Permissionless Blockchains. *Int. J. Adv. Comput. Sci. Appl.* 2020, *11*, 15–24. [CrossRef]
35. Huynh, T.; Nguyen, T.; Tan, H. A Survey on Security and Privacy Issues of Blockchain Technology. In Proceedings of the 2019 International Conference on System Science and Engineering (ICSSE), Dong Hoi, Vietnam, 20–21 July 2019; pp. 362–367.
36. Yassein, M.; Shatnawi, F.; Rawashdeh, S.; Mardin, W. Blockchain Technology: Characteristics, Security and Privacy; Issues and Solutions. In Proceedings of the 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 3–7 November 2019; pp. 1–8.
37. Ledger. Top Solana Memecoins in 2025. 2025. Available online: https://www.ledger.com/academy/topics/crypto/top-solana-memecoins-in-2025 (accessed on 6 October 2025).
38. ChainCatcher. Token Deflation Experiment: Hyperliquid and Pump.fun's Apple-Style Gamble. 2025. Available online: https://www.chaincatcher.com/en/article/2210259 (accessed on 6 October 2025).

39. CryptoDnes. Best Solana Meme Coins to Watch in 2025: A Full Guide. 2025. Available online: https://cryptodnes.bg/en/cryptocurrency/best-solana-meme-coins/ (accessed on 6 October 2025).
40. B2BinPay. Solana Meme Coins How to Create & Best Ones to Follow in 2025. 2025. Available online: https://b2binpay.com/en/news/10-best-solana-meme-coins-to-watch-in-2024 (accessed on 6 October 2025).
41. Meiklejohn, S.; Pomarole, M.; Jordan, G.; Levchenko, K.; McCoy, D.; Voelker, G.M.; Savage, S. A Fistful of Bitcoins: Characterizing Payments Among Men with No Names. In Proceedings of the 2013 Internet Measurement Conference, Barcelona, Spain, 23–25 October 2013; ACM: New York, NY, USA, 2013; pp. 127–140. [CrossRef]
42. InvestaX. *2024: The Year of Institutional Real World Asset Tokenization*; Technical Report; InvestaX: Singapore, 2024.
43. CoinGecko. What Are Real World Assets (RWA)? A Complete Guide for 2025. 2025. Available online: https://www.coingecko.com/learn/what-are-real-world-assets-exploring-rwa-protocols (accessed on 6 October 2025).
44. DeFiLlama. Real World Assets (RWA) Protocols—Total Value Locked (TVL). 2025. Available online: https://defillama.com/protocols/rwa (accessed on 11 September 2025).
45. Caldarelli, G. Understanding the Blockchain Oracle Problem: A Call for Action. *Information* **2020**, *11*, 509. [CrossRef]
46. Chainlink. Real-World Assets (RWAs) Explained. 2025. Available online: https://chain.link/education-hub/real-world-assets-rwas-explained (accessed on 6 October 2025).
47. NEFTURE Security. Understanding Crypto Perpetual Futures and the Hyperliquid Craze. 2024. Available online: https://medium.com/coinmonks/understanding-crypto-perpetual-futures-and-the-hyperliquid-craze-7d1c8b413444 (accessed on 6 October 2025).
48. CoinLedger. What Is Hyperliquid? 2025. Available online: https://coinledger.io/learn/what-is-hyperliquid (accessed on 6 October 2025).
49. DataWallet. Best Decentralized Perpetuals Exchanges in 2025. 2025. Available online: https://www.datawallet.com/crypto/best-decentralized-perpetuals-exchanges (accessed on 6 October 2025).
50. Chen, E.; Ma, M.; Nie, Z. Perpetual Future Contracts in Centralized and Decentralized Exchanges: Mechanism and Traders' Behavior. *Electron. Mark.* **2024**, *34*, 35. [CrossRef]
51. Chen, E.; Ma, M.; Nie, Z. Exploring the Impact: How Decentralized Exchange Designs Shape Traders' Behavior on Perpetual Future Contracts. *arXiv* **2024**, arXiv:2402.03953. [CrossRef]
52. DeFiLlama. Perpetuals (Perps)—Perpetual Volume Dashboard. 2025. Available online: https://defillama.com/perps (accessed on 11 September 2025).
53. Ruan, Q.; Streltsov, A. Perpetual Futures Contracts and Cryptocurrency Market Quality. *SSRN Electron. J.* **2022**. [CrossRef]
54. CCN. From $0.08 to $2.42: How Aster Became DeFi's Hottest (and Riskiest) Perp DEX. 2025. Available online: https://www.ccn.com/education/crypto/aster-defi-perpetual-futures-vs-hyperliquid-explained/ (accessed on 6 October 2025).
55. Cointelegraph. Kazakhstan Launches Alem Crypto Fund with First Investment in BNB. 2025. Available online: https://cointelegraph.com/news/kazakhstan-state-backed-crypto-fund-bnb-binance (accessed on 6 October 2025).
56. CoinMarketCap. Kazakhstan Launches State Crypto Fund with BNB. 2025. Available online: https://coinmarketcap.com/academy/article/kazakhstan-launches-state-crypto-fund-with-bnb (accessed on 6 October 2025).
57. BeInCrypto. Kazakhstan Launches State-Backed Crypto Fund. 2025. Available online: https://beincrypto.com/kazakhstan-launches-state-backed-crypto-fund/ (accessed on 6 October 2025).
58. The Crypto Basic. Kazakhstan Launches First Crypto Reserve with BNB. 2025. Available online: https://thecryptobasic.com/2025/09/30/kazakhstan-launches-first-crypto-reserve-with-bnb/ (accessed on 6 October 2025).
59. Wikipedia. Polymarket. 2025. Available online: https://en.wikipedia.org/wiki/Polymarket (accessed on 6 October 2025).
60. The New Political. Polymarket and the Rising Popularity of Political Prediction Markets. 2024. Available online: https://www.thenewpolitical.com/news/dvgt9l02yfs3x9vzu86xyus5t047tt (accessed on 6 October 2025).
61. University of Cincinnati. Election Results Show Potential of Prediction Markets, Blockchain. 2024. Available online: https://www.uc.edu/news/articles/2024/12/election-results-show-potential-of-prediction-markets.html (accessed on 6 October 2025).
62. Fortune. Exclusive: Election Betting Site Polymarket Gives Trump a 67% Chance of Winning But Is Rife with Fake 'Wash' Trading, Researchers Say. 2024. Available online: https://fortune.com/crypto/2024/10/30/polymarket-trump-election-crypto-wash-trading-researchers/ (accessed on 6 October 2025).
63. Qu, Y.; Uddin, M.; Gan, C.; Xiang, Y.; Gao, L.; Yearwood, J. Blockchain-Enabled Federated Learning: A Survey. *ACM Comput. Surv.* **2022**, *55*, 70. [CrossRef]
64. Nguyen, D.; Ding, M.; Pathirana, P.; Seneviratne, A.; Li, J.; Poor, H. Federated Learning Meets Blockchain in Edge Computing: Opportunities and Challenges. *IEEE Internet Things J.* **2021**, *8*, 12806–12825. [CrossRef]

65. Shayan, M.; Fung, C.; Yoon, C.; Beschastnikh, I. Biscotti: A Blockchain System for Private and Secure Federated Learning. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *32*, 1513–1525. [CrossRef]
66. Ullah, I.; Deng, X.; Pei, X.; Mushtaq, H.; Uzair, M.; Qayyum, S. A Blockchain-Based Federated Learning Framework Against Poisoning Attacks in the Internet of Vehicles. *Comput. Netw.* **2025**, *272*, 111705. [CrossRef]
67. Bao, Y.; Sun, J.; Cheng, X.; Qiu, W.; Nie, L. Blockchain-Based Privacy-Preserving Alternative Credit Data Sharing. *IEEE Trans. Comput. Soc. Syst.* 2025, *early access*. [CrossRef]
68. Liang, W.; Ji, N. Privacy Challenges of IoT-Based Blockchain: A Systematic Review. *Clust. Comput.* **2022**, *25*, 2203–2221. [CrossRef]
69. Yazici, A.; Zhumabekova, D.; Nurakhmetova, A.; Yergaliyev, Z.; Yatbaz, H.; Makisheva, Z.; Lewis, M.; Ever, E. A Smart E-Health Framework for Monitoring the Health of the Elderly and Disabled. *Internet Things* **2023**, *24*, 100971. [CrossRef]
70. Baltabay, M.; Yazici, A.; Sterling, M.; Ever, E. Designing Efficient and Lightweight Deep Learning Models for Healthcare Analysis. *Neural Process. Lett.* **2023**, *55*, 6947–6977. [CrossRef]
71. Yatbaz, H.; Ever, E.; Yazici, A. Activity Recognition and Anomaly Detection in E-Health Applications Using Color-Coded Representation and Lightweight CNN Architectures. *IEEE Sens. J.* **2021**, *21*, 14191–14202. [CrossRef]
72. Baimukhanov, S.; Ali, H.; Yazici, A. Enhancing ML-Based Anomaly Detection in Data Management for Security Through Integration of IoT, Cloud, and Edge Computing. *Expert Syst. Appl.* **2025**, *293*, 128700. [CrossRef]
73. Akhmetov, A.; Latif, Z.; Tyler, B.; Yazici, A. Enhancing Healthcare Data Privacy and Interoperability with Federated Learning. *PeerJ Comput. Sci.* **2025**, *11*, e2870. [CrossRef] [PubMed]
74. Bayan, T.; Galy, M.M.; Ziyashev, A.A. Image Search Module in a Digital Facilitator Network System: Architecture, Algorithms, and Implementation. *Orleu. Bull. Contin. Educ.* **2025**, 83–93. [CrossRef]
75. Bayan, T.; Nurbekova, Z.; Galy, M.M.; Baigusheva, K. Enhanced Multimodal Diagram Search for Professional Facilitator Networks: A Web-Based Educational Platform. In Proceedings of the 2025 10th International Conference on Computer Science and Engineering (UBMK), Istanbul, Turkiye, 17–21 September 2025; pp. 1252–1257. [CrossRef]
76. Wash, R.; Rader, E.; Berman, R.; Wellmer, Z. Understanding Password Choices: How Frequently Entered Passwords Are Re-used Across Websites. In Proceedings of the Symposium on Usable Privacy and Security (SOUPS), Boston, MA, USA, 10–11 August 2020; pp. 175–188.
77. Hankerson, D.; Menezes, A.J.; Vanstone, S. *Guide to Elliptic Curve Cryptography*; Springer: New York, NY, USA, 2004.
78. Chainalysis. Crypto Hacking Revenue Hits All-Time High as Scam Revenue Declines. 2024. Available online: https://www.chainalysis.com/blog/crypto-hacking-stolen-funds-2025/ (accessed on 11 September 2025).
79. LastPass. Notice of Recent Security Incident. LastPass Blog. 2023. Available online: https://blog.lastpass.com/2023/03/security-incident-update-recommended-actions/ (accessed on 7 October 2025).
80. CCN. Crypto Hacks 2025: Full List of Scams, Exchange Exploits & DeFi Vulnerabilities. 2025. Available online: https://www.ccn.com/education/crypto/crypto-hacks-exploits-full-list-scams-vulnerabilities/ (accessed on 11 September 2025).
81. Yaga, D.; Mell, P.; Roby, N.; Scarfone, K. *Blockchain Technology Overview*; NIST Internal Report-8202; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2019.
82. Shao, Y. Research on Blockchain Transaction Privacy Protection Methods Based on Deep Learning. *Future Internet* **2024**, *16*, 113. [CrossRef]
83. Xia, P.; Zhang, H.; Cao, B.; Wang, H.; Yu, J.; Wang, B. Ethereum Name Service: The Good, the Bad, and the Ugly. *arXiv* **2021**, arXiv:2104.05185. [CrossRef]
84. TokenInsight. Unmasking Metamask—Is Web3 Really Decentralized And Private? 2022. Available online: https://tokeninsight.medium.com/unmasking-metamask-is-web3-really-decentralized-and-private-307260d720be (accessed on 11 September 2025).
85. Alchemy. List of 43 RPC Node Providers. 2025. Available online: https://www.alchemy.com/dapps/best/rpc-node-providers (accessed on 11 September 2025).
86. Crypto Briefing. Infura, Alchemy Block Tornado Cash Following Treasury Ban. 2022. Available online: https://cryptobriefing.com/infura-alchemy-block-tornado-cash-following-treasury-ban/ (accessed on 11 September 2025).
87. Bitquery. Understanding Different MEV Attacks: Frontrunning, Backrunning and Other Attacks. 2024. Available online: https://bitquery.io/blog/different-mev-attacks (accessed on 11 September 2025).
88. CoinMarketCap. Frontrunners and MEV Explained: How to Beat the Bots. 2023. Available online: https://coinmarketcap.com/academy/article/frontrunners-and-mev-explained-how-to-beat-the-bots (accessed on 11 September 2025).
89. Blockworks. Ethereum Validator Goes Rogue, Frontruns MEV Bots for $25 M. 2023. Available online: https://blockworks.co/news/validator-frontruns-mev-bots (accessed on 11 September 2025).
90. Gate.io. Is Your Wallet Safe? How Hackers Exploit Permit, Uniswap Permit2, and Signatures for Phishing. 2024. Available online: https://www.gate.io/learn/articles/is-your-wallet-safe-how-hackers-exploit-permit-uniswap-permit2-and-signatures-for-phishing/4197 (accessed on 11 September 2025).

91. Halborn Security. *2025 Blockchain Security Forecast: Top Threats for the Year Ahead*; Technical report; Halborn Security: Miami, FL, USA, 2025.
92. Decrypt. Pepe Holder Loses $1.4 Million in Uniswap Permit2 Phishing Attack. 2024. Available online: https://decrypt.co/286076/pepe-uniswap-permit2-phishing-attack (accessed on 11 September 2025).
93. Johnson, D.; Menezes, A.; Vanstone, S. The Elliptic Curve Digital Signature Algorithm (ECDSA). *Int. J. Inf. Secur.* **2001**, *1*, 36–63. [CrossRef]
94. Atzei, N.; Bartoletti, M.; Cimoli, T. A Survey of Attacks on Ethereum Smart Contracts (SoK). In *Principles of Security and Trust, Proceedings of the 6th International Conference, POST 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, 24–25 April 2017*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2017; Volume 10204, pp. 164–186. [CrossRef]
95. OWASP Foundation. OWASP Smart Contract Top 10. 2025. Available online: https://owasp.org/www-project-smart-contract-top-10/ (accessed on 11 September 2025).
96. OpenZeppelin. OpenZeppelin Contracts: A Library for Secure Smart Contract Development. GitHub Repository. 2025. Available online: https://github.com/OpenZeppelin/openzeppelin-contracts (accessed on 11 September 2025).
97. Hejazi, N.; Lashkari, A. A Comprehensive Survey of Smart Contracts Vulnerability Detection Tools: Techniques and Methodologies. *J. Netw. Comput. Appl.* **2025**, *237*, 104142. [CrossRef]
98. Feng, X.; Li, Q.; Wang, H.; Sun, L. A Survey on Smart Contract Vulnerabilities: Data Sources, Detection and Repair. *J. Syst. Softw.* **2024**, *209*, 111919. [CrossRef]
99. Iuliano, G.; Di Nucci, D. Smart Contract Vulnerabilities, Tools, and Benchmarks: An Updated Systematic Literature Review. *arXiv* **2024**, arXiv:2412.01491. [CrossRef]
100. Gennaro, R.; Goldfeder, S.; Narayanan, A. Threshold-Optimal DSA/ECDSA Signatures and an Application to Bitcoin Wallet Security. In *Applied Cryptography and Network Security, Proceedings of the 14th International Conference, ACNS 2016, Guildford, UK, 19–22 June 2016*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9696, pp. 156–174. [CrossRef]
101. LF Decentralized Trust. A Year of Milestones: Let's Take a Look at 2024. 2024. Available online: https://www.lfdecentralizedtrust.org/blog/a-year-of-milestones-lets-take-a-look-at-2024 (accessed on 30 September 2025).
102. Yu, J.; Ge, L.; Wu, M. Proposal Distribution Optimization for Endorsement Strategy in Hyperledger Fabric. *J. Supercomput.* **2024**, *80*, 15038–15065. [CrossRef]
103. Al-Sumaidaee, G.; Alkhudary, R.; Zilic, Z.; Swidan, A. Performance Analysis of a Private Blockchain Network Built on Hyperledger Fabric for Healthcare. *Inf. Process. Manag.* **2023**, *60*, 103160. [CrossRef]
104. Ke, Z.; Park, N. Performance Modeling and Analysis of Hyperledger Fabric. *Cluster Comput.* **2023**, *26*, 2681–2699. [CrossRef]
105. Nedaković, A.; Hasselgren, A.; Kralevska, K.; Gligoroski, D. Hyperledger Fabric Platform for Healthcare Trust Relations—Proof-of-Concept. *Blockchain Res. Appl.* **2023**, *4*, 100156. [CrossRef]
106. Abang, J.; Takruri, H.; Al-Zaidi, R.; Al-Khalidi, M. Latency Performance Modelling in Hyperledger Fabric Blockchain: Challenges and Directions with an IoT Perspective. *Internet Things* **2024**, *26*, 101217. [CrossRef]
107. Sharma, P.; Jindal, R.; Borah, M. Blockchain-Based Distributed Application for Multimedia System Using Hyperledger Fabric. *Multimed. Tools Appl.* **2024**, *83*, 2473–2499. [CrossRef]
108. Monero Research Lab. View Tags and How They Reduce Wallet Synchronization Times. Monero Research Lab Technical Note. 2022. Available online: https://github.com/monero-project/research-lab/issues/73 (accessed on 16 November 2025).
109. Eberhardt, J.; Tai, S. ZoKrates-Scalable Privacy-Preserving Off-Chain Computations. In Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 30 July–3 August 2018; pp. 111–118.
110. Huang, R.; Chen, J.; Wang, Y.; Bi, T.; Nie, L.; Zheng, Z. An Overview of Web3 Technology: Infrastructure, Applications, and Popularity. *Blockchain Res. Appl.* **2024**, *5*, 100173. [CrossRef]
111. Wang, S.; Yuan, Y.; Wang, X.; Li, J.; Qin, R.; Wang, F. Exploring Web3 From the View of Blockchain. *arXiv* **2022**, arXiv:2206.08821. [CrossRef]
112. Liu, W.; Cao, B.; Peng, M. Web3 Technologies: Challenges and Opportunities. *IEEE Netw.* **2024**, *38*, 187–193. [CrossRef]
113. Allen, D.; Potts, J. Web3 Toolkits: A User Innovation Theory of Crypto Development. *J. Open Innov. Technol. Mark. Complex.* **2023**, *9*, 100050. [CrossRef]
114. SlowMist. Web3 Project Security Practice Requirements. 2024. Available online: https://github.com/slowmist/Web3-Project-Security-Practice-Requirements (accessed on 2 October 2025).

115. Trail of Bits. Echidna: A Fast Smart Contract Fuzzer. GitHub Repository. 2023. Available online: https://github.com/crytic/echidna (accessed on 11 September 2025).
116. Paradigm. Foundry: A Blazing Fast, Portable and Modular Toolkit for Ethereum Application Development. GitHub Repository. 2024. Available online: https://github.com/foundry-rs/foundry (accessed on 11 September 2025).
117. Amazon Web Services. Git-Secrets: Prevents You From Committing Secrets and Credentials Into Git Repositories. GitHub Repository. 2023. Available online: https://github.com/awslabs/git-secrets (accessed on 11 September 2025).
118. Nomic Foundation. Hardhat: Ethereum Development Environment for Professionals. GitHub Repository. 2024. Available online: https://github.com/NomicFoundation/hardhat (accessed on 11 September 2025).
119. Project Management Institute. *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, 7th ed.; Project Management Institute: Newtown Square, PA, USA, 2021.
120. Bayan, T.; Banach, R.; Nurbekov, A.; Galy, M.M.; Sabyrbayev, A.; Nurbekova, Z. Blockchain-enhanced Integrity Verification in Educational Content Assessment Platform: A Lightweight and Cost-Efficient Approach. *arXiv* **2024**, arXiv:2409.19828.
121. Filipčić, S. Web3 & DAOs: An Overview of the Development and Possibilities for the Implementation in Research and Education. In Proceedings of the 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 23–27 May 2022; pp. 1278–1283. [CrossRef]
122. Platzer, A.; Rozier, A.Y.; Pradella, M.; Rossi, M. (Eds.) *Formal Methods 2024, LNCS*; Springer: Berlin/Heidelberg, Germany, 2025; Volume 14933–14934.
123. Brain, M.; Polgreen, E. A Pyramid of (Formal) Software Verification. In *Formal Methods, Proceedings of the 26th International Symposium, FM 2024, Milan, Italy, 9–13 September 2024*; Springer: Cham, Switzerland, 2025; Volume 14933–14934.
124. Hildenbrandt, E.; Saxena, M.; Rodrigues, N.; Zhu, X.; Daian, P.; Guth, D.; Moore, B.; Park, D.; Zhang, Y.; Stefanescu, A.; et al. KEVM: A Complete Formal Semantics of the Ethereum Virtual Machine. In Proceedings of the 2018 IEEE 31st Computer Security Foundations Symposium (CSF), Oxford, UK, 9–12 July 2018; pp. 204–217.
125. Park, D.; Zhang, Y.; Saxena, M.; Daian, P.; Rosu, G. A Formal Verification Tool for Ethereum VM Bytecode. In Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE-18, Lake Buena Vista, FL, USA, 4–9 November 2018; ACM: New York, NY, USA, 2018; pp. 912–915.
126. Hirai, Y. Defining the Ethereum Virtual Machine for Interactive Theorem Provers. In *Financial Cryptography and Data Security, Proceedings of the FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, 7 April 2017*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2017; pp. 520–535.
127. Duan, Z.; Mao, H.; Chen, Z.; Bai, X.; Hu, K.; Talpin, J.P. Formal Modeling and Verification of Blockchain System. In Proceedings of the 10th International Conference on Computer Modeling and Simulation, ICCMS-18, Sydney, Australia, 8–10 January 2018; ACM: New York, NY, USA, 2018; pp. 231–235.
128. Al-awamy, A.; Al-shaibany, N.; Sikora, A.; Welte, D. Hybrid Consensus Mechanisms in Blockchain: A Comprehensive Review. *Int. J. Intel. Sys.* **2025**, *2025*, 5821997. [CrossRef]
129. Yang, Z.; Lei, H. A Formal Process Virtual Machine for EOS-Based Smart Contract Security Verification. In *Intelligent Computing and Block Chain, Proceedings of the First BenchCouncil International Federated Conferences, FICC 2020, Qingdao, China, 30 October–3 November 2020*; Springer: Singapore, 2020; pp. 253–263.
130. Murray, Y.; Anisi, D. Survey of Formal Verification Methods for Smart Contracts on Blockchain. In Proceedings of the 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Canary Islands, Spain, 24–26 June 2019; pp. 1–6.
131. Yang, Z.; Lei, H. Formal Process Virtual Machine for Smart Contracts Verification. *Int. J. Perf. Eng.* **2018**, *14*, 1726–1734. [CrossRef]
132. Bhargavan, K.; Delignat-Lavaud, A.; Fournet, C.; Gollamudi, A.; Gonthier, G.; Kobeissi, N.; Kulatova, N.; Rastogi, A.; Sibut-Pinote, T.; Swamy, N.; et al. Formal Verification of Smart Contracts: Short Paper. In Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security, Vienna, Austria, 24 October 2016; ACM: New York, NY, USA, 2016; pp. 91–96.
133. Abdellatif, T.; Brousmiche, K.-L. Formal Verification of Smart Contracts Based on Users and Blockchain Behaviors Models. In Proceedings of the 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 26–28 February 2018; pp. 1–5.
134. Liu, Z.; Liu, J. Formal Verification of Blockchain Smart Contract Based on Colored Petri Net Models. In Proceedings of the 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), Milwaukee, WI, USA, 15–19 July 2019; Volume 2, pp. 555–560.
135. Kulik, T.; Dongol, B.; Gorm Larsen, P.; Macedo, H.; Schneider, S.; Tran-Jorgensen, P.; Woodcock, J. A Survey of Practical Formal Methods for Security. *Form. Asp. Comp.* **2022**, *34*, 5. [CrossRef]

136. Li, K.; Gu, R.; Xu, J.; Chen, Z.; Wu, S.; Zhou, Y.; Zhang, M.; Luo, X.; Tang, Y.; Li, Y.; et al. Security Analysis and Formal Verification on Blockchain and its Applications. *Found. Trends Priv. Sec.* **2025**, *8*, 1–121. [CrossRef]
137. Shor, P. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994.
138. Shor, P. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *Siam J. Comput.* **1997**, *26*, 1484–1509. [CrossRef]
139. What is Post-Quantum Cryptography? 2025. Available online: https://www.nist.gov/cybersecurity/what-post-quantum-cryptography (accessed on 11 September 2025).
140. Bernstein, D.; Buchmann, J.; Dahmen, E. (Eds.) *Post-Quantum Cryptography*; Springer: Berlin/Heidelberg, Germany, 2009.
141. Module-Lattice-Based Key-Encapsulation Mechanism Standard. 2024. Available online: https://csrc.nist.gov/pubs/fips/203/final (accessed on 11 September 2025).
142. Module-Lattice-Based Digital Signature Standard. 2024. Available online: https://csrc.nist.gov/pubs/fips/204/final (accessed on 11 September 2025).
143. Stateless Hash-Based Digital Signature Standard. 2024. Available online: https://csrc.nist.gov/pubs/fips/205/final (accessed on 11 September 2025).
144. Kearney, J.; Perez-Delgado, C. Vulnerability of Blockchain Technologies to Quantum Attacks. *Array* **2021**, *10*, 100065. [CrossRef]
145. Kulkarni, A.; Jain, S.; Kumar, A. Quantum Computing and Quantum Blockchain: Recent Advancements, Analysis and Future Directions. In *Quantum and Blockchain for Modern Computing Systems: Vision and Advancements: Quantum and Blockchain Technologies: Current Trends and Challenges*; Springer: Cham, Switzerland, 2022; pp. 311–339.
146. Gandhi, M.; Mulay, C.; Durai, K.; Murali, G.; Masood, J.; Vijayarajan, V.; Gautam, K.; Chakravarthy, N.; Kumar, S.; Agarwal, S.; et al. Quantum blockchain: Trends, technologies, and future directions. *IET Quantum Commun.* **2024**, *5*, 516–542. [CrossRef]
147. Yang, Z.; Alfauri, H.; Farkiani, B.; Jain, R.; Pietro, R.D.; Erbad, A. A Survey and Comparison of Post-Quantum and Quantum Blockchains. *IEEE Commun. Surv. Tutorials* **2024**, *26*, 967–1002. [CrossRef]
148. SlowMist. SlowMist Founder Cos Shares at HKU: Blockchain Security—Offense, Defense, and Practices. 2025. Available online: https://slowmist.medium.com/slowmist-founder-cos-shares-at-hku-blockchain-security-offense-defense-and-practices-ac5970c8fbd8 (accessed on 30 September 2025).

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

future internet

MDPI

Article

Fraud Detection in Cryptocurrency Networks—An Exploration Using Anomaly Detection and Heterogeneous Graph Transformers

Víctor Pérez-Cano and Francisco Jurado *

Department of Computer Engineering, Universidad Autónoma de Madrid, 28049 Madrid, Spain; victor.perezcano@estudiante.uam.es
* Correspondence: francisco.jurado@uam.es

Abstract: Blockchains are the backbone behind cryptocurrency networks, which have developed rapidly in the last two decades. However, this growth has brought several challenges due to the features of these networks, specifically anonymity and decentralization. One of these challenges is the fight against fraudulent activities performed in these networks, which, among other things, involve financial schemes, phishing attacks or money laundering. This article will address the problem of identifying fraud cases among a large set of transactions extracted from the Bitcoin network. More specifically, our study's goal was to find reliable techniques to label Bitcoin transactions, taking into account their features. The approach followed involved two kinds of Machine Learning methods. On the one hand, anomaly detection algorithms were applied to determine whether fraudulent activities tend to show anomalous behaviour without resorting to manually obtained labels. On the other hand, Heterogeneous Graph Transformers were used to leverage the heterogeneous relational nature of the cryptocurrency information. As a result, the article will provide reasonable conclusions to acknowledge that unsupervised approaches can be useful for fraud detection on blockchain networks. Furthermore, the effectiveness of supervised graph methods was revalidated, emphasizing the importance of data heterogeneity.

Keywords: blockchain; cryptocurrency; fraudulent activity detection; machine learning

1. Introduction

Cryptocurrency networks such as Bitcoin or Ethereum are extensively used due to their features and differences from traditional money. Although these networks are considerably robust from the perspective of a distributed system thanks to being supported by the blockchain algorithm, cryptocurrencies face a challenge: the commission of fraudulent activities in these networks, mainly because of the lack of a regulatory entity [1,2].

Significant progress has already been made in the context of fraud detection, especially featuring the use of Machine Learning (ML) methods that can process large amounts of transaction data produced by cryptocurrency networks and use them to make predictions about their illicitness. To cite a few of the most recent works, we can highlight the use of traditional ML approaches like Naive Bayes, SVMs, Logistic Regression, Gradient Boosting, Ada-Boost, Random Forests and other Ensemble Learning models [3–8], but also Federation Learning [9], neural network approaches [10–12], anomaly detection [13] and more recent graph-based approaches like those that consider the relational nature of transactional data, like the Hybrid Graph Neural Network [14], graphs based on a Transformer Network [15], the subgraph-based contrastive learning algorithm for heterogeneous graphs [16] and graph anomaly detection [17].

Creating better and bigger datasets and developing more sophisticated deep learning models opens up new possibilities in the context of cryptocurrency fraud detection. However, all these supervised learning methods share a common problem: the need for a large enough labelled dataset to train them. From another perspective, in the absence of a manually labelled dataset, using unsupervised learning methods that can extract information about the data is another research field that, as far as we know, has not been extensively explored for the case of cryptocurrency fraud detection. In this context, this work aimed to revalidate the effectiveness of supervised learning methods on a representative Bitcoin transaction dataset by testing the Heterogeneous Graph Transformer (HGT) [18], a novel framework aimed at leveraging relational data with a heterogeneous nature. In addition, this study also explored the possibilities of different unsupervised learning methods based on anomaly detection through the use of k-means clustering and several feature-based outlier detection algorithms, such as the Local Outlier Factor (LOF) [19]. The obtained results revealed that the performance achieved with non-supervised methods was still far from that of one of the supervised approaches, even though they represent a considerable alternative due to not needing a manually labelled dataset.

Hence, the main contributions of our research can be summarized in the following points: (1) we provide a comprehensive evaluation of unsupervised and supervised methods using real-world datasets, discussing their limitations and potential improvements; (2) explore the potential of feature-based anomaly detection methods in the context of cryptocurrency networks and, as a result, propose a hybrid approach combining structural and intrinsic data features; and (3) put to the test a novel graph framework that leverages the heterogeneous nature of transactional data.

The structure of the rest of this article will be as follows: Section 2 will introduce the background on cryptocurrency fraud and describe the current literature about the topic. Then, Section 3 will present the related works. Later, Section 4 will formally describe the problem, the datasets and the methodology. Following this, Section 5 will detail the different experiments carried out, grouped into four categories: exploratory data analysis, anomaly detection, clustering and graph deep learning. Finally, Section 6 will draw conclusions, summarize the results and provide suggestions for future works.

2. Background on Cryptocurrency Fraud

In brief, cryptocurrency technologies are supported by the blockchain algorithm, which permits the creation of a distributed ledger. This ledger is structured as a sequence of blocks that are cryptographically linked, ensuring that the information stored on them remains immutable, as shown in Figure 1. Each block in the chain contains a set of transactions (tx_n), a timestamp and a cryptographic hash of the previous block ($block_{i-1}$), creating a secure and chronological record. This structure not only protects the integrity of the data but also facilitates trust among users, as they can independently verify transactions without relying on a central authority.

The blockchain operates on a decentralized network, meaning that no single entity has control over the entire system. This decentralization enhances security and transparency, as every participant in the network has access to the same information, making it exceedingly difficult for any malicious actor to alter the data without consensus from the majority.

However, the lack of a regulatory entity and the nature of these distributed systems, which involve a range of properties such as anonymity and decentralization, make these networks an ideal context for cybercriminals to act in.

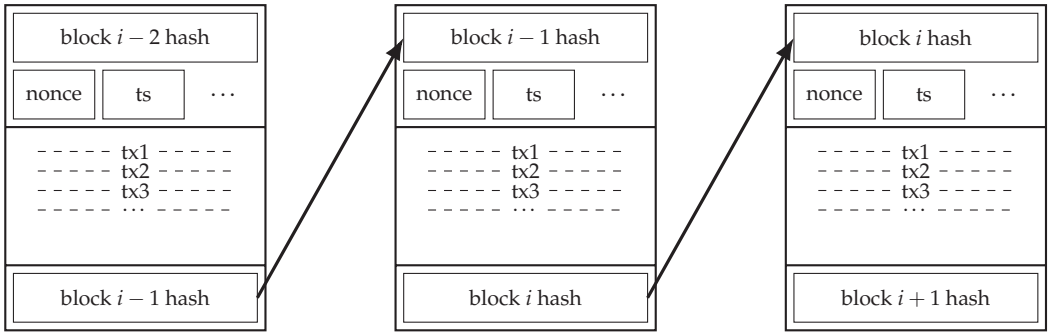


Figure 1. Example blockchain sequence.

According to recent studies conducted by the European Parliament [20] and Europol [21], some of the most prevalent types of fraud occurring within cryptocurrency networks include the following:

- Financial schemes. There are a variety of types of fraud that can be categorized in this group. Some examples include the following:
  - Pump and Dump. This type of fraud consists of a market manipulation technique involving speculation with a cryptocurrency derivative, in which the fraudster "pumps" up the price of the product with the intention of eventually selling all their holdings, consequently "dumping" the price of the derivative.
  - Ponzi Scheme. This is a pyramid scheme in which earlier investors are rewarded with the funds of the newer ones, typically attracted by the potential growth promised by the fraudsters.
  - Rug Pulls. This is a type of fraud in which victims are encouraged to invest in a promising Non-Fungible Token (NFT) or other token-like project, which is then abandoned by the developers, who quickly disappear, leaving the victims with useless assets.
- Phishing attacks. This is a type of fraud where the fraudster fakes their identity, for example, by impersonating a prestigious crypto-investor or creating a fake landing page for a well-known exchange portal.
- As a means of payment. Although this does not involve fraud by itself, cryptocurrencies tend to be the way to go for purchasing illegal goods or services on the dark net, given the lack of oversight and control of state authorities on this part of the web. In this manner, cryptocurrencies are used in markets associated with drugs, arms, sexual abuse or hitman services. Additionally, cryptocurrencies are often requested as payment for multiple kinds of cyberattacks, such as those using ransomware, due to their complicated traceability. As shown in the next section, remarkable progress has been achieved on this specific topic.
- Money laundering. Although this may not be considered fraud, it typically follows some criminal activity. The anonymity that cryptocurrency wallets offer is ideal for those who want the traceability of their money to be lost. However, it is important to note that cryptocurrency wallets are not always completely anonymous, but offer so-called pseudo-anonymity. When cryptocurrencies are purchased with traditional, traceable money, the cash flow can be traced back to its origin, therefore retrieving the purchaser's identity. To avoid this, special services known as mixers are available online, where users can send their money to be mixed with that of one of the other users before being sent to an untraceable output wallet. These services tend to have a sophisticated infrastructure and charge fees to their clients. Some popular examples available on the Bitcoin network are CoinJoin and JoinMarket. Some modern mixers are so-called non-custodial mixers, which means no entity has custody of the users' money at any time, with the logic of the mixer being handled by a smart contract on the correspondent network. An example is Tornado Cash on the Ethereum network.

When compared to normal transaction traffic on blockchain networks, different types of anomalies can be found on these networks beyond fraud that might occur because of network issues or delays without malicious intent, as detailed in works like [22–26]. As those works reveal, some of the main anomalies are the following:

- Consensus problems. These include situations that might affect the blockchain consensus mechanism, such as network partitioning, in which the blockchain is temporarily split into different forks, or vulnerabilities in the Proof of Work (PoW) algorithm.
- Double-spending attempts. These occur when the same digital asset is spent more than once.
- Smart contract vulnerabilities. These are flaws in the smart contract code that could lead to unintended behaviours or security risks.
- Atypical transaction patterns. These include sudden spikes in the transaction volume or irregular transaction sizes, which might indicate underlying issues.

Nevertheless, to the best of our knowledge, there is no approach that identifies fraud as an anomaly, and this is the issue we would like to address in this article.

Now that the main concepts of cryptocurrency fraud and blockchain anomalies have been briefly illustrated, the next section will detail related works to address the problem.

3. Related Works

As the cryptocurrency community has grown in recent years, so too has the number of fraud cases and types occurring on these networks. Several research works have attempted to extract useful information to identify the patterns of them. The most relevant investigations on this topic have proposed frameworks for modelling a cryptocurrency network and its transaction flow to capture important information such as temporal or structural features. These data can then be used along with different methods for a variety of tasks within the fraud detection context.

Earlier works showed the performance of traditional ML models applied to a classification task on transaction datasets extracted from the blockchain and manually labelled, such as Naive Bayes, SVMs, Logistic Regression, Gradient Boosting, Ada-Boost, Random Forests [3–8], Federation Learning [9], Long Short-Term Memory [10], Recurrent Neural Networks [11] or anomaly detection [13].

However, recent investigations have attempted to outperform the metrics obtained with these models by involving additional features that represent the structural topology of the transaction graph. Most of these works have made use of a variation of a Graph Neural Network [12,27], which is a deep learning model that can encode the underlying structure of a graph by combining a node's representation with that of one of its neighbours in a process known as message passing, which can consist of multiple layers, therefore combining the information of multiple degrees of the neighbourhood.

In this context, Di et al. [28] suggested a framework for modelling Bitcoin transactions as a random graph to exploit their structural properties and analyze them from the perspective of Graph Theory. This idea can be fitted in different ways to the available data and the models to be used. For example, wallets could be represented as graph nodes, with the edges being the transactions between them, as proposed by Tharani et al. [29]. For their part, Weber et al. [30] proposed a different framework in which transactions are taken as nodes themselves while edges represent the money flow between transactions. This approach turned out to be more suitable for the goal of mining data and especially for the task of identifying fraud patterns. More sophisticated works, like those performed by Lin et al. [31], suggested advanced dynamic models that also take into account the temporal features of the data.

Some of the most popular types of message-passing layers are the ones implemented in Graph Convolutional Networks (GCNs) [32] and Graph Attention Networks (GATs) [33]. Recent investigations have applied this kind of framework specifically to classify transactions as licit and illicit ones and, in most cases, no improvement has been obtained by the graph networks themselves. However, these message-passing networks can be used to

build high-dimensional embeddings that, concatenated to the nodes' original features, can improve the metrics obtained with more traditional methods such as Random Forests, as shown in [30]. Furthermore, newer studies have proposed novel frameworks, such as the Heterogeneous Graph Transformer (HGT) [18], that have not been tested in this specific context yet.

Leaving aside general fraud detection, numerous studies have been carried out with the objective of identifying specific kinds of fraud on cryptocurrency networks. This is the case in [14,15], which made use of novel data analytics techniques to identify phishing scams on different networks. In [14], a Data Augmentation Method and Hybrid Graph Neural Network Model was proposed to address sample imbalance and feature extraction issues, achieving impressive performance metrics on a real dataset. For their part, Choi and Buu [15] used a Deep Graph Traversal based on a Transformer for Scam Detection to address the complexities of Ethereum's transaction networks to detect phishing scams, outperforming traditional methods.

Other type-specific fraud detection studies have included ransomware activity tracking, mostly on the Bitcoin network, where these types of scams are commonly performed. Among these studies, the work of Akcora et al. [34] stands out due to their successful application of advanced topological blockchain graph analysis on a finely constructed dataset with different ransomware families. Other studies related to ransomware detection include [35,36].

From another perspective, a frequently unexplored approach to this topic is the use of unsupervised methods, which rely on the idea of making predictions based exclusively on the features of the data and not on the labels. The advantage of not using labels is especially important in this case since the labelling process is a complicated, costly and time-consuming process that can be skipped when following this strategy. Some of these approaches include clustering techniques such as the k-means or outlier detection methods such as the Local Outlier Factor (LOF) [19]. Most of them are traditional and well-established algorithms that have been proven to perform well in a wide range of applications and so could perform similarly in the context of this study. In fact, various works have already employed some of these techniques for anomaly detection, such as [25,37,38] with the use of an Unsupervised SVM or the Mahalanobis Distance, among others. In addition, the study by Shayegan et al. [39] proposed methods for detecting user-level anomalies by aggregating the data of multiple digital wallets owned by the same user, with the aim of identifying if their overall activity is anomalous instead of assessing individual addresses.

In addition to feature-based approaches, researchers have recently proposed models for structural anomaly detection, specifically adapted to graph data. One example is the DOMINANT [40] framework, which makes use of autoencoders [41] to combine both structural and feature-related anomalies. Applied to cryptocurrency fraud detection, the work detailed in [17] proposed the use of the Graph Convolutional Autoencoder model to enhance the detection of anomalies in cryptocurrency transactions by treating transactions as edges and accounts as nodes within a graph neural network.

4. Problem Statement and Methodology Definition

This section outlines the experiments conducted in this research, which focused on a binary classification task involving cryptocurrency transactions, using a dataset with a fixed number of features and evaluating the model's performance.

4.1. Problem Statement

In general terms, the experiments conducted in this research aimed to carry out a binary classification task on a cryptocurrency transaction (or wallet) set. More specifically, given a set of transactions (or wallets), V, with a fixed number of features each, the objective was to build a model to decide whether a given element, $v \in V$, belonged to the licit subset of transactions or the illicit one. To compute the performance of each model, it was tested

against a partially labelled dataset, which will be described in the next subsection. The approaches used in this research to build such a model followed different strategies.

On the one hand, a series of experiments using non-supervised methods were conducted to contrast them and explore the following hypothesis:

Hypothesis 1. *Generally, fraudulent activities tend to show anomalous behaviour in contrast to normal licit activities; therefore, illicit transactions could be detected by identifying unusual and abnormal patterns in the data, without resorting to manually obtained labels.*

The objective was to test several traditional and established feature-based outlier detection and clustering algorithms and to deepen the concept of anomalies in the context of fraud detection to find aspects that might improve the understanding of the relation between abnormal transactions and illicit ones.

On the other hand, a more traditional supervised approach was used. For this phase, transaction data were modelled as a graph, following the ideas of related research with state-of-the-art results. With the aim of outperforming those results, the novel HGT framework was tested, trying to leverage the heterogeneous nature of transaction data.

4.2. The Datasets and the Graph Nature of the Data

The data-gathering step is a particularly complicated task in the context of cryptocurrency transactions, even though raw transaction data are publicly available through different blockchain APIs. One of the main reasons for this is the difficulty of manually labelling transaction data, since this task requires exhaustive research to determine whether a given transaction is licit or not. Because of this, most available datasets are partially labelled. Another important factor is the great volume of transactions that are continuously being processed and the volatility of these data since they depend on different variables that are constantly changing, such as the price of the cryptocurrency on the market. Consequently, throughout this section, the different datasets used in the experimentation process of this research will be described, along with the handling of the data modelling.

As mentioned previously, transaction data have the peculiarity of carrying underlying relational information that can be extracted and utilized by classification models to make more accurate predictions. To achieve this, the dataset needs to be modelled as a graph, and the most suitable way of doing this is by modelling the dataset as a homogeneous graph, $\mathcal{G} = (V, E)$, where V is the transaction set and $E \subseteq V \times V$ is the set of edges so that $e = (s, t)$ represents the flow of cryptocurrencies or other assets from transaction $s \in V$ to transaction $t \in V$. This approach was utilized by researchers in [30] and applied to the Elliptic Bitcoin Dataset, a dataset with more than 200,000 Bitcoin transactions (approximately USD 6 billion) published by the Elliptic company in the same research work. Each of the transactions in this dataset is labelled as either licit (0), illicit (1) or unknown (2) and has 166 features. The first 94 features represent local information about the transaction, such as the number of inputs and outputs, the transaction fee and the output volume, while the other 72 features are obtained by aggregating the local ones. Further details on the feature extraction process are not available due to the intellectual property policies of the Elliptic company. Following the previously explained mechanism, the dataset also contains a list of approximately 230,000 edges representing the Bitcoin Unspent Transaction Output (UTXO) flow between transactions. The dataset is described as being constructed from a set of transactions explicitly known to be licit and illicit and then extended with random transactions broadcast to the blockchain in evenly spaced time windows. Thus, one of the features of the transactions is a discrete temporal component ranging from 1 to 49, referred to as the timestep. For this reason, we believe that the Elliptic dataset constitutes a reliable representation of the Bitcoin transaction flow, and including different fraud categories in the illicit label made it ideal for the purpose of this study. Therefore, this dataset was widely used in a good part of the experiments in our work.

Additionally, some popular datasets might also include features for both transactions and wallets, as well as the relations between them. In these cases, the transaction graph can be modelled as a heterogeneous graph, $\mathcal{G} = (V_{tx}, V_w, E^w_{tx}, E^{tx}_w)$, in which V_{tx} is the set of transactions (with its transaction-specific features), V_w is the set of wallets (with its wallet-specific features) and $E^w_{tx} \subseteq V_{tx} \times V_w$ and $E^{tx}_w \subseteq V_w \times V_{tx}$ are the sets of the `Transaction`→`Wallet` and `Wallet`→`Transaction` edges, respectively. This is the case for the Elliptic++ [12] dataset, presented in 2023 as an improved extension of the Elliptic dataset, containing a set of around 800,000 wallets with 55 features each and the relations between them as actors in the transactions of the original dataset. Given the heterogeneous nature of this dataset, it was also used for another part of the experiments in this research.

A relevant aspect of both datasets is that the number of transactions (and wallets) labelled as illicit in relation to the licit ones is highly imbalanced, which is a fact that had to be taken into account during the experimentation process.

4.3. Methodology

The experimentation phase of this research was carried out using popular Python libraries such as Scikit-Learn for common ML algorithms and Pytorch for tensor handling and more advanced deep learning algorithms, in conjunction with Pytorch Geometric (PyG) for the implementation of GNNs and other graph-related applications.

Due to the high volume of the data and the complexity of the utilized models, specifically the GNNs, it was necessary to apply a mini-batch training technique to avoid Out Of Memory (OOM) errors. To split the transactions graph into separate batches without losing structural information, it was utilized as a unit for batching the subgraph corresponding to each timestep, since the transactions from different timesteps were not linked to each other. Formally, each subgraph can be defined in the following manner:

$$G_t = (V_t, E_t) \quad \text{where} \quad \begin{cases} V_t = \{v \in V \mid \text{TIMESTEP}(v) = t\} \\ \text{and} \\ E_t = \{(s,t) \in E \mid s,t \in V_t\} \end{cases} \qquad t = 1, \ldots, 49 \qquad (1)$$

Then, the G_ts could be grouped to form training batches of a specific size.

Concerning benchmarking and the evaluation of the models' performance, selected metrics were chosen to account for the imbalance of the data. Thus, the Accuracy metric was discarded since it only measures the number of successful predictions, which can be very high for the licit class but low for the illicit class. Instead, other evaluation functions such as the Precision, Recall and F1 were selected, since they penalize the amount of false positives, false negatives and both, respectively. Additionally, for the sake of displaying an accurate picture of the predictions performed by the models, a confusion matrix is provided for each experiment.

Finally, as a complementary measure, the ROC curve graphic is also provided for each experiment, showing the proportion of true positives to false positives as the decision threshold grew. This threshold could be a probability in the case of supervised learning models or an outlier factor in the case of anomaly detection experiments. The Area Under the Curve (AUC) is also provided as another metric.

5. Experimentation Process

The collection of experiments carried out in this research can be grouped into four categories:

1. First, an exploratory data analysis (EDA) was conducted, including several tasks such as a reduced component visualization of the dataset, a class imbalance analysis and a graph subsample visualization.
2. Second, a set of anomaly detection experiments were carried out using different popular algorithms to find the relation between the fraudulent transactions and the anomalies, as stated in the hypothesis that was described in detail at the beginning of this section.

3. Third, clustering was conducted using the k-means algorithm on the transactions dataset, employing a non-supervised approach to identify correlations between the transaction labels and the resulting clusters.
4. Lastly, a modern GNN algorithm was tested on the full Elliptic++ dataset, this time following a supervised methodology, which has traditionally been used more in this context. Specifically, the deep learning framework used was the Heterogeneous Graph Transformer (HGT) and aimed to leverage the heterogeneous nature of the Elliptic++ dataset to reach or even outperform the benchmarks obtained with other deep learning algorithms already tested on this dataset.

5.1. Exploratory Data Analysis

This subsection will introduce several graphs and diagrams that illustrate the nature of the Elliptic and Elliptic++ datasets to extract useful information about their features, the class imbalance and the relational structure.

An important thing to consider before diving into the experiments is the high dimensionality of the data being worked with. Specifically, the transactions dataset had a total of 183 features. Dealing effectively with this number of features was crucial for both the experiments and the visualization of the data, due to the noise they can produce. In later sections, dimensionality reduction methods oriented to experimentation, such as PCA, will be treated in detail. However, for the sake of a clear visualization of the data, other methods tend to be more suitable, such as t-Distributed Stochastic Neighbour Embedding (t-SNE). In this manner, a precise representation of the original data can be obtained in a reduced two- or three-dimensional space that can be represented on a concise plot.

Accordingly, Figure 2 shows the output of the t-SNE algorithm applied in a three-dimensional space to the subset of transactions that occurred at the seventh timestep. The first subfigure includes the transactions with a known label, namely licit (in green) or illicit (in red). The second one also includes the transactions with an unknown label (in orange). Both of them give a hint about the imbalance that exists between the different labels of the dataset.

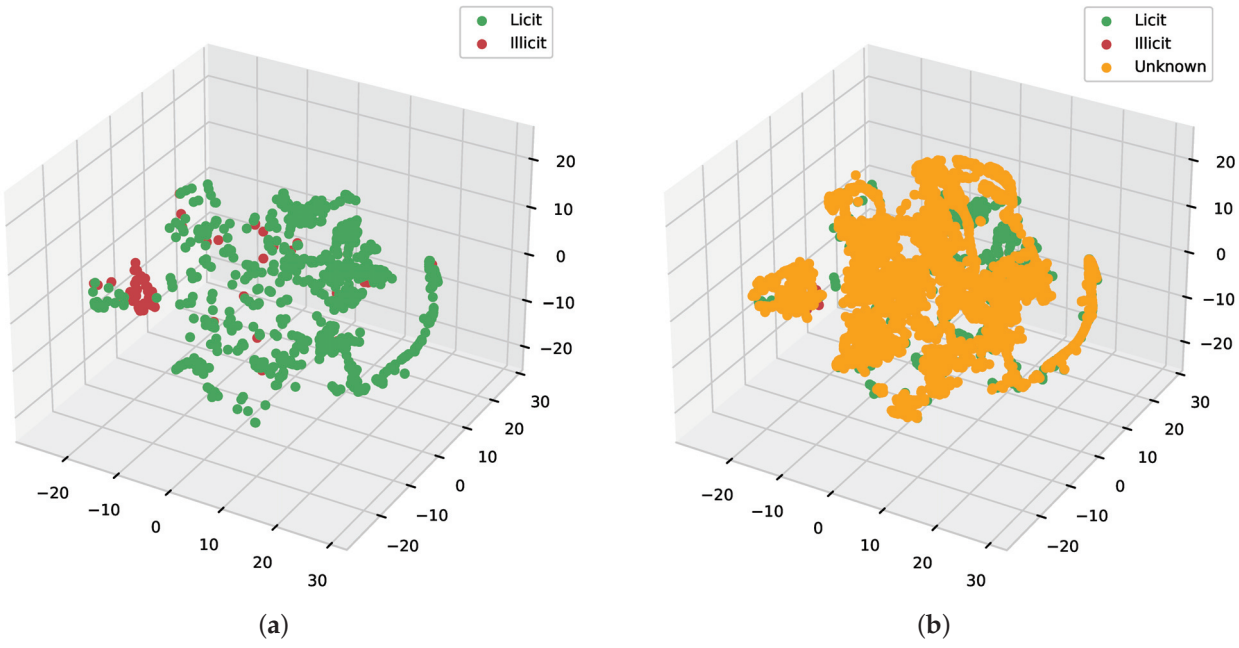


Figure 2. Reduced visualization of transactions at timestep 7. (**a**) Known label transactions; (**b**) known and unknown label transactions.

To take a closer look at the imbalance, Figure 3 shows the number of transactions of every label (licit, illicit and unknown) at each timestep of the dataset, from 1 to 49. A high imbalance between different classes can be observed in the figure, and it is also remarkable that this imbalance varied from timestep to timestep with high volatility. Additionally, Figure 4 shows the proportion of licit and illicit transactions at each timestep, confirming this tendency.

Equivalently, Figures 5 and 6 show that the set of wallets showed similar behaviour to that of the transactions set.

Using a probability theory approach, the mutual information shared between the label and a feature, treated as two random variables, could be estimated by sampling both from the dataset. This provided a measure of their dependency, allowing us to draw conclusions about the extent to which a feature influenced the class. In this manner, Figures 7 and 8 show the amount of estimated mutual information of each feature concerning the class (licit or illicit) for both the 183 features of the transactions and the 55 features of the wallets, respectively.

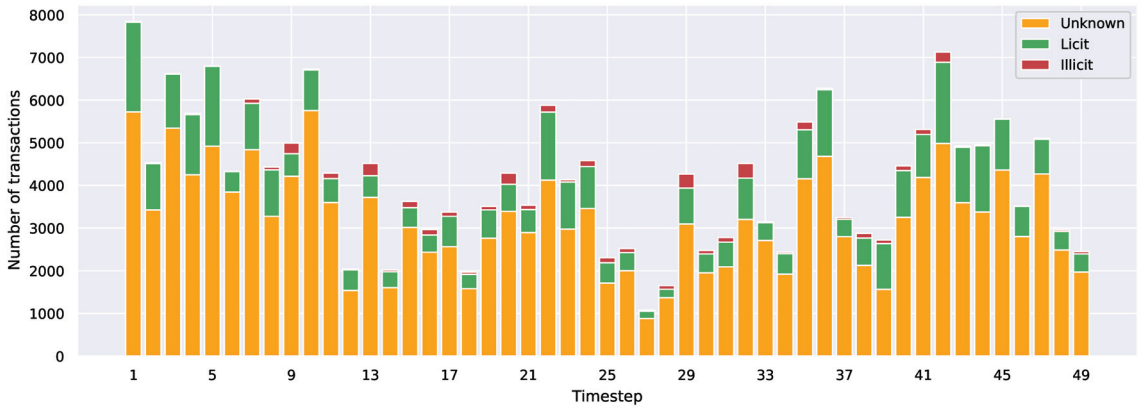


Figure 3. Number of transactions in each class per timestep.

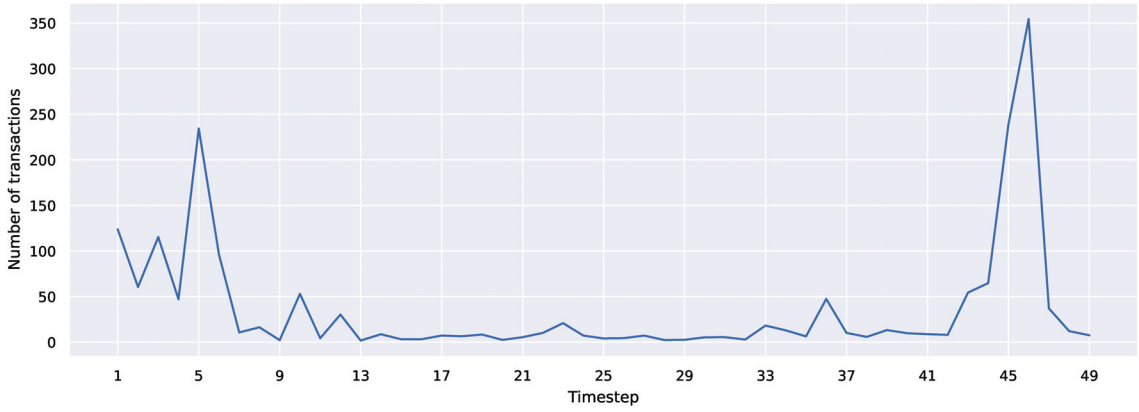


Figure 4. Imbalance between licit and illicit transactions per timestep.

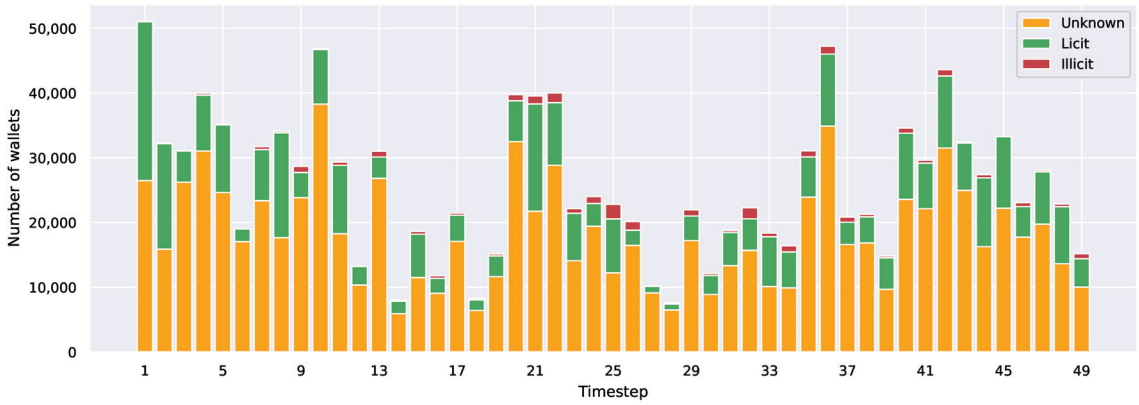


Figure 5. Number of wallets in each class per timestep.

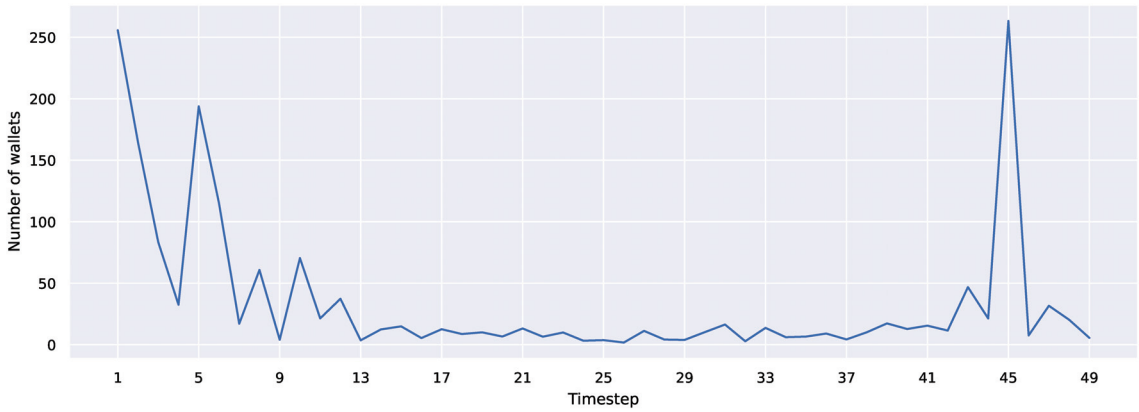


Figure 6. Imbalance between licit and illicit wallets per timestep.

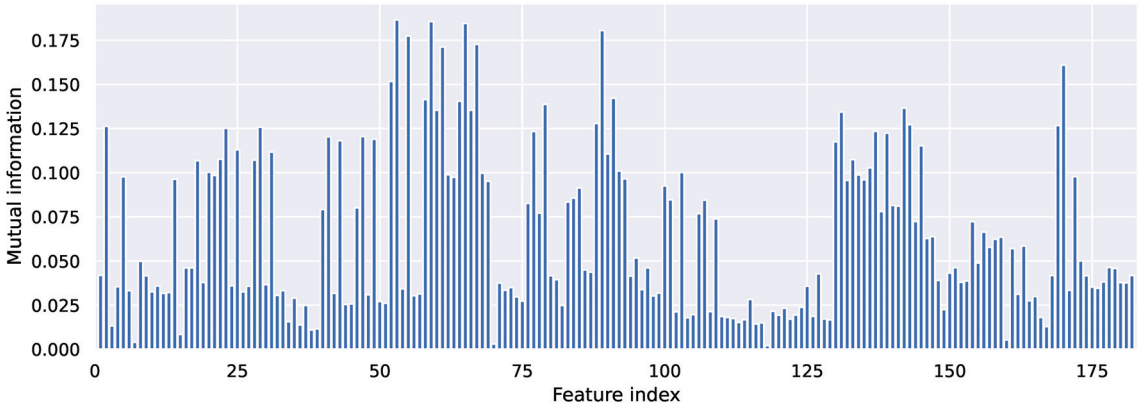


Figure 7. Mutual information of transaction features.

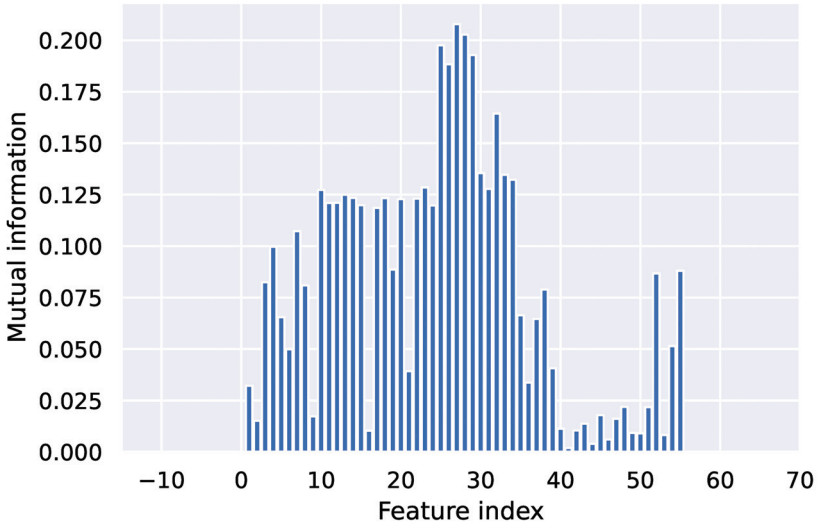
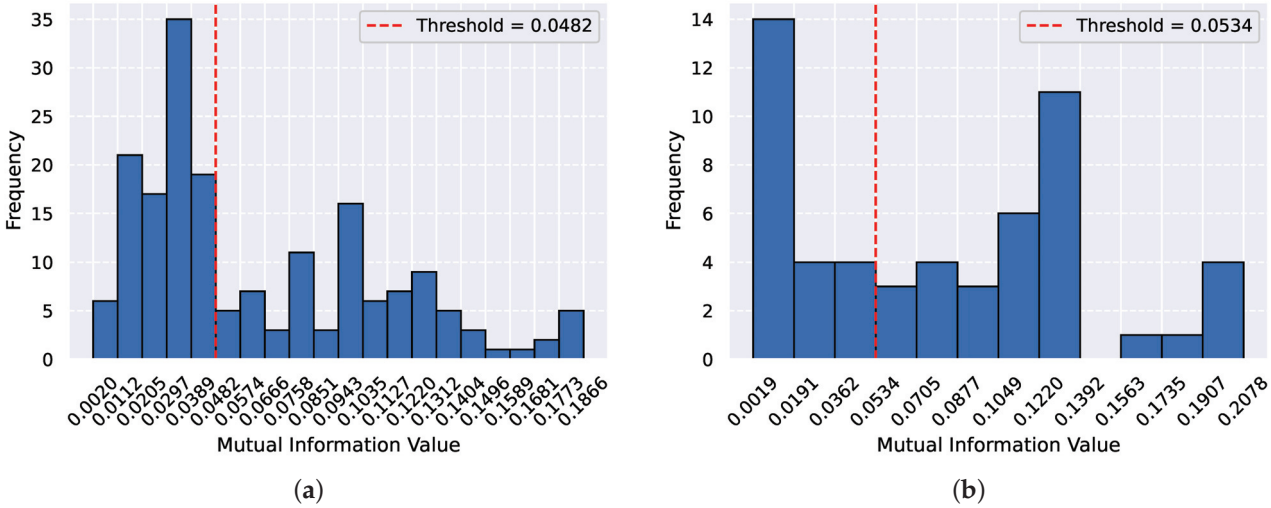


Figure 8. Mutual information of wallet features

As can be observed in the figures, especially for the transactions, there were slight patterns in some groups of features. This could be attributed to the fact that some features may have been obtained by aggregating other features. However, as stated in Section 4.2, the authors of the dataset do not provide such information. Although in both cases, the amount of mutual information was generally low, there existed slight differences between the features, which suggested that a component reduction algorithm could be beneficial for some models. On the first iteration, a feature ablation analysis was carried out by removing all the features below a threshold in the mutual information quantity. In order to select this threshold, a histogram was plotted for both transaction and wallet features to analyze the distribution of the mutual information values in both cases. Figure 9a,b show these histograms with the chosen threshold for both transactions and wallets, respectively.



(**a**) (**b**)

Figure 9. Distribution analysis of MI of features. (**a**) MI histogram for transaction features; (**b**) MI histogram for wallet features.

Figures 10 and 11 show the final result of the feature ablation compared to the previous mutual information plots. Unfortunately, this component reduction technique showed no improvement in subsequent experiments compared to other feature reduction algorithms such as Principal Component Analysis (PCA).

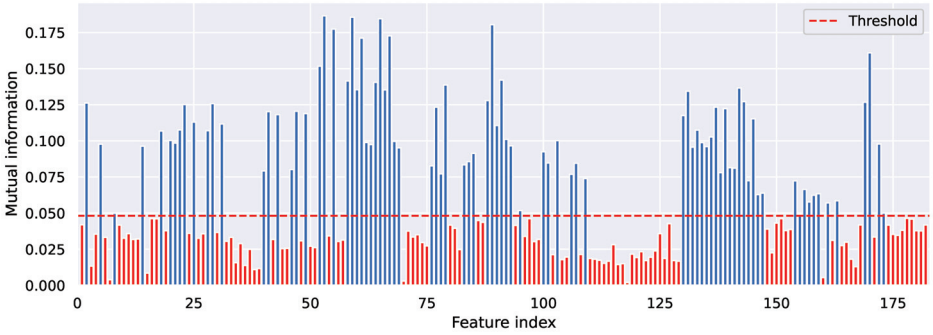


Figure 10. Ablation of transaction features. The red bars represent transaction features that have been removed because their mutual information is below the threshold line.

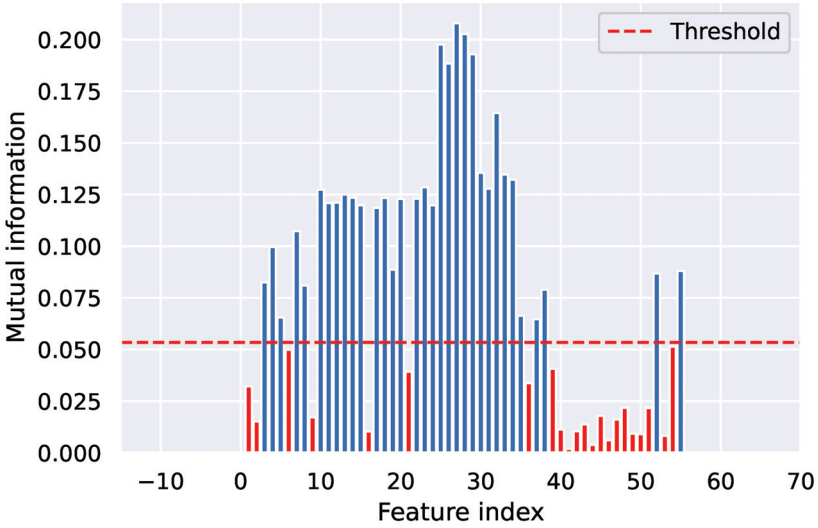


Figure 11. Ablation of wallet features. The red bars represent wallet features that have been removed because their mutual information is below the threshold line.

Unlike this technique, PCA is a more mathematical approach based on a linear dimensionality reduction aimed at maximizing the variance captured by the principal components. In order to find a suitable number of components, the cumulative explained variance was plotted and is shown in Figure 12. The elbow at 75 components indicates that this number of reduced features is enough to be input into the outlier detection algorithms since it balances dimensionality reduction and information retention.

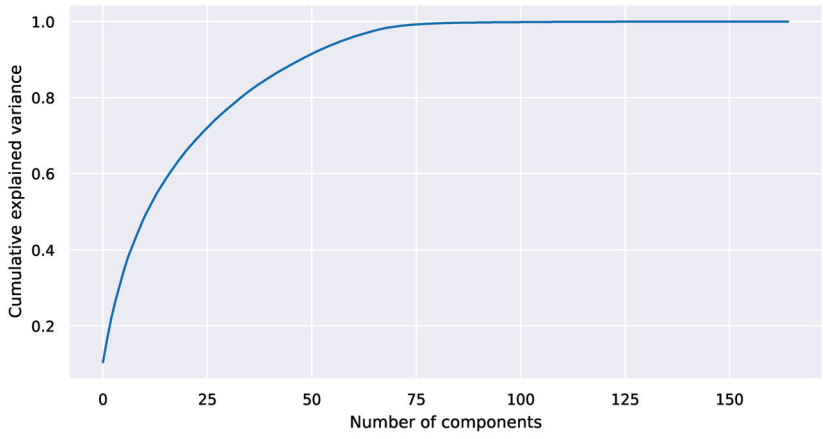


Figure 12. Cumulative explained variance (PCA).

After analyzing the features, visualizing a sample of the transactions graph can be useful for understanding the Bitcoin flow between wallets, but its construction requires the selection of a node and the careful exploration of the graph obtained from it. More specifically, an illicit wallet was chosen as a starting node to ensure the existence of fraudulent entities in the subsample, and a Breadth-First Search-like algorithm was applied to it with a depth of three levels. Figure 13 shows the resulting subgraph.

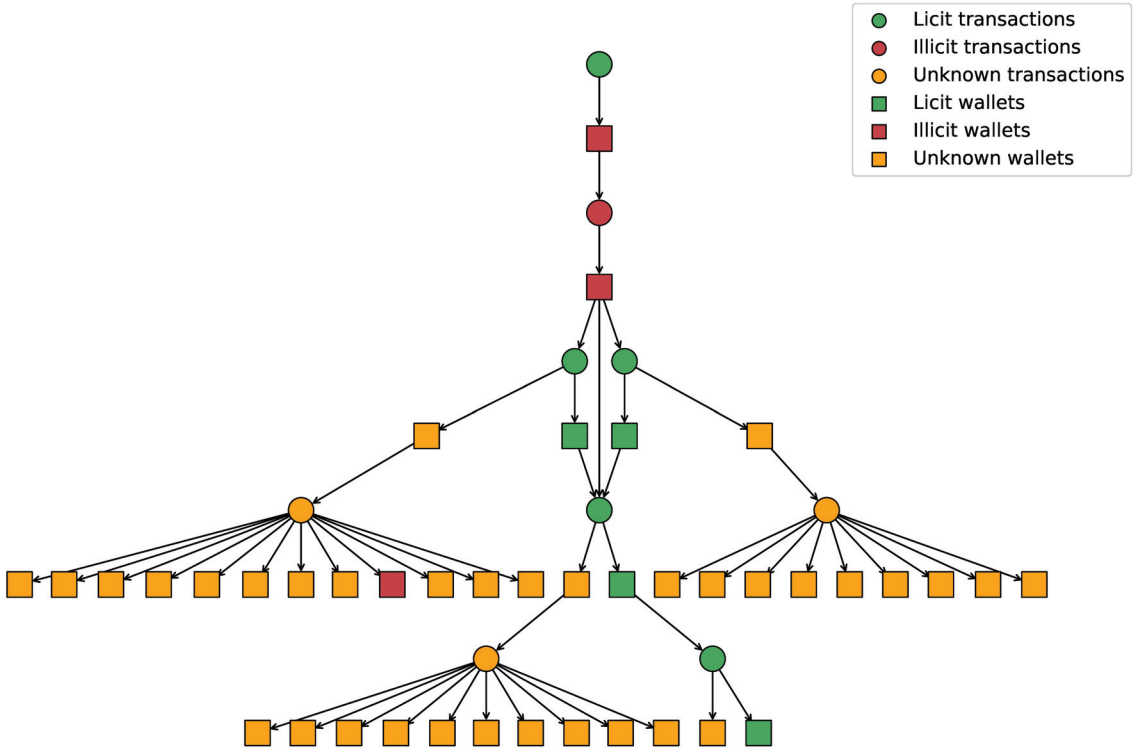


Figure 13. A visualization of a sampled subgraph.

5.2. Anomaly Detection

Throughout this section, we describe how the Elliptic transaction dataset was tested against a series of well-established outlier detection algorithms for the sake of testing Hypothesis 1. The objective was to find a correlation, at least a partial one, between the transactions classified as outliers by the algorithms and those labelled as illicit.

The choice of the anomaly detection algorithm needed to take into consideration different aspects intrinsic to the anomaly concept. For this experiment, a set of four common outlier detection algorithms was selected, as they represent some of the main techniques that are widely used in feature-based outlier detection.

- Local Outlier Factor (LOF) [19]: This algorithm evaluates the local outlier factor for an element based on the local density of its K nearest neighbours, considering their features for computing the distance. It is particularly robust when applied to datasets with varying densities or clusters. Therefore, it can be effective in detecting some types of fraud since they can often appear as localized anomalies, for example, a transaction with unusual patterns within a cluster of transactions.
- Isolation Forest (ISOF) [42]: This method uses several random binary trees with different decision thresholds for randomly selected features to isolate anomalous samples on the leaves. It is particularly optimal for high-dimensional data, which were analyzed in this study, and has a very efficient time complexity, which could help scale the algorithm to large datasets such as the entire blockchain.

- SGD One-Class SVM: This approach adapts the classical Support Vector Machine to a non-supervised approach by using a Stochastic Gradient Descent to efficiently fit a boundary that includes inlier data and keeps outliers on the outside. The ability to model complex boundaries can make it effective in fraud detection, since anomalous patterns can be distinctly separated from normal behaviour.
- Elliptic Envelope: This method assumes that the data follow a Gaussian distribution and fits an ellipse that serves as a boundary between inliers and outliers. Many types of financial fraud often appear as deviations from standard Gaussian-like behaviour [43] (for example, the distributions of transaction amounts). Under these assumptions, the algorithm can efficiently identify these deviations.

All the algorithms were applied to the 75 reduced dimensions obtained after PCA, described in the preceding subsection. While both labelled and unlabelled data from the transactions dataset were fed to the outlier detection algorithms, only labelled and known data were taken into account for measuring the performance of these models. Since all these algorithms return an outlier score for each transaction, a threshold had to be fixed to establish the boundary between inliers and outliers. Therefore, and to deal with the high class imbalance of the dataset, this threshold was set so that 8% of the predictions were classified as outliers, following a licit/illicit proportion as near as possible to the real labels of the dataset. As a result, Table 1 shows the metrics of the four aforementioned algorithms, taking as positive predictions (illicit) those transactions classified as outliers and negative predictions (licit) those classified as inliers.

Table 1. Outlier detection methods' performance. I: inlier. O: outlier.

Algorithm	Precision		Recall		F1 Score		Accuracy	ROC-AUC
	I	O	I	O	I	O		
LOF ($K = 2$)	0.91	0.22	0.95	0.13	0.93	0.17	0.87	0.495
ISOF ($N = 1000$)	0.87	0.00	0.71	0.01	0.78	0.00	0.64	0.096
SGD One-Class SVM	0.93	0.13	0.49	0.68	0.65	0.21	0.51	0.567
Elliptic Envelope	0.87	0.01	0.72	0.03	0.79	0.02	0.66	0.173

Some of the best metrics were obtained with the LOF method using $K = 2$ neighbours, even though experiments conducted with other values of K obtained a very similar performance. This suggests that the local density was not that relevant for anomaly detection on this dataset. Figure 14 shows the confusion matrix of the results obtained in this specific experiment.

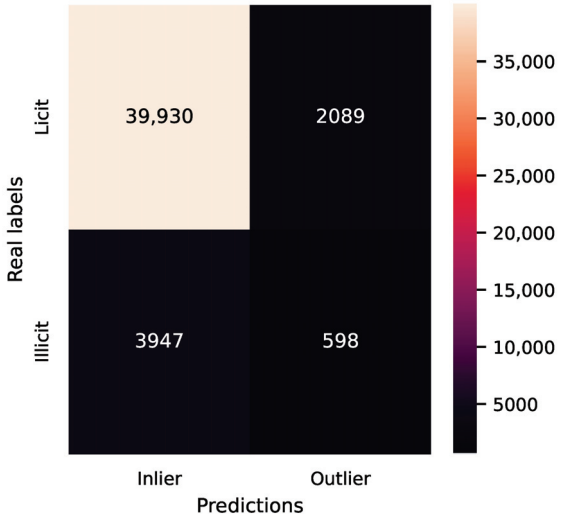


Figure 14. Confusion matrix for Local Outlier Factor ($K = 2$).

Although the table shows that most algorithms seemed to obtain poor performance, particularly when predicting illicit labels, it is worth noting that these predictions were based only on the features of the data and did not take into account their labels at any moment. Thus, they were expected not to reach the performance of traditional supervised learning models, but on the other hand, there was the advantage of not needing manually obtained labels, avoiding a tedious and complicated process. Moreover, these methods are noticeably more performant than others based on deep learning frameworks or graph-related solutions and therefore are more suitable to be applied on massive datasets or even the entire blockchain.

Still, these results show that Hypothesis 1 barely holds true when using strictly feature-based anomalies. Thus, the experiments suggested that the anomaly concept might be extended by taking into account other aspects such as the relational information of the transactions, which indeed, has been proven to be crucial in the context of fraud detection, as shown in other investigations. This could be achieved by making use of more sophisticated graph anomaly detection frameworks such as Graph Autoencoders [44], which identify structural anomalies based on high link reconstruction errors, or hybrid models such as DOMINANT [40], which combine both feature and structural anomalies.

5.3. Clustering

In this phase of the experimentation, another non-supervised learning approach was tested on the transactions' data: the k-means clustering algorithm. A clustering algorithm aims to group elements of the dataset into different clusters based on the similarity of their features.

For this specific case, splitting the set of transactions into different clusters can help distinguish between fraudulent and legal transactions depending on the cluster they belong to. In this manner, the results of different experiments were analyzed by contrasting each transaction's cluster with its label (licit or illicit). This analysis could be summarized in the corresponding contingency matrix. Additionally, to evaluate the quality of the generated clusters, a silhouette score analysis was carried out for each experiment to measure the cohesion of each cluster.

To handle the large number of points in the dataset, the experiments were carried out using a specific implementation of the algorithm called the mini-batch k-means, which carries out the creation of batches with a minimal impact on the results. After tweaking several parameters, including the number of clusters (K), some interesting results were obtained with the values of $K = 7$, $K = 33$ and $K = 49$. Figures 15–17 show their respective contingency matrices. In addition, Figure 18 shows the silhouette diagram for the $K = 33$ case.

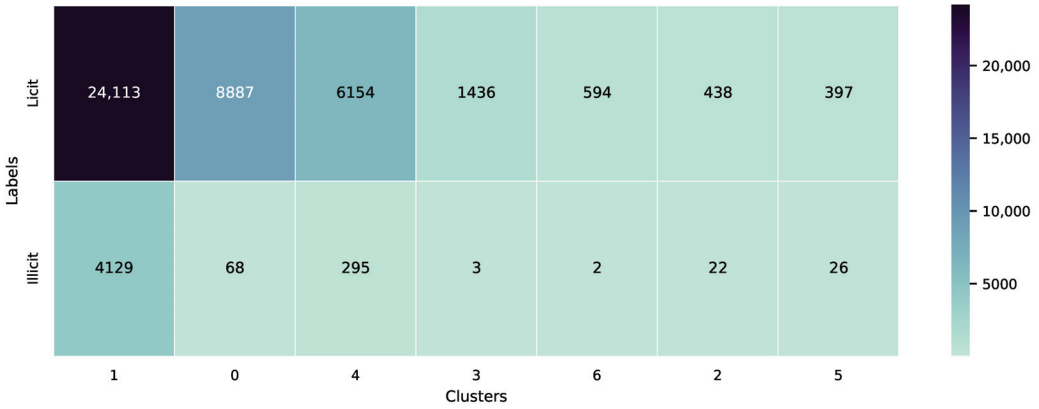


Figure 15. Contingency matrix of k-means $(K = 7)$.

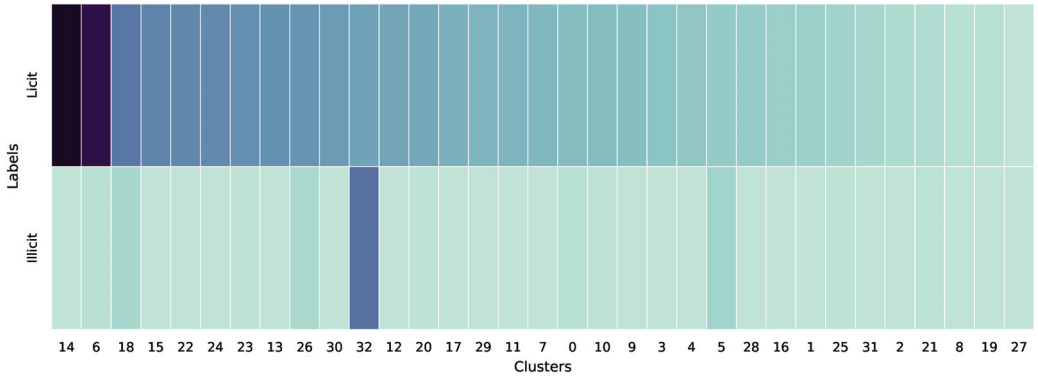


Figure 16. Contingency matrix of k-means ($K = 33$).

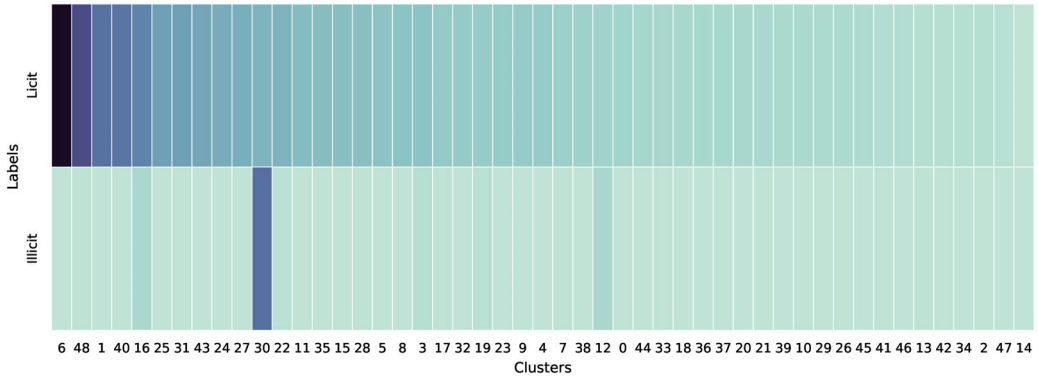


Figure 17. Contingency matrix of k-means ($K = 49$).

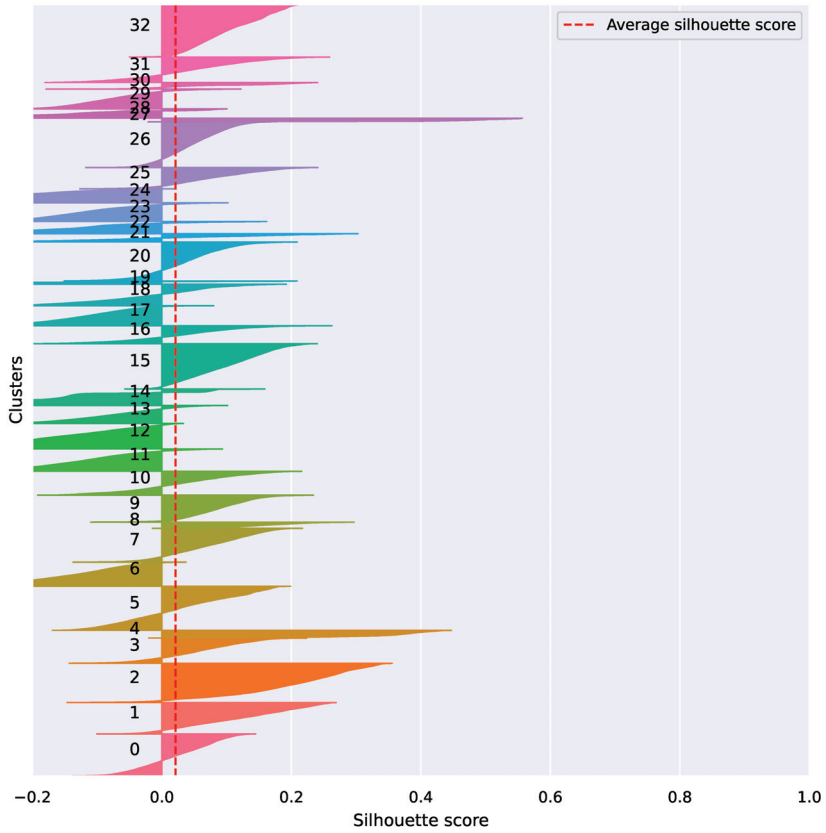


Figure 18. Silhouette score diagram of k-means ($k = 33$). Each color represents a different cluster.

While the results obtained with $K = 7$ showed a cluster with a remarkable number of illicit transactions (4129), that same cluster also contained the majority of licit transactions (24,113), making it useless for the task of distinguishing between both labels. Conversely, both experiments conducted with $K = 33$ and $K = 49$ revealed a cluster with a large number of illicit transactions while keeping the number of licit ones low, as illustrated by clusters 10 and 30 in Figures 16 and 17, respectively. The outcome of these experiments is significantly more relevant for the proposed task, as features shared by the transactions in those clusters might indicate a fraudulent tendency.

On the other hand, Figure 18 shows a very poor performance in terms of the silhouette analysis, with an average silhouette score close to 0. This indicates a low cohesion of the generated clusters and a generally bad clustering quality, suggesting that the features of the data, and especially the high dimensionality, hindered their clusterization.

5.4. Graph Deep Learning

Both Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs) have shown excellent performance when applied to traditional supervised learning tasks on cryptocurrency transactions and more specifically on the Elliptic dataset, as stated in previous related works. Following this approach, this section of the experiments aimed to test a recent state-of-the-art model that can leverage the heterogeneous nature of the Elliptic++ dataset.

The Heterogeneous Graph Transformer (HGT) [18] is a novel deep learning framework designed for heterogeneous graphs that combines the ideas of conventional GNNs with the Transformer [45] architecture, using different mutual attention mechanisms depending on the graph node type. This complex architecture can be used to manage different attention coefficients and weights for each node type in the Elliptic++ dataset, namely, transactions and wallets.

In order to carry out the experiments with this framework, an HGT Binary Classifier model was developed specifically for this case, using a linear transformation that projected the feature vectors into the embedding space and a series of HGT convolutional layers, followed by Leaky ReLU activation functions. Finally, another linear transformation was applied to obtain an output logit. The parameters for this model that were used in the experiment are displayed in Table 2.

Table 2. The parameters for the HGT classifier.

Model Parameter	Value
Embedding dimension	128
Number of message-passing layers	2
Number of attention heads	4

As described in Section 4.3, a mini-batch training strategy was applied to fit the complex architecture of the HGT model into the CUDA device memory. The subgraphs corresponding to each timestep in the dataset were used as individual units for batching. In addition, the feature matrices of both transactions and wallets were normalized in advance. For the training phase, an Adam optimizer was utilized with a custom loss function that aimed to minimize the loss of both transactions and wallets, taking into account a custom parameter, β, used to balance both loss functions, as shown in the following formula:

$$L = \beta \cdot L_{wallet} + (1 - \beta) \cdot L_{tnx} \tag{2}$$

Both transaction and wallet loss functions consisted of a binary cross entropy loss with a custom positive weight that helped with the class imbalance. Additionally, to deal with the class imbalance differences between timesteps previously analyzed in Section 5.1, a stratified train/test split strategy was applied to the dataset. All the parameters used during the training phase can be found in Table 3.

Table 3. Training parameters.

Training Parameter	Value
Batch size	8
Positive weight for wallets	5
Positive weight for transactions	5
Number of epochs	600
Learning rate	0.01
Weight decay	0.001
Wallet/tnx loss balance (β)	0.4

Figure 19 illustrates the evolution of the binary cross entropy loss of both wallets and transactions and the combined loss function throughout the 600 epochs of training. This plot shows an irregular decreasing trend with numerous fluctuations in most loss functions, typical of batch training in which gradients are computed for each batch.

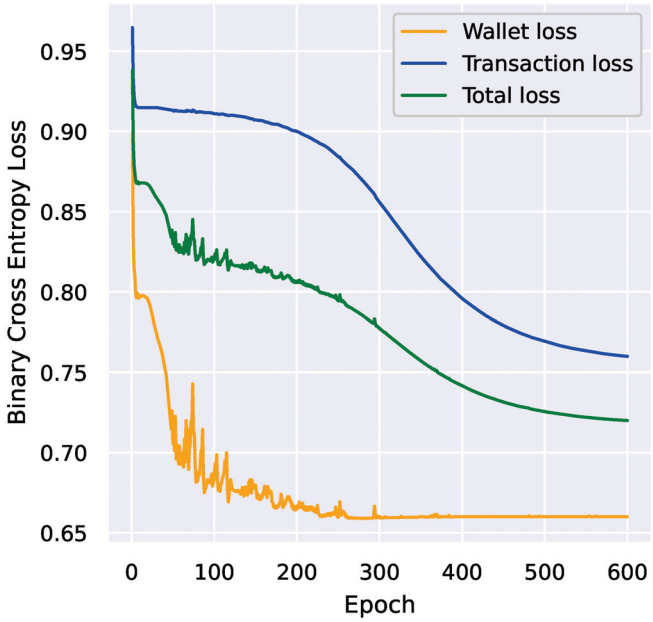


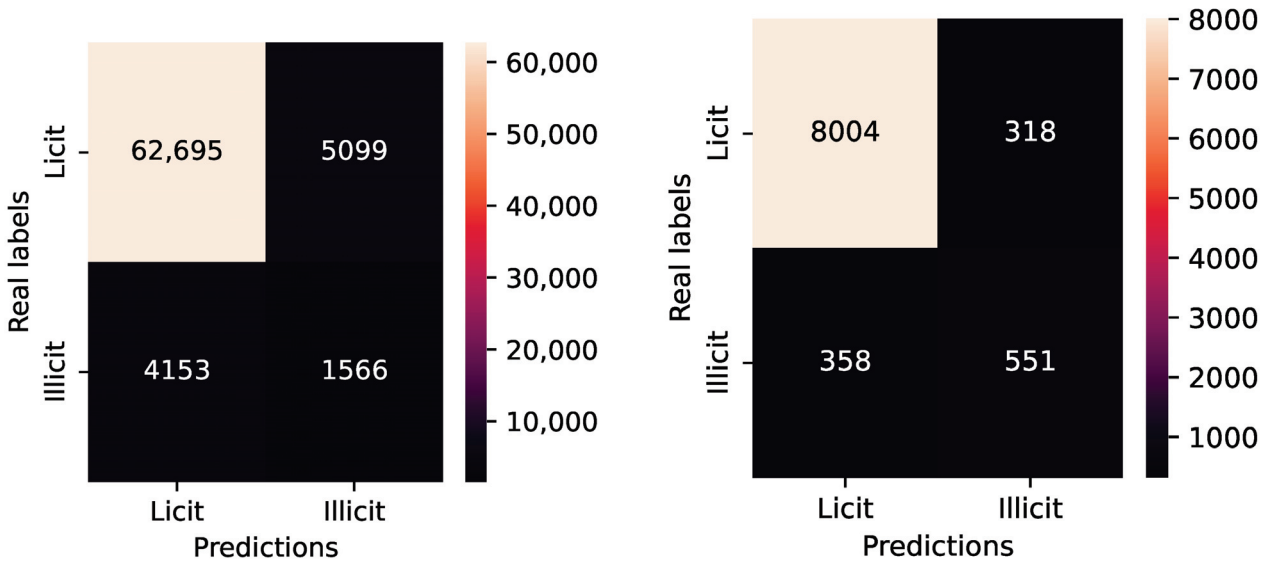
Figure 19. Loss functions during HGT model training.

Finally, the results of the experiments are displayed in Figures 20 and 21 with the confusion matrices and ROC curves of both wallets and transactions. Additionally, Table 4 summarizes the evaluation metrics of this experiment for both node types in the graph.

Table 4. The performance metrics of the HGT classifier experiment.

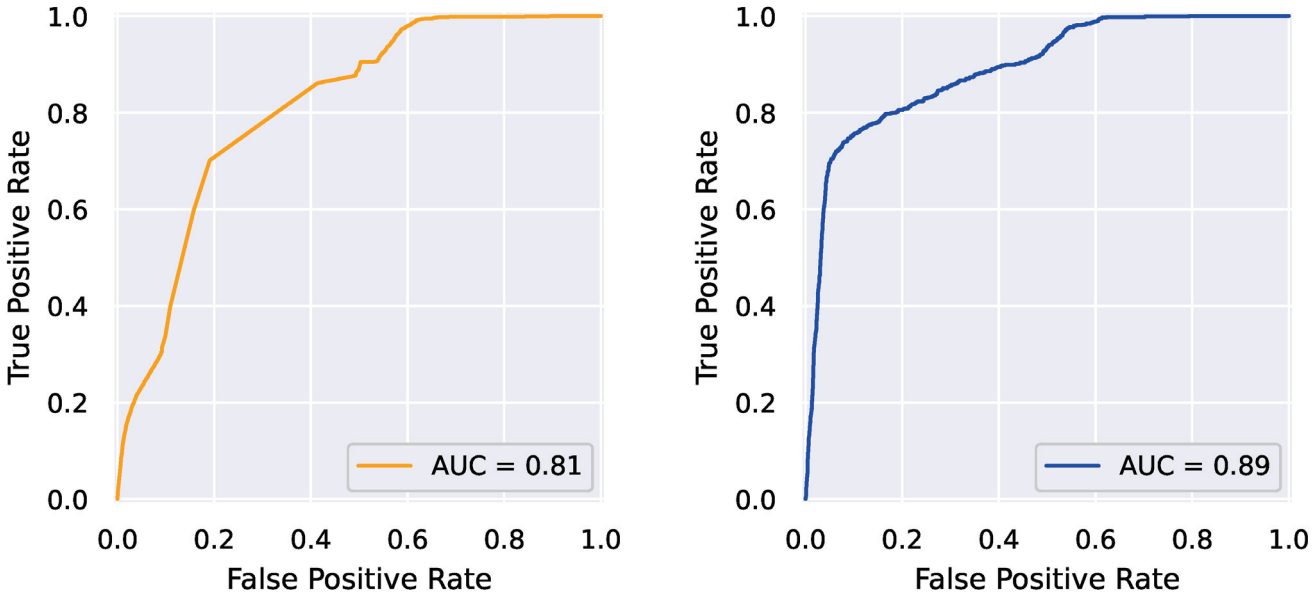
Node Type	Precision		Recall		F1 Score		Accuracy	ROC-AUC
	Licit	Illicit	Licit	Illicit	Licit	Illicit		
Wallets	0.94	0.23	0.92	0.27	0.93	0.25	0.87	0.812
Transactions	0.96	0.63	0.96	0.61	0.96	0.62	0.93	0.892

Although both confusion matrices show results that are not diagonal, it is important to take into account the class imbalance present in both sets. Therefore, it was expected to have a considerably large number of elements in the upper left quadrant of both matrices compared to the other quadrants. On the other hand, both ROC curves indicate correct behaviour, with an area under the curve (AUC) close to 1 in both cases.



(**a**) (**b**)

Figure 20. Confusion matrices of results of HGT model. (**a**) Confusion matrix for wallets; (**b**) confusion matrix for transactions.



(**a**) (**b**)

Figure 21. ROC curves of results of HGT model. (**a**) ROC curve for wallets; (**b**) ROC curve for transactions.

In addition, Table 5 provides a comparison of the HGT results for transaction nodes with other experiments from the literature that applied graph techniques to the standard Elliptic transaction dataset. Although the HGT did not directly outperform the other models, it should be noted that our experiments were carried out on a larger dataset with two different types of nodes (wallets and transactions). No direct comparisons with other experiments on the Elliptic++ dataset were carried out since no works that apply heterogeneous graph approaches to this dataset were found in the current literature.

Table 5. A comparison of the HGT with other models. The first two were trained and evaluated using the standard Elliptic dataset, while the HGT was applied to the entire heterogeneous Elliptic++ dataset, and only the metrics for illicit transactions are shown.

Method	Illicit Transactions		
	Precision	Recall	F1 Score
Graph Convolutional Network (GCN) ^{Elliptic} [30]	0.81	0.51	0.63
Graph Attention Network (GAT) ^{Elliptic} [46]	0.92	0.86	0.89
Heterogeneous Graph Transformer (HGT) $^{Elliptic++\ TX}$	0.63	0.61	0.62

Overall, the HGT has proven to be an excellent framework for heterogeneous graph deep learning, achieving state-of-the-art results in the proposed task and revalidating the effectiveness of the classical supervised learning approach to fraud detection, despite requiring a large set of manually labelled samples.

6. Conclusions

With the growth of cryptocurrency, the fight against fraudulent activities has become a challenge. After introducing the background on cryptocurrency fraud and analyzing the related works, this article addressed the problem of identifying fraud in a large set of transactions extracted from the Bitcoin network. The goal was to find reliable methods to label Bitcoin transactions, taking into account their features.

This study analyzed the effectiveness of a variety of ML methods with different approaches to the task of identifying fraud on the Bitcoin cryptocurrency network. Although the work could be generalized to other currencies such as Ethereum by following a similar method, a large enough dataset is needed. Moreover, the transaction graph for each network should be adapted to the intrinsic features of that specific currency. For example, not all cryptocurrencies follow the UTXO model used by Bitcoin, which is the mechanism that was used to build the edges between transactions in the Elliptic dataset. At the same time, the Elliptic dataset also has some limitations, such as the fact that its authors do not provide detailed information about the feature selection, the transaction gathering or the graph construction processes and if graph-specific properties like the density or node degree distribution were taken into account.

On one hand, the results obtained with unsupervised learning methods have shown the usefulness of these approaches for detecting illicit transactions in the absence of a manually labelled dataset, despite still being far from the performance of supervised methods. Unfortunately, these methods also involve a series of potential drawbacks and limitations, such as the absence of labelled data for validation, difficulty in distinguishing between fraudulent patterns and other anomalous but licit activities, and sensitivity to noise and irrelevant data.

On the other hand, the experiments carried out using the HGT framework revalidated the capability of deep learning methods in supervised approaches, particularly graph-based methods. The importance of data heterogeneity was shown in the results, although no further experiments were performed to isolate which specific components of the HGT model were responsible for this.

Combining both unsupervised and supervised approaches in a more ambitious experiment could bring several benefits, such as enhancing the anomaly detection process, filtering out outliers before the use of deep learning models, handling newer and unknown fraud patterns or improving the explainability of the results.

The results emphasize the importance of integrating both the structural information and the intrinsic features of the data, particularly in the context of transactional data due to their relational nature. Aggregating the information of a given transaction with that of its neighbouring transactions is a crucial step to decide whether it might be illicit or not, since the commission of the fraud itself can occur several hops before or after the given transaction. This aligns with related works that applied graph frameworks to supervised

learning methods, obtaining remarkable improvements. With this work, we believe that unsupervised methods can also benefit from the use of these frameworks through the detection of structural anomalies or the aggregation of neighbouring features.

As threats to the validity of our work, due to the dataset's high imbalance, it is necessary to consider additional evaluation metrics for the supervised approach, such as the Matthews correlation coefficient or the area under the Precision–Recall curve, because the Precision, Recall, F1 Score and ROC-AUC may not capture all the nuances. Also, for the unsupervised approaches, alternative clustering methods and pre-processing techniques must be explored to improve the cluster cohesion.

Although considerable progress has been made in recent years, several research lines remain open in this field. Therefore, in future works, one significant challenge will be to improve the quality, size and availability of labelled cryptocurrency datasets. On the other hand, tuning models and enhancing their adaptability to other types of data is another major challenge. Furthermore, incorporating hybrid approaches that can leverage the potential of graph deep learning frameworks in a non-supervised fashion is another interesting direction for future research.

Author Contributions: Conceptualization, V.P.-C. and F.J.; data curation, V.P.-C. and F.J.; investigation, V.P.-C. and F.J.; methodology, V.P.-C. and F.J.; software, V.P.-C.; validation, V.P.-C. and F.J.; writing—original draft, V.P.-C. and F.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been partially supported by the Center for Research in Forensic and Security Sciences of the Universidad Autónoma de Madrid (ICFS-UAM) (in Spanish *Centro de Investigación en Ciencias Forenses y de la Seguridad de la Universidad Autónoma de Madrid*, ICFS-UAM).

Data Availability Statement: The datasets used in this study can be found at the following links: Elliptic dataset, available at https://www.kaggle.com/datasets/ellipticco/elliptic-data-set accessed on 15 January 2025; and Elliptic++ available at https://github.com/git-disl/EllipticPlusPlus, accessed on 16 January 2025. The original contributions presented in this study are included in the article.

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

References

1. Burgess, T. A multi-jurisdictional perspective: To what extent can cryptocurrency be regulated? And if so, who should regulate cryptocurrency? *J. Econ. Criminol.* **2024**, *5*, 100086. [CrossRef]
2. Perdana, A.; Jhee Jiow, H. Crypto-Cognitive Exploitation: Integrating Cognitive, Social, and Technological perspectives on cryptocurrency fraud. *Telemat. Inform.* **2024**, *95*, 102191. [CrossRef]
3. Chauhan, R.; Mehtar, K.; Kaur, H.; Alankar, B. Evaluating Cyber-Crime Using Machine Learning and AI Approach for Environmental Sustainability. In Proceedings of the Sustainable Development through Machine Learning, AI and IoT, Virtual Event, 27–28 April 2024; pp. 37–49. [CrossRef]
4. Taher, S.S.; Ameen, S.Y.; Ahmed, J.A. Advanced Fraud Detection in Blockchain Transactions: An Ensemble Learning and Explainable AI Approach. *Eng. Technol. Appl. Sci. Res.* **2024**, *14*, 12822–12830. [CrossRef]
5. Snigdha, K.; Reddy, P.S.N.; Hema, D.; Gayathri, S. BitPredict: End-to-End Context-Aware Detection of Anomalies in Bitcoin Transactions using Stack Model Network. In Proceedings of the 3rd International Conference on Advances in Computing, Communication and Applied Informatics, ACCAI 2024, Chennai, India, 9–10 May 2024. [CrossRef]
6. Dutta, S.; Sharma, A.; Rajgor, J. Ethereum Fraud Prevention: A Supervised Learning Approach for Fraudulent Account Recognition. In Proceedings of the 2024 1st International Conference on Trends in Engineering Systems and Technologies, ICTEST 2024, Kochi, India, 11–13 April 2024. [CrossRef]
7. Md, A.Q.; Narayanan, S.M.S.S.; Sabireen, H.; Sivaraman, A.K.; Tee, K.F. A novel approach to detect fraud in Ethereum transactions using stacking. *Expert Syst.* **2023**, *40*, e13255. [CrossRef]
8. Cunha, L.L.; Brito, M.A.; Oliveira, D.F.; Martins, A.P. Active Learning in the Detection of Anomalies in Cryptocurrency Transactions. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1717–1745. [CrossRef]
9. Ahmed, A.A.; Alabi, O.O. Secure and Scalable Blockchain-Based Federated Learning for Cryptocurrency Fraud Detection: A Systematic Review. *IEEE Access* **2024**, *12*, 102219–102241. [CrossRef]
10. Gürfidan, R. Suspicious transaction alert and blocking system for cryptocurrency exchanges in metaverse's social media universes: RG-guard. *Neural Comput. Appl.* **2024**, *36*, 18825–18840. [CrossRef]
11. Qasim Abdulkadhim, R.; Abdullah, H.S.; Hadi, M.J. Surveying the prediction of risks in cryptocurrency investments using recurrent neural networks. *Open Eng.* **2024**, *14*. [CrossRef]

12. Elmougy, Y.; Liu, L. Demystifying Fraudulent Transactions and Illicit Nodes in the Bitcoin Network for Financial Forensics. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Long Beach, CA, USA, 6–10 August 2023; pp. 3979–3990. [CrossRef]
13. Hisham, S.; Makhtar, M.; Aziz, A.A. Anomaly detection in smart contracts based on optimal relevance hybrid features analysis in the Ethereum blockchain employing ensemble learning. *Int. J. Adv. Technol. Eng. Explor.* **2023**, *10*, 1552–1579. [CrossRef]
14. Chen, Z.; Liu, S.Z.; Huang, J.; Xiu, Y.H.; Zhang, H.; Long, H.X. Ethereum Phishing Scam Detection Based on Data Augmentation Method and Hybrid Graph Neural Network Model. *Sensors* **2024**, *24*, 4022. [CrossRef]
15. Choi, S.H.; Buu, S.J. Learning to Traverse Cryptocurrency Transaction Graphs Based on Transformer Network for Phishing Scam Detection. *Electronics* **2024**, *13*, 1298. [CrossRef]
16. Ouyang, S.; Bai, Q.; Feng, H.; Hu, B. Bitcoin Money Laundering Detection via Subgraph Contrastive Learning. *Entropy* **2024**, *26*, 211. [CrossRef] [PubMed]
17. Kang, J.; Buu, S.J. Graph Anomaly Detection with Disentangled Prototypical Autoencoder for Phishing Scam Detection in Cryptocurrency Transactions. *IEEE Access* **2024**, *12*, 91075–91088. [CrossRef]
18. Hu, Z.; Dong, Y.; Wang, K.; Sun, Y. Heterogeneous Graph Transformer. *arXiv* **2020**, arXiv:2003.01332. [CrossRef]
19. Breunig, M.M.; Kriegel, H.P.; Ng, R.T.; Sander, J. LOF: Identifying density-based local outliers. In Proceedings of the SIGMOD '00, 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 15–18 May 2000; Association for Computing Machinery: New York, NY, USA, 2000; pp. 93–104. [CrossRef]
20. Directorate-General for Internal Policies of the Union (European Parliament); Snyers, A.; Houben, R. *Cryptocurrencies and Blockchain—Legal Context and Implications for Financial Crime, Money Laundering and Tax Evasion*; Technical Report; EU Publications: Luxembourg, 2018. [CrossRef]
21. Europol. *Cryptocurrencies—Tracing the Evolution of Criminal Finances*; Technical Report; Europol: The Hague, The Netherlands, 2021. [CrossRef]
22. Cholevas, C.; Angeli, E.; Sereti, Z.; Mavrikos, E.; Tsekouras, G.E. Anomaly Detection in Blockchain Networks Using Unsupervised Learning: A Survey. *Algorithms* **2024**, *17*, 201. [CrossRef]
23. Natoli, C.; Gramoli, V. The Blockchain Anomaly. In Proceedings of the 2016 IEEE 15th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 31 October–2 November 2016; pp. 310–317. [CrossRef]
24. Kircanski, A.; Tarvis, T. Coinbugs: Enumerating Common Blockchain Implementation-Level Vulnerabilities. *arXiv* **2021**, arXiv:2104.06540. [CrossRef]
25. Siddamsetti, S. Anomaly Detection in Blockchain Using Machine Learning. *J. Electr. Syst.* **2024**, *20*, 619–634. [CrossRef]
26. Apiecionek, L.; Karbowski, P. Fuzzy Neural Network for Detecting Anomalies in Blockchain Transactions. *Electronics* **2024**, *13*, 4646. [CrossRef]
27. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The Graph Neural Network Model. *IEEE Trans. Neural Netw.* **2009**, *20*, 61–80. [CrossRef] [PubMed]
28. Di, Z.; Wang, G.; Jia, L.; Chen, Z. Bitcoin transactions as a graph. *IET Blockchain* **2022**, *2*, 57–66. [CrossRef]
29. Tharani, J.S.; Charles, E.Y.A.; Hóu, Z.; Palaniswami, M.; Muthukkumarasamy, V. Graph Based Visualisation Techniques for Analysis of Blockchain Transactions. In Proceedings of the 2021 IEEE 46th Conference on Local Computer Networks (LCN), Edmonton, AB, Canada, 4–7 October 2021; pp. 427–430. [CrossRef]
30. Weber, M.; Domeniconi, G.; Chen, J.; Weidele, D.K.I.; Bellei, C.; Robinson, T.; Leiserson, C.E. Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics. *arXiv* **2019**, arXiv:1908.02591. [CrossRef]
31. Lin, D.; Wu, J.; Yuan, Q.; Zheng, Z. Modeling and Understanding Ethereum Transaction Records via a Complex Network Approach. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 2737–2741. [CrossRef]
32. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2017**, arXiv:1609.02907. [CrossRef]
33. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2018**, arXiv:1710.10903. [CrossRef]
34. Akcora, C.G.; Li, Y.; Gel, Y.R.; Kantarcioglu, M. BitcoinHeist: Topological Data Analysis for Ransomware Detection on the Bitcoin Blockchain. *arXiv* **2019**, arXiv:1906.07852. [CrossRef]
35. Wang, K.; Pang, J.; Chen, D.; Zhao, Y.; Huang, D.; Chen, C.; Han, W. A Large-scale Empirical Analysis of Ransomware Activities in Bitcoin. *ACM Trans. Web* **2021**, *16*, 1–29. [CrossRef]
36. Huang, D.Y.; Aliapoulios, M.M.; Li, V.G.; Invernizzi, L.; Bursztein, E.; McRoberts, K.; Levin, J.; Levchenko, K.; Snoeren, A.C.; McCoy, D. Tracking Ransomware End-to-end. In Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 20–24 May 2018; pp. 618–631. [CrossRef]
37. Hasan, M.; Rahman, M.S.; Janicke, H.; Sarker, I.H. Detecting anomalies in blockchain transactions using machine learning classifiers and explainability analysis. *Blockchain Res. Appl.* **2024**, *5*, 100207. [CrossRef]
38. Pham, T.; Lee, S. Anomaly Detection in Bitcoin Network Using Unsupervised Learning Methods. *arXiv* **2017**, arXiv:1611.03941. [CrossRef]
39. Shayegan, M.J.; Sabor, H.R.; Uddin, M.; Chen, C.L. A Collective Anomaly Detection Technique to Detect Crypto Wallet Frauds on Bitcoin Network. *Symmetry* **2022**, *14*, 328. [CrossRef]

40. Ding, K.; Li, J.; Bhanushali, R.; Liu, H. Deep Anomaly Detection on Attributed Networks. In Proceedings of the SIAM International Conference on Data Mining (SDM), Calgary, AB, Canada, 2–4 May 2019.
41. Bank, D.; Koenigstein, N.; Giryes, R. Autoencoders. *arXiv* **2021**, arXiv:2003.05991. [CrossRef]
42. Liu, F.T.; Ting, K.M.; Zhou, Z.H. Isolation Forest. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; ICDM'08; pp. 413–422. [CrossRef]
43. West, J.; Bhattacharya, M. Intelligent financial fraud detection: A comprehensive review. *Comput. Secur.* **2016**, *57*, 47–66. [CrossRef]
44. Zhu, D.; Ma, Y.; Liu, Y. Anomaly Detection with Deep Graph Autoencoders on Attributed Networks. In Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020; pp. 1–6. [CrossRef]
45. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762. [CrossRef]
46. K1SHIN. GAT Fraud Detection on Elliptic Dataset. 2021. Available online: https://www.kaggle.com/code/k1shin/gat-fraud-detection-illicit-f1-0-89 (accessed on 15 January 2025).

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

future internet

MDPI

Article

Dragon_Pi: IoT Side-Channel Power Data Intrusion Detection Dataset and Unsupervised Convolutional Autoencoder for Intrusion Detection

Dominic Lightbody *, Duc-Minh Ngo, Andriy Temko, Colin C. Murphy and Emanuel Popovici *

Electrical and Electronic Engineering, University College Cork, T12 K8AF Cork, Ireland;
120220051@umail.ucc.ie (D.-M.N.); atemko@ucc.ie (A.T.); colinmurphy@ucc.ie (C.C.M.)
* Correspondence: 121100122@umail.ucc.ie (D.L.); e.popovici@ucc.ie (E.P.)

Abstract: The growth of the Internet of Things (IoT) has led to a significant rise in cyber attacks and an expanded attack surface for the average consumer. In order to protect consumers and infrastructure, research into detecting malicious IoT activity must be of the highest priority. Security research in this area has two key issues: the lack of datasets for training artificial intelligence (AI)-based intrusion detection models and the fact that most existing datasets concentrate only on one type of network traffic. Thus, this study introduces Dragon_Pi, an intrusion detection dataset designed for IoT devices based on side-channel power consumption data. Dragon_Pi comprises a collection of normal and under-attack power consumption traces from separate testbeds featuring a DragonBoard 410c and a Raspberry Pi. Dragon_Slice is trained on this dataset; it is an unsupervised convolutional autoencoder (CAE) trained exclusively on held-out normal slices from Dragon_Pi for anomaly detection. The Dragon_Slice network has two iterations in this study. The original achieves 0.78 AUC without post-processing and 0.876 AUC with post-processing. A second iteration of Dragon_Slice, utilising dropout to further impede the CAE's ability to reconstruct anomalies, outperforms the original network with a raw AUC of 0.764 and a post-processed AUC of 0.89.

Keywords: IoT time series power dataset; IoT hardware security; IoT intrusion detection; convolutional autoencoder; unsupervised learning; cyber attacks on IoT; explainable AI

1. Introduction

The Internet of Things (IoT) has become a cornerstone of modern networks. the IoT, a term synonymous with the interconnectivity of "Things", has led to internet-facing devices flooding the markets in recent years, with a prediction of approximately 25 billion devices by 2030 [1]. This rapid expansion has led many companies to release their IoT product into the growing market as fast as possible. Security is often a second thought, or not considered at all for many of these devices, as it would only slow down production. As such, the market overflows with an ocean of insecure, unprotected and, in some cases, hazardous IoT devices [2]. Indeed, the expansive Internet of Things has vastly spread the attack surface for the average citizen, likely exposing them in ways they do not even consider.

For example, in 2021, the cloud-based security camera service Verkada was breached after malicious actors discovered viable admin credentials on the internet, exposing the live feeds of over 150,000 cameras worldwide [3]. These cameras were located in hospitals, classrooms, jails, factories and many other sensitive locations. Poor user privilege hygiene emerged as the leading cause of the breach, with over 100 Verkada personnel having 'super admin' permissions. Even without the breach, this was a significant infringement of customers' privacy as these super admins had access to clients' live feeds, likely without the customers' knowledge.

In the same year, critical national infrastructure was attacked in Florida, USA. This time, the target was a city's water treatment facility [3]. Malicious actors accessed a plant employee's computer remotely and, using the applications on this PC, adjusted the water's sodium hydroxide (NaOH) levels by a factor of 100. This might have proven to be one of history's most deadly cyber attacks, if not for swift intervention from the facility, readjusting the chemical levels. This attack illustrates the potential danger of outdated systems, which most critical national infrastructures rely upon, and the danger of omitting safety systems from critical infrastructure. If a safety system (for example, requiring two operators to authorise the change in chemical concentrations far outside normal specifications or the swiping of a physical key card to authorise the operation) was used, it would have made it much harder for attackers.

Due to the unique, and often resource-constrained nature of the IoT ecosystem, traditional security practices cannot be easily applied. Extensive research has been conducted in signature-based intrusion detection systems (IDSs) and anomaly detection-based IDSs for IoT devices to detect cyber attacks. Typically, signature-based IDS relies on training with well-known normal and attack signatures, exhibiting high performance on previously encountered attacks but limited capability to handle unpredictable or unseen zero-day attacks. On the other hand, anomaly-based detectors operate by identifying deviations from the baseline, usually trained solely on normal behaviour. They excel at detecting unseen attacks but may not perform as well as signature-based approaches with well-known attacks.

Parallel to the growth of IoT in recent years, artificial intelligence (AI) has seen substantial growth in activity and use, especially in security. Fully supervised AI approaches have been deployed for signature-based detection in IDSs and, more recently, semi-supervised and unsupervised approaches have been deployed for anomaly detection in IoT. These recent approaches will be reviewed in the related works Section 2.

One main limiting factor disadvantaging security operators is the lack of training data. This training data can be considered both from the perspective of training humans on attack patterns to understand and mitigate them and training automated AI-based detection methods. In our previous work [4], we discussed IoT security from a somewhat unconventional point of view, focusing on device power consumption (as a linear time series) rather than internet packets. In that work, we attempted to illustrate that studying these signatures from a human perspective is an invaluable asset for security teams in understanding how the attacks work and how to defend against them. Importantly, we showed that device power, while not as readily available as internet traffic, should not be overlooked in security. To this end, we adopt some key concepts from popular side-channel attack methodologies, e.g., simple power analysis (SPA) and differential power analysis (DPA) [5], to study the power signatures for use in security. The more data sources and defensible positions we consider in security research, the more we can close the gap between attacker and defender.

To this end, we present Dragon_Pi, which is an IoT power dataset containing labelled normal and attack scenarios captured from two of the most popular IoT platforms (Raspberry Pi and Snapdragon). We also present Dragon_Slice, which is the first attempt at designing an automated IDS based on these data. Dragon_Pi fills a fraction of the massive gap in IoT security. To the best of the authors' knowledge, Dragon_Pi and Dragon_Slice, respectively, constitute the first publicly available IoT security power dataset and unsupervised convolutional autoencoder based on IoT power consumption. Dragon_Pi is the first step of many in supplementing traditional security approaches to IoT security using side-channel data, such as power consumption. We invite others to repeat the process and generate datasets of a similar nature, exploring IoT security from a different perspective.

1.1. Contributions

The Dragon_Pi, Dragon_Slice framework extends our previous work [6]. Indeed, this work can be considered as part two of our recent work [4], where we discuss and analyse the signatures of both normal and attack behaviours within the Dragon_Pi dataset. This

recent work seeks to illustrate the value of power data in cybersecurity and discuss possible usages. Dragon_Pi and Dragon_Slice seek to fulfil some of the possible works that we said were possible based on the data in [4,6]. The specific contributions are as follows:

1. Dragon_Pi: The first contribution is a fully labelled device power behaviour dataset containing normal and attack scenarios from two popular IoT devices. Dragon_Pi explores various attack scenarios and, to the authors' knowledge, is the only publically available IoT Security power dataset. Dragon_Pi focuses on IoT devices, and the insights gained from studying this dataset can be applied to the entire cybersecurity field. Dragon_Pi serves as a step forward for IoT security, encouraging security professionals to consider as many data sources as possible rather than focus on internet traffic.
2. Dragon_Slice: The second contribution is Dragon_Slice for anomaly detection within linear time series. Importantly, Dragon_Slice is a single model trained only on the normal power behaviours of both the DragonBoard and Raspberry Pi. Using the Dragon_Slice framework, a single IDS model can be trained on the normal power consumption data of different devices. This model can then detect unseen anomalies across the different devices. This is a massive contribution to the field of IoT, which is filled with heterogeneous devices. In this way, a single model can serve as many devices as desired, rather than requiring specifically trained models per device type.
3. Explainability: The third contribution involves explaining the AI used in this paper. To this end, we explain Dragon_Slice from the perspective of inputs and outputs, and, thus, why the AI is making particular decisions. In AI research, many works under-explain the presented AI. This hinders readers from effectively understanding what the AI is doing or from reconstructing similar AI for themselves. Another key issue with under-explained AI is the lack of verification of the AI functionality. Readers can only confirm the plausibility of the presented results if they are able to understand the AI demonstrated.

1.2. Organisation

The remainder of this work is organised as follows. Section 2 surveys some of the most popular IoT security datasets and some recent AI applied to those datasets. Section 3 discusses the over-arching framework and a brief overview of the dataset itself. Section 4 describes the Dragon_Slice architecture and training methodology as well as the results from the two paths taken for the AI. Section 5 discusses some of the key findings of the study and further explains some aspects of the architecture. Limitations of the study are discussed in Section 6. The conclusions and possible further works are presented in Section 7 .

2. Related Works

As IoT Security is an emerging and essential field, much work has been conducted to create automated security systems (such as IDSs) for them. Much of this research has used AI to secure these IoT systems. AI is a pervasive paradigm within most recent technology fields and has seen great success in IoT security and the wider security field.

Before any AI algorithm can be trained, there must be data available to train and test upon. As such, a large and vital portion of IoT security research has involved creating and labelling datasets of IoT attack data. The data must be realistic, relevant, and of good quality.

Much surveying has been conducted in the field of IoT intrusion detection [7–9], in particular, presented extensive research on a wide range of algorithms used by researchers and summarised the popularity of well-known datasets in the reviewed work. Importantly, they realised that most of the works they surveyed used simulated or outdated datasets. This clarifies the need for as many new and up-to-date datasets as possible in cybersecurity and IoT security. The overuse of outdated datasets has led to research results, which may

not translate to acceptable performance in the modern cyber threat topography. Table 1 lists some well-known datasets chronologically, along with their proportionate usage in [9].

In [9], the oldest work they reference is from 2016; when consulting Table 1, we can see a disproportionate percentage of papers using outdated datasets. The fact that researchers must rely on such outdated data illustrates the distinct lack of data in the field of IoT cybersecurity and, thus, motivates the creation of the Dragon_Pi dataset presented in this work. By consulting Table 1, it is also clear that the overwhelming majority of the dataset work conducted on IoT focuses entirely on network packets. While this makes sense, since cybersecurity has been dominated by network-based security for decades, it is imperative to consider security from as many points of view as possible in the modern cyber threat topography.

Table 1. Popular IoT datasets and recent works.

Dataset	Type of Data	Year Published	Recent Studies on This Dataset	Percentage Usage Observed in [9]
KDD CUP 99 [10,11]	Network Packets	1999	[12,13]	9%
NSL-KDD [14]	Network Packets	2009	[15–18]	12%
UNSW15-NB15 [19]	Network Packets	2015	[20–22]	N/A
N-BaIoT [23]	Network Packets	2018	[24–27]	9%
ToN-IoT [28]	Telemetry data of IoT Services and Network Packets	2020	[29,30]	2%
IoT 23 [31]	Network Packets	2020	[32–34]	N/A

Derived from the 1998 Defence Advanced Research Project Agency (DARPA) dataset [35], the Knowledge Discovery and Data Mining (KDD) 1999 dataset was created to be used in the development of security-focused ML [10,11]. In essence, the KDD dataset transforms the DARPA dataset's network traces into a collection of features found in the connection between two hosts, with each connection constituting its own record in the dataset. These features are broken into basic (e.g., duration), content (e.g., logged_in), time-based (e.g., srv_count), and connection-based (e.g., dst_host_count) categories.

Due to its acceptance as a de facto benchmark for cybersecurity research, many research works today are still conducted using the KDD dataset, regardless of its age and, indeed, its outdatedness. In 2019, ref. [13] applied three different artificial neural network (ANN) approaches to the KDD dataset. Utilising learning vector quantization (LVQ), the radial basis function (RBF), and multilayer perceptron (MLP), they achieved accuracies of 97.5%, 99.4%, and 99.86%, respectively. Notably, however, the presumably inference time performance of the algorithms was compared, and it was found that LVQs were far more time efficient, requiring only 4.56 s for the best-performing model, with time requirements of 22,610 s for RBFs and 28,811 s for MLP s.

In 2022, ref. [12] compared K-nearest neighbour (K-NN), support vector machines (SVMs), and a novel hybrid approach utilising the SNORT IDS [36] and a naïve Bayes (NB) algorithm on the KDD 99 dataset. In this hybrid approach, traffic is first analysed by the SNORT IDS and, if SNORT classifies the traffic as normal, the NB algorithm is applied to the traffic to confirm the result. The work achieved results of 93% for SVMs, 94.4% for K-NNs, and 96.6% for the proposed SNORT-NB hybrid model.

Given the infancy of networking security at the time, the KDD dataset naturally leaves much to be desired, being one of the first of its type in the field. In the decades since its release, the dataset has garnered much criticism; for example, in [14], the authors present the argument that excess redundant and duplicate records are some critical shortfalls of the KDD dataset. In 2009, they created NSL-KDD, a shorter and more curated version of the KDD dataset. They achieved this by removing duplicate records or, more precisely, records that seemed like duplicates, which were generated during the creation process of the KDD dataset. Notably, in the creation process of the KDD dataset, the start time of connection, source, and destination IP addresses and source and destination ports were stripped from

the internet packets. Thus, many traffic flows, such as ICMP, resulted in what appeared to be duplicated and redundant traffic flows, as they were missing these five key pieces of information that could distinguish them.

In 2016, ref. [17] proposed a hybrid approach to IoT security. A proposed framework based on the MapReduce architecture was developed with the goal of distributed detection capabilities. Importantly, this framework considered IoT security from two key perspectives: internet traffic and wireless sensor network (WSN) traffic. The proposed framework featured both misuse-based and anomaly-based methodologies, utilising optimum path forest models. Based on the RPL routing protocol, a WSN was created to simulate the 6LoWPAN traffic. For the internet traffic aspect of the framework, the NSL-KDD dataset was implemented. The proposed hybrid approach achieved detection rates (DRs) of 80.95% and 96.20% for the anomaly detection and misuse-based modules, respectively. These modules also achieved false alarm rates (FARs) of 5.92% and 1.44%, respectively. Notably, the misuse-based model focused on internet traffic, while the anomaly detection model focused on the generated WSN traffic.

In 2017, ref. [16] proposed an unsupervised conditional variational autoencoder (CVAE) IoT network-IDS (NIDS) trained on the NSL-KDD dataset. The architecture integrates intrusion labels within the decoder layers, and the approach boasts better performance than other unsupervised approaches, with a multi-class accuracy of 80.10%, all while being less complex. An essential aspect of the proposed architecture is the ability to reconstruct features from incomplete training sets, in particular, the protocol, flag, and service features, with accuracies of 99.02%, 92.84% and 71.42%, respectively. This is an important feature for the KDD and NSL-KDD datasets, which lost much information when the KDD dataset was created from the DARPA dataset. This framework introduced the first algorithm capable of performing feature recovery while also being the first work to present a CVAE in network intrusion detection.

To combat the lack of comprehensive network-based datasets, UNSW-NB15 [19] was created. Previous benchmarks, DARPA98, KDD99, and NSL-KDD09, had become significantly outdated. Thus, UNSW-NB15 was generated to provide a dataset representing modern network traffic, information regarding the depth structure of the network, and minimal footprint intrusion. UNSW-NB15 contains real normal network traffic mixed with synthesised attack traffic.

In 2021, ref. [22] proposed an edge-centric NIDS utilising deep neural networks, particularly long short-term memory (LSTM) models and gated recurrent units (GRUs), with the specific goal of DDoS detection in IoT. The model was trained and evaluated on the UNSW-NB15 dataset, achieving a high accuracy of 99%. The model proposed in the framework is lightweight and requires few resources, allowing for implementation in resource-constrained environments, such as the IoT. In [22], the proposed framework models were deployed on Raspberry Pi, and their resource usage was compared to find the most suitable model for IoT, balancing accuracy with resource usage. From this comparison, it was found that the FastGRNN (fast-gated recurrent neural network) was the most suitable, with an accuracy of 99.95% and reasonably small resource requirements and inference times.

In 2022, ref. [20] presented an IDS for IoT trained on the UNSW-NB15 dataset. The proposed framework utilised particle swarm optimisation-based gradient descent (PSO-LightGBM) for feature extraction. These features were then fed into a one-class SVM (OC-SVM) for classification between normal and anomalous. Evaluating the UNSW-NB15 dataset, the framework achieved an accuracy of 86.68%. The framework was specifically robust at detecting small sample data within UNSW-NB15, such as Shellcode, Worms, and Backdoor.

To mitigate the threat caused by compromised IoT devices, as well as develop a technique that can differentiate between hours-long and seconds-long attacks, the N-BaIoT [23] model, and the resulting dataset, were created in 2018. N-BaIoT is an IoT NIDS framework that can extract snapshots from network traffic behaviour of interest and detect compromised IoT devices using deep autoencoders. To generate the data for training and testing,

testbeds consisting of commercially available IoT devices were infected with the Mirai and BASHLITE IoT-based botnets. The proposed methodology had a true positive rate (TPR) of 100% detecting attacks launched from a compromised IoT device with a low false positive rate (FPR) of 0.007% and only between 174 ms and 212 ms to detect an attack.

In 2023, ref. [27] proposed a semi-supervised AE approach to the N-BaIoT dataset. To cope with the scalability of IoT, the proposed framework allows for the training of a single model, which can work across many devices. The N-BaIoT dataset was split into multiple datasets based on the separate IoT devices in the testbed. The Deep AE was then trained on only the normal traffic flows from separate IoT devices, with the malicious traffic constituting separate test sets based on different IoT devices. The proposed approach achieved high performance with an all-in-one F1 score of 0.9996–0.9998, depending on the specific IoT device.

Thus far, recent decades have brought very few IoT-based intrusion detection benchmark datasets. The datasets released all have the common theme of capturing normal and malicious network traffic only, without any new data vectors included for cybersecurity research. To combat the lack of available benchmarks and provide a new avenue of data to explore for data-driven IDS, the ToN_IoT [28] telemetry dataset was developed and released in 2020. ToN_IoT includes not only the network traffic of the IoT network but also the operating system logs and telemetry data of IoT services. ToN_IoT has an advantage over other datasets in the area as it contains IoT telemetry data and diverse attack scenarios, a representative and realistic testbed with IoT scenarios, labels for the data, and heterogeneity of IoT data sources. Many machine learning approaches were explored, such as logistic regression (LR), linear discriminant analysis (LDA), K-nearest neighbour (k-NN), classification and regression tree (CART), random forest (RF), naïve Bayes (NB), support vector machines (SVMs), and long short-term memory (LSTM). The machine learning algorithms first were evaluated on specific device datasets, i.e., fridge sensor (best performance: LSTM with 100% accuracy), garage door (best performance: all methodologies with 100% accuracy), GPS sensor (best performance: k-NN with 88% accuracy), Modbus (best performance: CART with 98% accuracy), Light_Motion (best performance: all approximately 58% accuracy), thermostat (best performance: all approximately 66% accuracy) and, finally, weather (best performance: CART with 100% accuracy). Combining all individual IoT device datasets into one, the CART approach emerged as dominant with an overall accuracy of 88% for binary classification and 77% for multi-class classification.

As part of the Avast AIC lab, the IoT-23 [31] dataset was published in 2020. The dataset contains labelled traffic flows of benign and malicious behaviour captured between 2018 and 2019. IoT-23, named because it consists of 23 separate scenarios, contains over 300 million labelled records from sixteen categories, each record having eighteen features. These features are divided into static, basic, content, and time.

In 2021, ref. [33] described designing and developing a convolutional neural network (CNN) for IoT network anomaly detection. Using the IoT-23 [31] dataset, the multi-class CNN model was initially evaluated. CNNs using 1D, 2D, and 3D convolutions were then evaluated. Then, using transfer learning, a binary classifier CNN was trained from these initial multi-class models. In this use case, transfer learning from multi to binary classification proved effective as the binary classifier was trained on a subset of the data used to train the multi-class classifier. Over the datasets mentioned above, both the binary and multi-class classifiers achieved average accuracies of over 99% for 1D, 2D, and 3D CNN models.

In particular, in the field of cyber security, and as witnessed in the majority of datasets and the works reviewed above, security datasets tend to be focussed on one data vector: network traffic. This can easily be justified as the main avenue to explore in cybersecurity, as the network tends to be the key ingress point for most malicious activity. But even if network traffic is the main defence concern, exploring other data types within the security spectrum is also important. One main concern for most authors is the distinct lack of IoT-based internet traffic datasets—if datasets are sparse in the most explored area in IoT

security, how much more sparsity is there in other data types? TON_IoT [28] is one work that seeks to bring new data types to the cybersecurity dataset repertoire; however, the field remains completely unsaturated in all forms of data. As such, the work presented in this Dragon_Pi study calls for more datasets in the field. Only by exploring as many avenues for security as possible will defence personnel be empowered to catch up to the ever-adapting malicious actors.

Table 2 summarises the performance of the reviewed works on the reviewed datasets.

Table 2. Performance of the reviewed works.

Dataset	Publication	Accuracy
KDD CUP 99 [10,11]	[13]	97.5% (LVQ), 99.4% (RBF), 99.86% (MLP)
	[12]	93% (SVM), 94.4% (K-NN), 96.6% (SNORT-NB)
NSL-KDD [14]	[16]	80.1% (CVAE)
UNSW15-NB15 [19]	[22]	99% (LSTM, GRU), 99.95% (FastGRNN)
	[20]	86.68% (PSO-LightGBM & OC-SVM)
ToN-IoT [28]	[28]	88% (Binary Class CART), 77% (Multi-Class CART)
IoT-23 [31]	[33]	99% (Binary and Multi-Class 1D, 2D and 3D CNNs)
		Detection Rate (DR) and False Alarm Rate (FAR)
NSL-KDD [14]	[17]	80.95% DR (Anomaly Based MapReduce), 96.20% DR (Misuse Based MapReduce), 5.92% FAR (Anomaly Based MapReduce), 1.44% FAR (Misuse Based MapReduce)
		True Positive Rate (TPR) False Positive Rate (FPR)
N-BaIoT [23]	[23]	100% (TPR Deep-AE), 0.007% (FPR Deep-AE)
		F1-Score
N-BaIoT [23]	[27]	0.9996–0.9998 (Deep-AE)
		Lost Feature Reconstruction Accuracy
NSL-KDD [14]	[16]	99.02% (Protocol CVAE), 92.84% (Flag CVAE), 71.89% (Services CVAE)

Regarding alternative data types, some research has been conducted into anomaly detection using various types of time series data.

In [37], a reconstruction-based approach to anomaly detection in the linear time series was taken. Autoencoder-based models were utilised in the anomaly detection process, which is similar to the research presented in this work. Rather than fine-tune hyperparameters by hand, a neural architecture search technique was employed to find the best model configuration. A custom dataset that included time series features, such as temperature and luminance, CO_2 concentration, power consumption, and humidity, was used to train the AE models. Notably, the authors had no access to an anomaly detection dataset during testing; as such, synthetic anomalies were injected into the normal behaviour of the custom dataset during testing.

Generally, deep learning requires a vast amount of labelled data to train a model effectively. When working with real-world times series, the data collection is thought to be easy; the annotation of data, however, is time-consuming and relatively expensive. The works surveyed in [38] demonstrate a solution to this problem through deep transfer learning. In essence, deep transfer learning allows for the training of a model on a far more extensive or diverse dataset, even if it is unrelated to the current problem. This pre-trained model gains far more general features from the larger datasets, which can be transferred effectively to the target problem during the fine-tuning stage on the much smaller and more limited dataset.

Another concern stemming from the lack of data in cybersecurity is the performance creep of many models on these well-known benchmark datasets. Part of the survey performed in this work is to illustrate that, even in recent years, studies are still being conducted on decades-old and obsolete datasets, likely due to the lack of available data.

With datasets so over-explored, researchers may feel the need to set themselves, attempting techniques that may end up negating the validity of their model, such as the biased selection of samples for training and testing as a subset of an overall dataset. Far too many models are achieving results of 99% and 100% on datasets. These results are unlikely to transfer to real-life scenarios and should strongly indicate that either the model or dataset is lacking (either with models trained on the path of least resistance features, or datasets far too simple for the ever-adapting cyber landscape).

There is a clear need for more datasets as well as datasets of varying data types. This need guides the research that follows in this study. Dragon_Pi and Dragon_Slice represent initial steps to explore the effectiveness of power consumption data in security. The insights gained during this study will guide future efforts to incorporate heterogeneous data sources into the security pipeline.

3. Materials and Methods

3.1. Testbed, Data Acquisition, and Framework

For the testbed, two different IoT devices were chosen: Raspberry Pi 3 Model B and DragonBoard 410. DragonBoard, powered by a Qualcomm Snapdragon 410 processor, has gained widespread popularity, boasting over 200 million units shipped. Similarly, Raspberry Pi, a prevalent single-board computer, has seen a significant market presence with over 40 million units sold over its ten-year history.

Despite variations in manufacturers and typical applications, both devices surprisingly share almost identical specifications, possessing Quad Core 1.2 GHz 64-bit CPUs and 1.2 GB of RAM; this deliberate selection aims to provide a thorough understanding of common attack signatures across devices of different manufacturers, software, etc. Our previous study [4] investigated whether these signatures exhibit general patterns, possibly simplifying the detection and understanding of attacks on IoT devices and heterogeneous networks. The previous work justifies the choice of devices and further explains other aspects of the device operation and testbed composition.

Both boards were connected to the internet via Wi-Fi, and SSH was enabled to broaden the potential attack surface. Bluetooth functionality enabled communication with a TI CC2650STK SensorTag, equipped with ten low-power MEMS sensors. A Python script on the testbed requested sensor data over Bluetooth, and DragonBoard and Raspberry Pi were linked to an Agilent Technologies N6705A DC Power Analyzer to unobtrusively monitor side-channel IoT power behaviour.

The framework, outlined in Figure 1, proposes a generic data acquisition framework for IoT. This framework was initially proposed in our previous work [4]. In addition to the previous framework, Figure 1 showcases the focus of this current study, emphasising AI-based anomaly detection in IoT security based on the power acquisition framework. In this framework, power consumption data serve as input to the AI security system, returning a classification of either normal or anomalous for each power data segment.

The convolutional autoencoder (CAE) architecture employed in this study operates on the principle of reconstruction, where the AI attempts to reconstruct the input based on what it was trained on. In the case of the framework in this study, the AI is trained only on "clean-room" normal data obtained from the testbed. In this case, "clean room" describes a situation where the normal data were sampled and held aside, almost like it was quarantined in a different room. As such, the training set and test set are mutually distinct, and the training and test sets are two separate datasets, with one sampled in a quarantined environment with no anomalies and one with more real-world events and attacks.

During testing, real-world normal and attack data are passed to the AI, which attempts to reconstruct these inputs. If trained properly, a CAE should be able to reconstruct normal unseen data well, as they are similar to the training data. However, this methodology should struggle to reconstruct the anomalous attack data, which looks different from the normal baseline. The quality of the reconstruction and the resultant mean squared error

(MSE) lead to a normal or anomaly classification (with low error meaning normal, and high error meaning anomaly).

This framework results in quite explainable AI, where the reconstructed output provides insights into what the AI perceives as normal or anomalous within the input signal, aiding in understanding the factors influencing its decision. This can significantly aid security experts in further identifying an attack or, possibly, a false positive from the AI output. Further analysing these factors would allow human intervention within the AI loop, or hierarchical AI, to oversee the output of many models. Human experts can assess both the input and output, making crucial decisions regarding security procedures. While the system can function autonomously, human oversight is vital as certain decisions can only be partially entrusted to AI. For example, AI can detect anomalous activity on a critical system, such as radar or health monitoring equipment. However, only a human can be responsible for making the call to shut down that device.

In hierarchical approaches, federated models could greatly enhance the system's capabilities. Individual models could operate per device, or IoT gateway, overseen by a hierarchical AI that could supervise the models' outputs, making predictions over many AI model outputs. This would enable security teams to make more informed decisions for individual devices or even families of devices, thereby facilitating the identification of individual devices or device groups under attack, predicting potential targets, or tracing the propagation vector of an attack through a network. Such insights empower security teams to make more informed decisions during an attack, swiftly isolating affected devices without shutting down an entire network topology.

This framework could extend beyond IoT devices to encompass every level of computing. Any system exhibiting a baseline normal operation could benefit from this framework, including internal servers, workstations, high-performance computers, manufacturing devices, robotics, and supervisory control and data acquisition (SCADA) systems.

Attacks can be very intensive on a system; even if the normal information is very noisy, training an AI framework, such as the CAE featured in this study, should make it simple to detect large-amplitude anomalies. For example, denial of service (DOS) and distributed DOS (DDOS) attacks could be easily detected using AI trained on the server's, router's, or gateway's normal baseline power consumption. This type of attack has maximum detectability since it exhibits such large power consumption signatures. This detection capability is enhanced by the increased availability of resources at these levels, which can allow more complex models to be deployed.

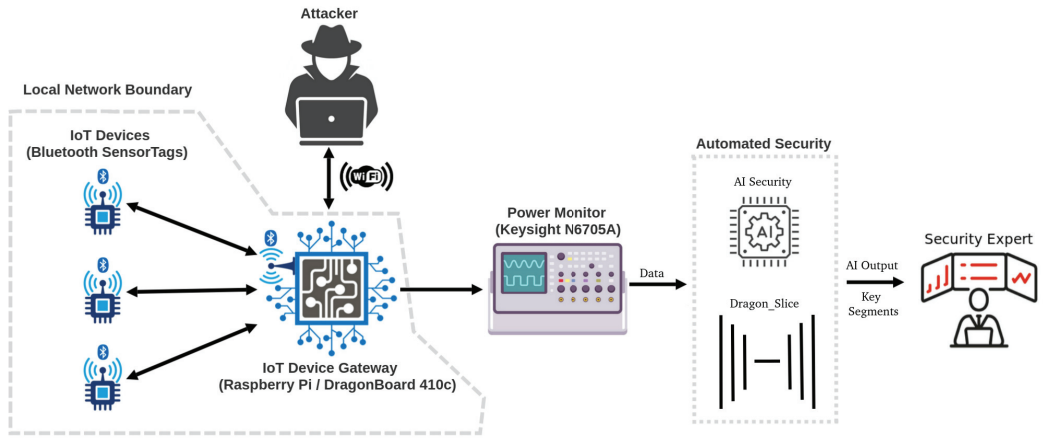


Figure 1. IoT power data acquisition and anomaly detection framework.

3.2. Dataset Overview

In this work, the test environment faced three distinct attack scenario topics, each encompassing a fundamental security concept related to the CIA triad [39]: confidentiality, integrity, and availability. Specifically, these attack scenarios are of three main types:

reconnaissance, brute force, and denial of service. Our previous work [4] provides a more detailed exploration of these attacks.

The dataset is comprised of 5 normal files and 65 attack files. Each file contains the linear time series power consumption of the testbed, either under normal or attack conditions. All data were sampled at a rate of 488,28 samples per second. This was the maximum rate the power analyser used in this study could achieve with the specific settings. The normal data were collected in a controlled environment, ensuring complete normalcy without any anomalies. This becomes crucial when considering using these five normal files as the training class for a semi-supervised or unsupervised approach. The remaining 65 files can then serve as an entirely unseen test set. The 65 under-attack files, in general, contain 50% normal and 50% attack signatures. This makes them perfect as balanced and unseen test sets.

Each file contains three levels of annotation specificity, a simple normal/anomaly annotation for each signature within the file, the specific attack type and, finally, the specific tools and settings used for that specific signature. Each signature in each file is labelled individually and precisely, rather than a label spanning an entire file.

From the topic of reconnaissance attacks, the port scan attack was applied. In this case, the tool Nmap was used to probe the testbed for open ports. This process was repeated many times to give as many attack signatures as possible. The attack was also repeated for Nmap tool settings "T2–T5" to give a range of signatures from different tool settings. The "T" setting for the Nmap tool adjusts the timing and performance of the attack. Increasing this value increases how aggressive the attack is. Each Nmap setting has its own file, as shown in Table 3.

For the brute force attacks, the SSH brute force attack was applied. The Hydra tool was used to perform the attack on the testbed. Like the port scan attacks, the brute force attacks were repeated for different tool settings. In this case, the adjustment of the Hydra "t" option between "1–32" controls the number of connections in parallel to the target. The smaller this value, the smaller the resultant signature, but with a reduced password attempt frequency. For example, in the files pertaining to Hydra "t1", these resultant signatures are each for a single password attempt, while in files related to Hydra "t32", the resultant signatures are (simultaneously) for 32 password attempts. Importantly, for the brute forcing related files in the dataset, half of the files will be named "IKE" and the other "CKE". IKE, in this case, refers to an incorrect key event; in other words, the password was not found while brute-forcing the testbed. In the CKE files, the password was included in the Wordlist used to brute force the testbed. Thus, the CKE stands for "Correct Key Event" and refers to a little signature after the brute force attempt, resulting in the correct password being attempted. This concept is discussed in much greater detail in our previous work [4]. Each Hydra setting has its own file, as shown in Table 3.

Using the Hping3 tool, the SYNFlood denial of service attack was applied to the testbed. There was no repeating of the attack or variation in the tool settings as the resultant signatures for the tool were as expected (large amplitude, etc.) and were not as intricate as the other signatures. In the Raspberry Pi SYNFlood DOS file, the attack seems to die midway. However, it was impossible to discern if the attack had actually stopped and if the testbed had returned to normal behaviour. As such, the entire period when the tool was attacking the testbed was labelled as an anomaly. This may lead to some false negatives when training machine learning for this specific file.

The final file type involves capture the flag (CTF) scenarios. These files utilise the reconnaissance and brute force topics from before in a more realistic scenario. In the CTF scenarios, rather than repeating the attacks to simply view the signatures, the attacks are used in a more practical and realistic way to infiltrate the testbed and exfiltrate a hidden file. The generic process of one of the CTF attacks would be as follows: ping the testbed, conduct a port scan attack to identify a potentially open port for the attack, launch a brute force attack to gain entry, SSH login with the stolen credentials, explore the testbed by viewing files remotely (using Linux commands on the device), exfiltrate the hidden file when found,

and finally withdraw from the testbed. In some of the CTF scenarios, a file is edited on the device to simulate the attacker affecting the integrity of the attacked system. In other CTF scenarios, a payload is uploaded to the device, which, when executed, performs intensive memory operations, simulating a methodology the attacker may take to drain the battery of the device after they exfiltrate the important information.

Each file in the dataset is accompanied by a legend file detailing what signatures to expect and the exact indexes of those signatures. This can aid in efficiently finding specific attack signatures.

The attack data files typically contain equal portions of normal and anomalous information. For instance, in a 5 min file, 2.5 min may contain anomalous data, while the remaining 2.5 min may consist of normal data interspersed around anomalies.

The data are tightly labelled, with each anomalous instance within a file labelled to the best of our ability. Figure 2 depicts the accuracy of labelling in this dataset. Specifically, the figure illustrates a 1-second data window with a port scan attack in the middle, showcasing the tight labelling, which is approximately 0.6 s long. These precise labels are crucial for dataset users to pinpoint the specific anomalous segment in each signal.

Figure 3 displays an example header for one of the data frames from the dataset, featuring multiple levels of annotation. "anno_string" provides a binary classification annotation (normal or anomaly) for each sample. "anno_type" indicates the overall type of anomaly, such as port scan or brute force. "anno_specific" provides an exact annotation derived from the attack tool and settings that produced the signature. For example, it includes instances such as "Hydra_T16", representing an SSH brute force signature observed when the Hydra tool was employed with T16 settings to target the testbed.

Table 3 enumerates the dataset contents, with each file categorised by the showcased anomaly. Generally, each file maintains a balanced annotation with a 50/50 distribution of anomalous to normal behaviour (excluding files consisting solely of normal data).

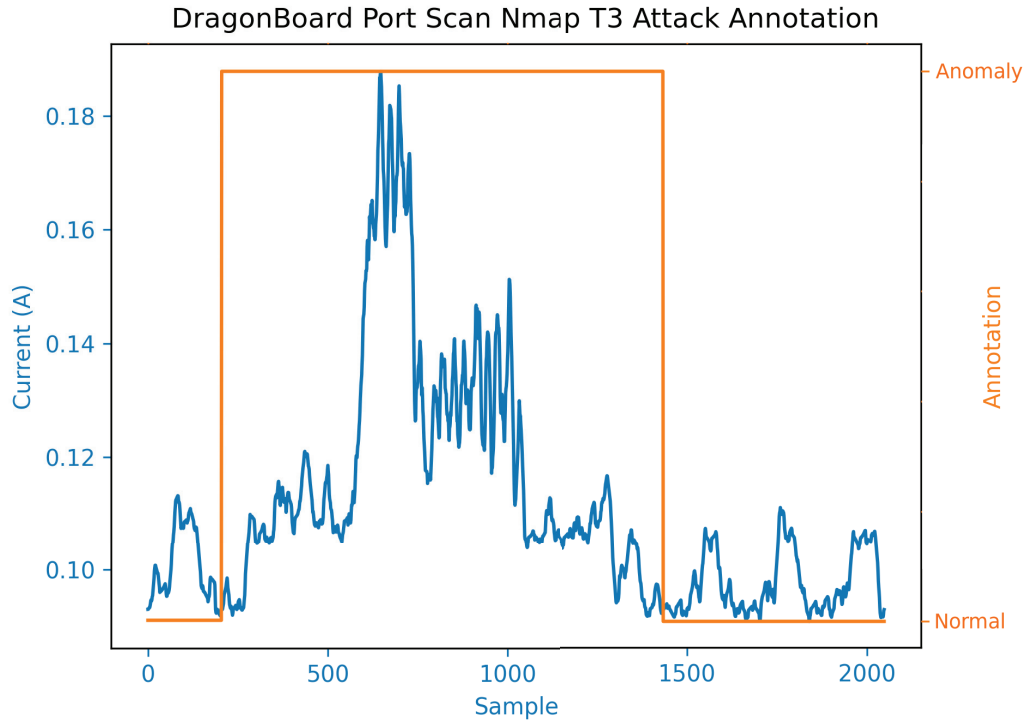


Figure 2. Example of annotation accuracy from the dataset.

Time	Current	anno_string	anno_type	anno_specific
28.799877	0.299654	Normal	Normal	Normal
28.799898	0.298771	Normal	Normal	Normal
28.799918	0.298418	Anomaly	SSH_Bruteforce	Hydra_T16
28.799939	0.298683	Anomaly	SSH_Bruteforce	Hydra_T16
28.799959	0.299360	Anomaly	SSH_Bruteforce	Hydra_T16

Figure 3. Example data frame header from the dataset.

Table 3. Enumeration of files in the Dragon_Pi dataset.

Label Type	Specific Label	# Files DragonBoard	# Files Raspberry Pi
Normal	Normal	3	2
Port Scan Attack	Nmap_T5	2	1
	Nmap_T4	1	1
	Nmap_T3	1	1
	Nmap_T2	1	1
SSH Brute Force	Hydra_T32	4	2
	Hydra_T16	16	2
	Hydra_T4	8	2
	Hydra_T1	5	2
SYNFlood DOS	SYNFlood DOS	1	1
Capture the Flag	Misc Attacks	3	5

4. Methodology

Figure 4 illustrates the machine learning pipeline featured in this work. Take note of the two separate datasets for training and testing the machine learning model, rather than the test dataset being a subset of the training dataset. Both the training and test data underwent the same preprocessing. Importantly, the training and, thus, testing and post-processing stages were repeated for compressed regions of sizes 1024, 512, and 256. The contents of Figure 4 will be discussed in far greater detail in the respective sections below.

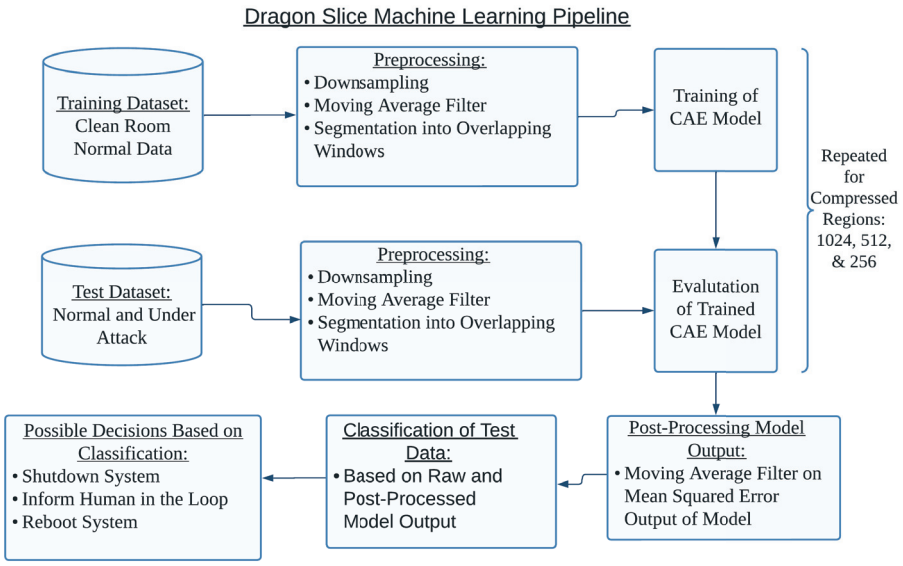


Figure 4. UML diagram of the machine learning pipeline featured in this work.

4.1. Data Preprocessing

In our previous study [4], there was no need to preprocess the data. However, some minor preprocessing was required to train an algorithm effectively with these data as input here.

Before preprocessing, the raw data were sampled from the testbed at 48,828 samples per second (SPS), which is the maximum sampling rate of the power analyser used in this study. This is a relatively high data sampling rate and tends to require an expensive piece of equipment. However, in this study, preprocessing reveals that one can obtain excellent results with a much lower sampling rate and, thus, significantly less expensive equipment.

Before any preprocessing, it was essential to ascertain the input length of the algorithm in real-time (in other words, how many seconds of data the algorithm would process). This epoch was determined by studying the normal data, and judging how long it took for a sufficient amount of normal data to accrue in a window to make an informed decision. This epoch was determined to be exactly one second for this work. As visible in Figure 5, the normal Bluetooth signature fits perfectly into a window of one second. Although one could use a slightly smaller window, perhaps half a second, this study urges caution against using too little input data.

It is essential to provide the algorithm (in this case, a convolutional autoencoder (CAE)) with sufficient data to learn complex patterns and useful global features. Presenting the algorithm with a window containing too little important information results in too little for it to learn. For instance, if one uses an epoch of 100 ms, there might not be enough information for the algorithm to make a useful, informed decision.

In fact, in the case of a CAE with fewer data in a window, one could expect more overfitting as a result, as the features the network is trained on become less complex. Imagine a detailed landscape painting. When you zoom out to view the entire piece, there is a rich feature landscape; when you zoom in too far, all that is visible are strokes with no discernible relation to the overall picture. In this analogy, if the network learns to reconstruct simple lines and angles with no complex global features, it can easily reconstruct most normal or abnormal inputs. This is not the desired ability here. With the application of a CAE in this work, the network should be able to reconstruct only the normal data well while being unable to reconstruct any anomalous data.

Once the desired input window length was determined to be one second in real-time, it was essential to ascertain a decimation rate for the data. One second of the raw data contains 48,828 samples, which is too long an input window (in samples) for many algorithms. Windows of this size would prove to be far too memory-intensive during data loading while training, especially regarding the amount of memory required per batch on the GPU. As such, a new sampling rate, much lower than the original, was chosen as the decimation target. The new sampling rate of 2048 sps was chosen. This new sampling rate is relatively compatible with readily available and inexpensive off-the-shelf hardware on the market today, meaning one would not require an expensive and large power analyser to recreate a framework similar to this study.

The sampling rate of 2048 sps was chosen as it jointly minimises the window size in samples while maximising the resolution of the signature. Sampling rates below this, for example, 1024 sps or 512 sps, have a visible reduction in quality. Vital information could be lost and, thus, the network might not be trained sufficiently if these lower sampling rates were used. Studying Figure 5, it is difficult to visually discern any difference between the raw data (A) at 48,828 sps and the decimated data (B) at 2048 sps.

After decimating the data, some noise remained. This noise is visible in Figure 5A,B as spikes above and below the baseline data. In our previous work [4], the fundamental aspects of the signatures were studied. It is safe to assume these spikes are noise, possibly from the power analyser, which can be removed. As such, a moving average filter (MAF) was used to remove this noise in the background. A window length of 15 samples was used for the MAF as this was the maximum window size, which reduced the background noise without negatively affecting the signature of the data. The effect of the MAF can be

seen in Figure 5, where the MAF not only reduces the background noise but also effectively cleans the signature of the normal data.

Finally, the data were segmented into windows of equal length for input into the network. One second was already determined as the correct window size, so the data were segmented into windows of 2048 samples. As there was a limited amount of training and testing data, a relatively long sliding window overlap per segment of 75% was chosen for the data. This gives the network more opportunity to view specific features from slightly different perspectives, with new information being introduced with each overlapping window.

Notably, the success observed in this work using this lower sampling rate proves that one can obtain excellent results with relatively inexpensive equipment. This fact should enable the scalability of such a framework, where expensive equipment is not imperative.

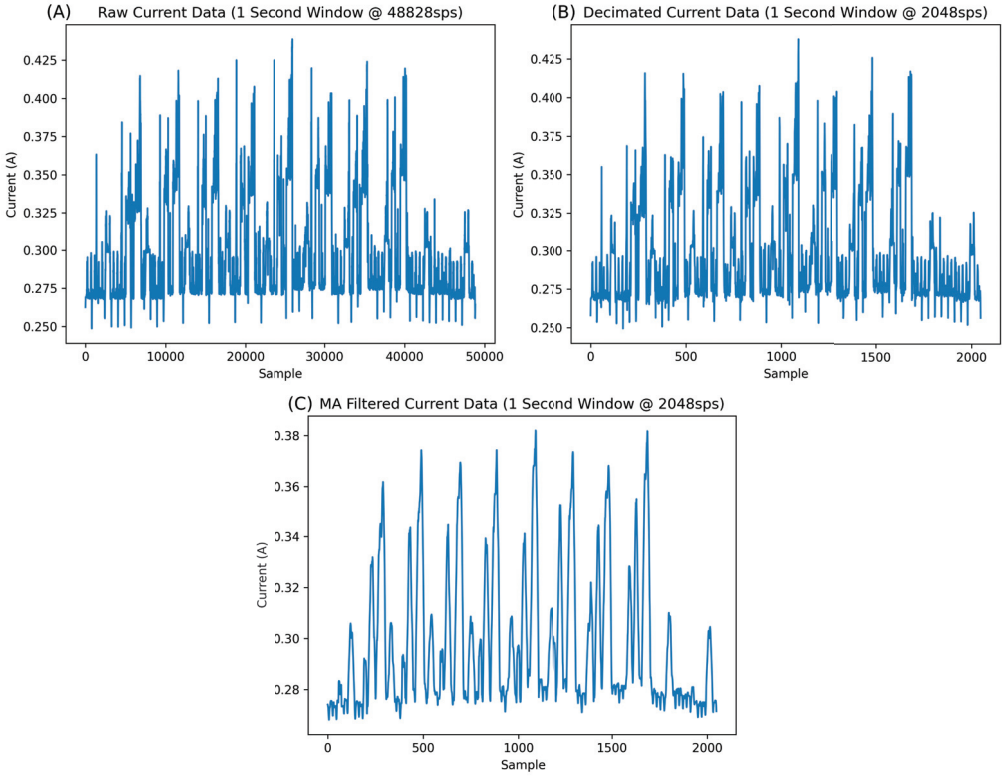


Figure 5. Preprocessing pipeline: (**A**) raw data at 48,828 sps, (**B**) decimated data at 2048 sps, and (**C**) MAF filtered data, 2048 sps, MAF size 15.

4.2. Metric Definition

The receiver operating characteristic (ROC) curve is a graphical representation depicting a classification model's performance across various classification thresholds. The following is imperative for this metric: the calculation of true positive (TP), i.e., correct positive case classifications, e.g., a sample correctly identified as an anomaly; false positive (FP), i.e., incorrect positive case classifications, e.g., a sample incorrectly identified as an anomaly; true negative (TN), i.e., correct negative case classifications, e.g., a sample correctly identified as normal; false negative (FN), i.e., incorrect negative case classifications, e.g., a sample incorrectly identified as normal.

The ROC curve illustrates the relationship between the following two key parameters:

1. True positive rate (TPR), also known as recall, is defined as follows:

$$TPR = \frac{TP}{TP + FN}$$

2. False positive rate (FPR) is defined as follows:

$$FPR = \frac{FP}{FP + TN}$$

The ROC curve showcases the TPR vs. FPR at different classification thresholds. Lowering the threshold increases the number of positive classification actions, thereby increasing false positives and true positives; raising it results in more negative classifications, increasing both false negatives and true negatives.

The area under the ROC curve (AUC) quantifies the entire two-dimensional area beneath the complete ROC curve, ranging from (0,0) to (1,1). It offers a measure of performance across all potential classification thresholds. A way to interpret AUC is the probability that the model assigns a higher rank to a randomly chosen positive example than a randomly chosen negative one. A value closer to 1.0 on the AUC is optimal for a system.

4.3. Architecture and Training

4.3.1. Architecture

In our [4], we highlighted the dynamic nature of attack signatures, which vary from one attack to another and can be changed significantly, simply by adjusting the attack tool settings. Consequently, we discouraged the reliance on supervised or signature-based detection methods for anomaly detection in IoT power traces.

Supervised learning (signature-based) involves training an algorithm on a fully labelled dataset, often with multiple classes. This would mean training the algorithm on normal and anomaly data from the dataset. On the other hand, unsupervised learning operates without requiring two or more training classes. In this study, an unsupervised approach was chosen, training an algorithm using only the held-out normal data. This trained algorithm is tested on an unseen dataset comprising contaminated normal and attack data, essentially a completely separate dataset. This approach helps mitigate complications arising from the variability in anomaly signatures while effectively utilising easily labelled and obtainable normal data.

Convolutional autoencoder (CAE) architecture was chosen to fulfil the need for an unsupervised algorithm. CAEs comprise encoder and decoder sides, employing convolutional layers for feature extraction. The embedding layer between the encoder and decoder compresses input data, and the decoding process seeks to reconstruct input from this compressed representation. The network's performance is evaluated by comparing the mean squared error (MSE) between the reconstructed and original inputs. The MSE was chosen as larger errors are penalised more than smaller ones, reducing the number of misclassifications of normal samples as anomalous. For simplicity, MSE was chosen as the only metric. During training, the objective is to minimise this error, ensuring optimal reconstruction quality, particularly for normal data (the training target). In the testing phase, the network's architecture allows for the adequate reconstruction of normal-unseen data in the test set but should encounter challenges when reconstructing anomalous-unseen data.

Convolutional layers with strides of 2 in the encoding phase were used for downsampling on the encoder side instead of convolutional layers, followed by max-pooling. This allowed for a more straightforward approach to designing the decoder side. Using transposed convolutional layers with a stride of 2, the compressed input can be upsampled to its original dimensions (reversing the encoder process). The use of convolutional layers with strides and transposed convolutional layers with strides simplifies the overall architecture, allowing for max-pooling to be omitted from the design. The omission of max-pooling also avoids any complications arising from using the dropout along with convolutional layers and pooling [40].

In the context of the particular architecture highlighted in this study, the network takes as input a segment of a 1D current trace from the testbed, specifically a length of 2048 samples. Consequently, the network output is an attempted reconstruction of this input. The desired functionality of the network involves achieving high performance in

reconstructing previously unseen normal data while encountering challenges in reconstructing unseen anomalous data.

As mentioned, the architecture featured in this work consists of encoder and decoder blocks. Focusing initially on the encoder, it can be seen that it consists of three convolution banks, consisting of 3, 2, and 2 separate convolution layers each. Instead of using max pooling, the spatial dimensions of the input are reduced using strides. As such, the last convolutional layer of each bank, with a stride of 2, halves the spatial dimension of the input. At the end of the encoder block, the network is flattened to feed into the dense compression region, where the main compression of this autoencoder architecture occurs. The dimension of 8192 for the flat layer can be obtained by multiplying the dimensions of the output of the last convolutional layer (256×1, due to the stride of 2) by the filter depth.

In the dense compression region, the dimensions of the network are further compressed from 8192 to three levels of 1024, 512, and 256 (or $8\times$, $16\times$, and $32\times$ compression) to reduce the input further. The more compression, the harder it should be for the network to reconstruct the inputs.

The decoder process is simply an inverse of the encoder, using transposed convolutions with a stride of 2 to increase the dimensions of the input (i.e., upsampling). Each convolutional layer features a filter depth of 32 and a kernel size of 3.

Figure 6 shows the Dragon_Slice CAE architecture designed in this work.

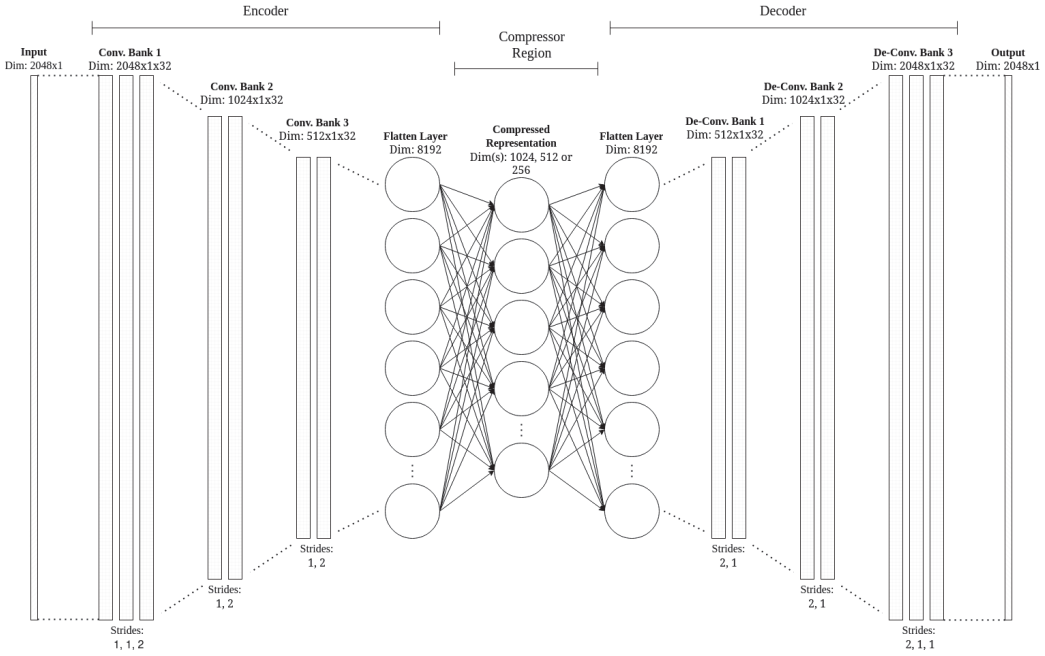


Figure 6. Dragon_Slice CAE architecture.

4.3.2. Training

Like any problem in machine learning or deep learning, the approach to the train–test split is a critical detail. In this study, the training is exclusively conducted on one class, the normal class. It is crucial to understand that the network in this research is trained solely on "clean room normal" data.

In this context, "clean room normal" implies that the training set has no overlap with the test set, ensuring the complete novelty of the test set. Consider a room with no external influence; the normal training set is sampled within that room, while the testbed operates as normal, free from external influences on the sampled data. The test data are hypothetically sampled outside this room. Although approximately 50% of the test set also consists of normal data, this normal information is interspersed between attacks, just before or after attacks, etc. Therefore, the normal training data must be sampled and kept separate without mixing with the test set to guarantee the unseen nature of the test set's

normal data. The normal data of the "clean room" will be referred to as "held-out" normal data for the remainder of this work.

The theory behind this work relies on the network's ability to learn key features comprising normal behaviour and, thus, enabling it to reconstruct unseen normal information encountered in real-world scenarios, while facing challenges in reconstructing unseen abnormal data, leading to the detection of this anomaly.

It is essential to note that this network was trained on the held-out normal data from both devices within a single dataset, allowing one model to work across multiple device types. Our previous work [4] demonstrated the similarity in signatures of attacks and normal behaviour across these devices, albeit with different current amplitudes.

Like most deep learning methodologies, autoencoders tend to overfit as they train over many epochs. Considering the limited data available for training and testing, validation during training was omitted. Instead, multiple models were saved during training, and testing was conducted over these epochs. While this could be considered fitting to the test set without a validation set as a bias buffer, the abundance of test data (compared to train) and unseen nature justify this approach. The network was exposed to various unseen anomalies and normal behaviours during testing, providing a valid representation of real-life circumstances.

In addition to obtaining multiple models over numerous epochs, the architecture was trained with multiple compression ratios in the embedding layer to explore the impact of compression on anomalous reconstruction. The objective was to balance poor reconstructive abilities of high compression (ineffective on normal but even more so on anomalous) and the good reconstructive abilities of low compression (effective on both normal and anomalous). Too many reconstructive errors in the normal class could lead to excessive false alarms. In contrast, overly good reconstructive abilities might classify abnormal, anomalous behaviour as normal; both situations need minimisation for real-world applications.

The specific target during training was to minimise the mean squared error (MSE) of the reconstruction compared to the input. Reducing this error enhances the network's ability to reconstruct seen and unseen information.

The construction and training of each model were executed utilising the Python PyTorch package. A consistent learning rate of 5e-5, a batch size of 64, and the Adam optimiser were used during the training process. No learning rate scheduling was implemented throughout training.

The network's performance in training and testing relies on the MSE of the reconstruction versus the input. During training, the MSE serves as the loss function for the network to reduce and better fit the training set. In testing, the MSE indicates how poorly the network has reconstructed an input test sample. Ideally, the MSE for a normal unseen data segment should be minimal, while abnormal unseen data should result in a large error.

It is crucial to note that the MSE is a value obtained for an input–output pair, meaning that there is only one MSE value for every 2048 samples in the input segment. This aspect will be explored in greater detail when discussing the results, considering metric performances and reconstructions, and visually inspecting the MSE in later sections.

The latent compression in the central squeeze of the network increases incrementally, with the chosen compression ratios in this work being 8× (1024), 16× (512), and 32× (256).

4.3.3. Results

As outlined in Section 4.3.2, the CAE network detailed in this study underwent training on held-out normal data, progressively increasing the compression in the central squeeze of the network (i.e., a smaller latent space). The chosen compression ratios were 8× (1024), 16× (512), and 32× (256).

The best-performing epochs for each compression ratio are compared below. The use of validation or cross-fold validation was omitted from this work. Instead, these models were tested on the entire unseen test set simultaneously, resulting in performance metrics for the entire set of 66,306 unseen segments. It is essential to emphasise that the test set is

entirely unseen, treated separately from the held-out training set. The training and test sets are distinct entities, resembling a clean room analogy where normal training data have no contamination, and the test data are obtained later after acquiring the normal training data. The unseen test set should not influence the design or training of the model. As such, it was deemed inappropriate to use data from the unseen test set for validation during the training process in the case of this specific work.

Crucially, the mean square error (MSE) of the test data underwent processing through a moving average filter (MAF) to broaden the inference response and reduce noisy spikes during normal inferences. The impact of this filtering is evident in Table 4. This study considers the moving average-filtered MSE when determining the best epoch for each model.

Various MAF window lengths were experimented with to find the most optimal over the unseen test set, as shown in Figure 7. While there is debate about fitting a filter or a threshold to the test set directly, the study presents a range of area under the curve (AUC) values per MAF window length in Figure 7. This provides a suitable MAF range as a suggested approach. The remainder of the study operates based on the best-performing MAF for this network. Given that the MAF is fit over the entire unseen test set, much larger than the training set, this should be closer to an appropriate value as in a real-world scenario. It is also crucial to remember that the unseen data in this study represent real-world data, as the normal test set was held out before the test set was even sampled.

An MAF with a window length of 16 (approximately 4 s of data) was found to be the most effective over the entire dataset. It is worth exploring whether employing specific MAF window lengths for each anomaly type, varying between larger and smaller lengths, could enhance network performance, but this is out of the scope here.

The necessity of the MAF on the MSE stems from the labelling of the dataset. Since the data were hand-labelled, care was taken to label each anomaly correctly. Instances were labelled as anomalous from the start of a known anomaly (easily identifiable) to when the data returned to normal (somewhat ambiguous). However, the CAE may not capture the entire window, leading to false negative and positive predictions of anomalies, especially in cases like brute force segments. In such cases, a longer label was applied, and the MAF widened the MSE response (over more input windows), aiding the network in aligning with longer labels.

The network's best performance, albeit marginally, occurred at a compression level of 1024 in the central squeeze of the network (8× compression), which is a reasonable compression amount for a network.

There was a slight performance drop at higher compression ratios, but the training time to reach the best-performing epoch increased considerably. This was expected, as more compression makes it more challenging for the network to reconstruct the input, necessitating a longer training time to learn the same data. The most extreme case was the best performance for the highest compression ratio (32× at a latent size of 256), requiring 4750 epochs, as shown in Table 4.

Figure 7 further explores the performance of the best-performing network (1024 in the latent space compression). ROCs for the raw MSE (without MAF) are depicted, revealing a result of approximately 0.778 AUC. This indicates a skilled classifier, as an AUC of 0.50 represents random choice. Figure 7B illustrates the AUC of the network over the entire unseen test set while increasing the MAF window length for the MSE output. This figure demonstrates improved performance with MAF values, likely influenced by how the dataset is labelled. An MAF window length of 16 (approximately 4 s) performed the best, exhibiting a substantial increase in AUC from 0.778 raw to 0.876 with an MAF of 16. This ROC is visible in Figure 7C.

The good detection versus false alarm rate is demonstrated in Figure 7D. This figure illustrates the event-based metric of the number of anomalies correctly identified compared to the total number of anomalies (as opposed to each window, it looks at each instance of an anomaly). A false alarm is calculated as a rate measured over one hour. As visible in Figure 7, this network can detect up to 50% of anomalies with no false detections, 70%

of anomalies with fewer than ten false detections per hour, and 80% with approximately 50 false detections per hour. False detections per hour are crucial, as more false detections can disrupt the system's normal operation.

The effect of such an MAF on the MSE can be seen in Figure 8. It is essential to note that each data point in this figure represents the MSE value for an entire window input to the network, to be explored further in this work. As seen in this figure, the MAF not only broadens the network response to anomalies—aiding in detecting more of the overall anomaly—but also helps in smoothing out and reducing the Bluetooth segment error, reducing the number of false positives where normal data have momentarily higher errors than the threshold (anomalous regions and normal Bluetooth regions are labelled in Figure 8).

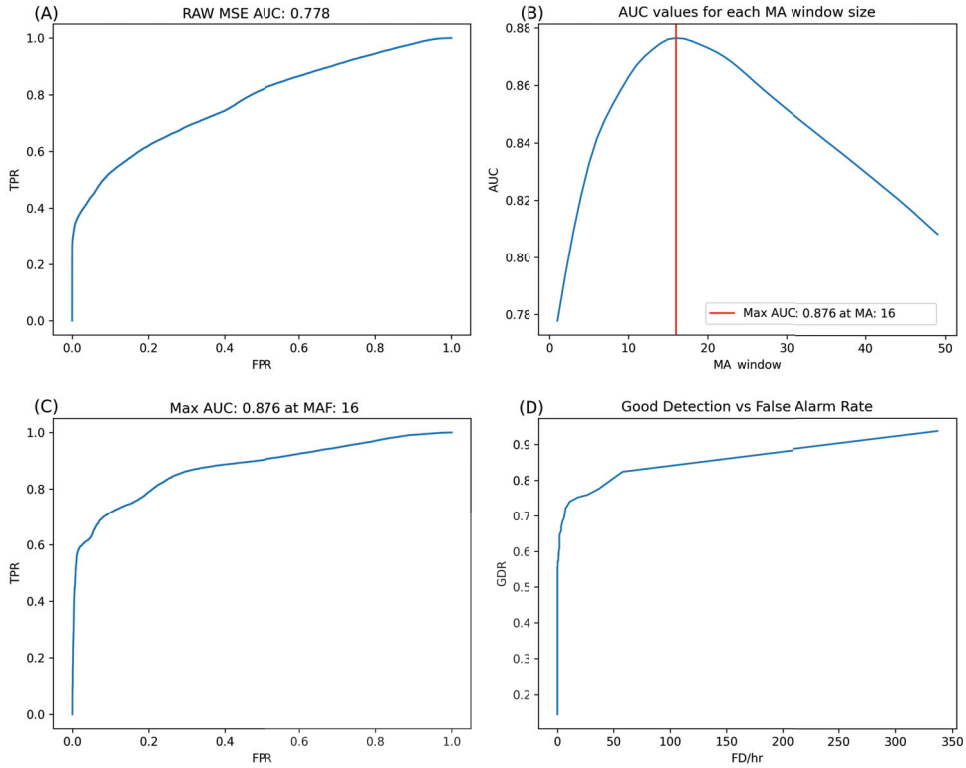


Figure 7. Performance of the Dragon_Slice_1024 CAE architecture in terms of raw AUC (**A**), best MAF size for MSE (**B**), AUC at best MAF size, (**C**) and GDR (**D**).

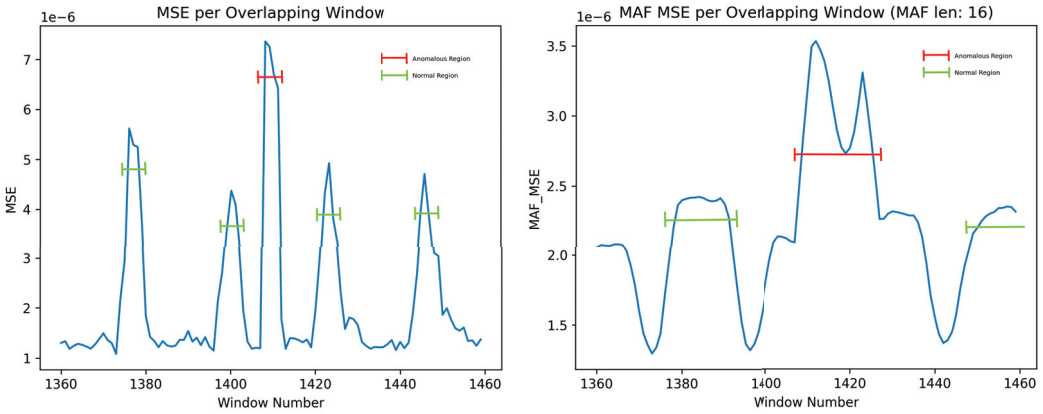


Figure 8. Effect of the MAF on the MSE.

Table 4. Best performing epoch for each compression amount.

Compression	Best Epoch	RAW MSE AUC	MAF MSE AUC	MAF Window Length
1024	250	0.778	0.876	16
512	500	0.768	0.874	16
256	4750	0.772	0.868	16

4.3.4. Reconstruction Examples

In this age of AI, it is crucial to prioritise explainability in conjunction with the chosen methodologies. This entails justifying the steps taken and delving into a network's functionality from a comprehension perspective.

As previously mentioned, the input to the CAE network is a one-second segment of the testbed's current data. With preprocessing, this segment consists of 2048 samples in length. The output, being a reconstruction-based approach, represents the network's attempted reconstruction of this input. An input sample is classified as normal or anomalous based on how well, or poorly, the network reconstructs the unseen input data. The metric for reconstruction quality is the mean squared error (MSE). This also serves as the loss function during training, and while other loss/error options exist, MSE is the industry standard for this problem case. Examples of the actual MSE output of the network will be detailed in the next section, but firstly, the network's reconstructive ability will be explored.

The fact that the network produces a tangible output is of utmost importance in explaining this AI, understanding how it operates, and making informed decisions on the presented unseen data.. Crucially, the architecture in this study outputs no probabilistic output but, rather, a reconstruction and the associated MSE for the entire input window. Inspection of this can be far more explainable than probability-based outputs.

Using the reconstructed output, it is possible to identify areas where the network had difficulty reconstructing. This allows for the referral of that specific anomalous segment to a higher level, such as a human expert, for inspection. This utility is somewhat similar to a fully convolutional neural network (FCNN) operation, whose unique design enables tracing back from the probabilistic output to the exact part in the input signal that contributed to that decision.

Figure 9 displays the reconstruction of various input types, comparing the reconstruction for DragonBoard versus Raspberry Pi. It is important to remember that this network was trained on both devices' held-out normal data simultaneously. Thus, this model works across two devices. Our previous work showed the similarity in signature, albeit at vastly different amplitudes, attacks, and normal behaviours across these devices. The ability to use one model across two devices suggests great scalability for such architecture.

Figure 9A,B illustrate the network's reconstructive ability on unseen normal data, demonstrating impressive results. This proves that the network successfully learned to reconstruct the normal information. The lower the associated error compared to anomalies, the better the network's performance.

The reconstructions in Figure 9A,B support the hypothesis from [4], i.e., the shape of the signatures of the normal data were close enough, regardless of the significant difference in current amplitude between the two boards, allowing one network to be trained on both boards' normal data and detect anomalies in both. This similarity in signature, despite the difference in maximum amplitude, can be seen in Figure 9A,B, ranging from approximately 0.16 Amps for the DragonBoard to approximately 0.38 Amps for Raspberry Pi. However, regardless of this amplitude difference, the network performed remarkably at reconstructing the normal information for each board.

More importantly, the network's reconstruction of anomalies is investigated throughout the rest of Figure 9. In Figure 9C,D, the reconstruction of a port scan attack, namely Nmap_T3, is shown for both DragonBoard and Raspberry Pi. Notably, the signature for this specific attack setting for DragonBoard, Figure 9C, has a remarkably low amplitude, almost equal to Bluetooth for DragonBoard and far lower than that of Raspberry Pi. Anomalies

like this are instrumental in demonstrating that the network is reconstructing based on a learned shape rather than simply struggling with high amplitudes. The only way for the network to struggle to reconstruct this specific anomaly signature is based on its shape, which is the main contributor to the anomalous classification of this segment.

In the case of Raspberry Pi in Figure 9D, while it is observable that the network struggles with the shape, most of the error contributing to the anomalous classification comes from the much higher amplitude of this anomaly over the normal expected amplitude. In fact, throughout the entire training set, the normal data peaks for Raspberry Pi reach an amplitude of approximately 0.38A. It follows that the network cannot reconstruct inputs far greater than those to which it was fit during training, which is most certainly the case in Figure 9D.

Figure 9E shows the reconstruction of a brief and low amplitude signature SSH brute force attempt, technically, a single password attempt using Hydra_T1. As visible in Figure 9D, the amplitude of the attack is not that different from that of the Dragon_Pi's normal Bluetooth and is much lower than that of Raspberry Pi. As such, it can be assumed that the error contributing to the anomalous classification must be from the abnormal shape of the signature rather than the network struggling with a large amplitude signature.

Figure 9F shows the same Hydra_T1 SSH brute force attack on Raspberry Pi. In this case, the amplitude is much lower than that of the port scan attack at Nmap Setting T3. While the amplitude is slightly higher than the expected normal at approximately 0.45A, it would be more appropriate to say that the contributing factors are almost equally shaped parts of the signature and its amplitude, as the network struggles to reconstruct both.

Figure 9G shows the reconstruction attempt for Hydra_T16 on the DragonBoard. While this amplitude is double what is usually expected from the DragonBoard, it is still lower overall than the maximum expected for the normal data overall. As such, it is possible to conclude that the contributing factor to the error is, in fact, the shape of this anomaly rather than amplitude. It is also clear that with such a divergent shape from normality, the network greatly struggles with reconstruction.

Figure 9H shows the reconstruction of Hydra_T16 on Raspberry Pi, a much higher amplitude signature than that of DragonBoard and, indeed, that of any normal information. As such, it is clear that the main contributing factor for the error in this case and, thus, the anomalous classification, is this large amplitude. It is safe to assume that anything largely above normal behaviour will be completely irreconstructible, always resulting in an anomaly classification.

Figure 9I shows signatures taken when the devices were infiltrated by attackers rather than just a network response; specifically, Figure 9I shows a command run on the device. This signature is unique and much lower than the expected normal behaviour. As such, studying the input in Figure 9I, it is simple to conclude that the contributing factor to the error is a mixture of both the inability of the network to reconstruct the unseen pattern and the inability to reconstruct the high-frequency component that can be seen in parts of Figure 9I.

Figure 9J shows signatures obtained while the attacker infiltrated Raspberry Pi; specifically, the attacker viewed some files on the target over SSH using the CAT command. In this case, as the signature is somewhat higher in amplitude than expected in the normal behaviour, the main contributing factor to the anomalous classification is the amplitude, with the shape being the secondary contributing factor.

By studying the reconstruction of the network over both normal expected behaviours and anomalies, it is straightforward to draw some conclusions about this specific network's operation. Firstly, the network fits well with the normal data exhibited in Figure 9A,B. This positively demonstrates half of the desired operation of the network, which must (a) well-reconstruct unseen data close to their training class and (b) struggle to reconstruct anomalous patterns entirely dissimilar to the normal training class.

Regarding Figure 9C, which is a low amplitude anomaly, while there is clearly an increase in error, the network does a surprisingly good job of reconstructing the anomaly.

While there is some struggle with the shape of the anomaly during reconstruction, it seems that the network can fit unseen data to a degree. This ability is undesired, with the desired behaviour being a significant struggle with anomalous samples. This specific case serves as a good benchmark for reconstructive ability; as such, it will be further explored in Section 4.3.5.

In cases such as Figure 9D,F,H, it is clear there is a maximum learned amplitude that the network can reconstruct, with a complete inability to reconstruct past this point. This was learned passively during training, as the training set peaked at approximately 0.38 A. In cases such as Figure 9C,E,G, the amplitude of the anomaly is significantly below that of the maximum expected normal behaviour, so it can be inferred that the main contributing factor to the error, in these cases, would be the unexpected shape. This is the most desirable behaviour of the network—the purpose of using a CAE is the ability to learn and reconstruct complex expected normal patterns while struggling, as much as possible, with patterns that would be entirely unexpected for the network.

Overall, the reconstructions are relatively close to the input. This is not desirable. The best-case scenario is to minimise the normal class error while maximising the abnormal class error. Figure 9 shows a network that fits very well with normal examples while only minimally struggling with lower amplitude anomalies. Of course, anomalies vastly outside normal operating amplitudes are completely irreconstructible. This suggests overfitting in the network or at least some similar phenomenon. This can occur when a network is too complex for the task, or the test set is far smaller than the training set. In the case of this network, the former is more likely the issue. Before any changes are made to the network, a specific MSE case is explored in Section 4.3.5.

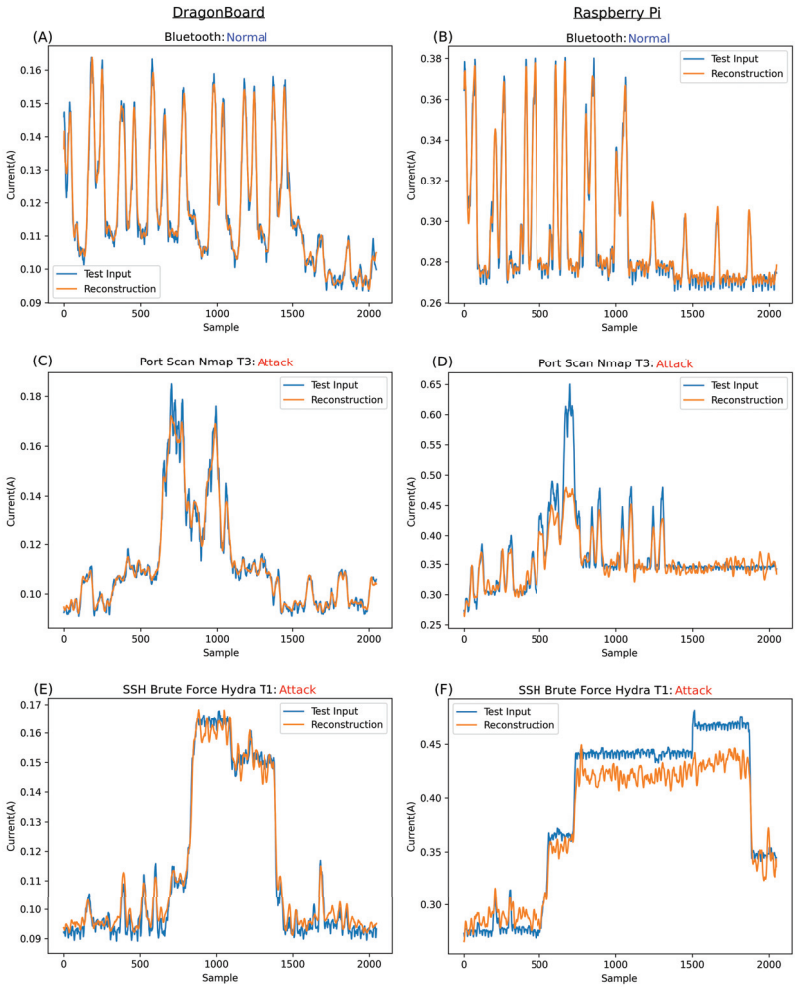


Figure 9. *Cont.*

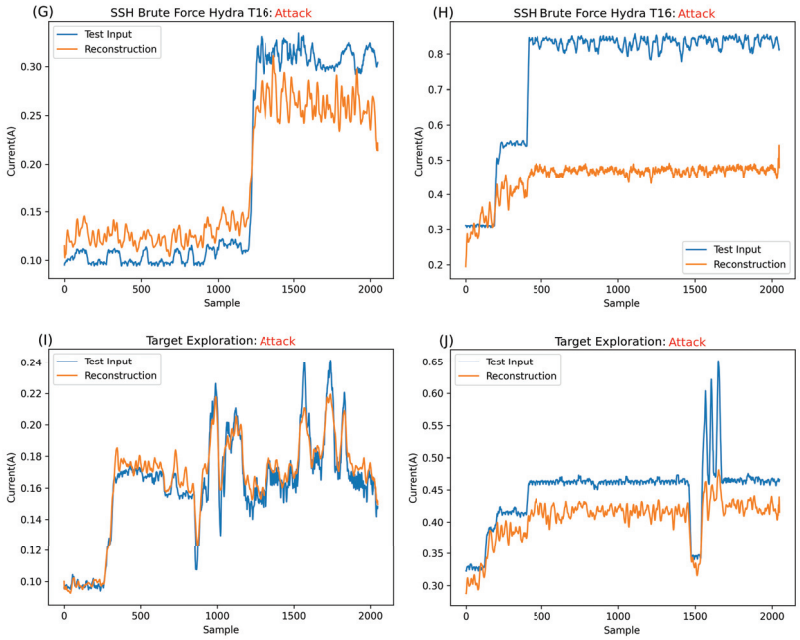


Figure 9. Dragon_Slice_1024 DragonBoard and Pi reconstructions.

4.3.5. MSE Exploration

Given its significant role in the network's operation, it is imperative to explore the mean square error (MSE) of the network's reconstruction output. In the era of AI, clarifying how decisions are made in the network is crucial. This involves examining both the reconstructions (the actual outputs of the network) and the errors associated with these reconstructions.

Figure 10A illustrates the MSE of 101 segments (specifically segments 1360 to 1460). It is essential to note that each point in this graph represents the MSE of an entire output segment (a reconstruction of 2048 samples) versus the original input (the original 2048 samples corresponding to the output).

In the normal class, a burst of Bluetooth behaviour occurs every five seconds, leading to an expected rise in the MSE. However, this error increase should not surpass the baseline significantly and should be markedly lower than any error spike caused by an anomaly.

Upon closer inspection of Figure 10, some concerns arise. Despite the network's relatively good metric-based performance, the MSE output for the anomalous window (e.g., window 1410, highlighted in red) appears too close to the normal Bluetooth segments (e.g., windows 1400 and 1423, highlighted in green). A detailed examination of these windows reveals that the network excels in reconstructing normal Bluetooth behaviour (Figure 10B,C) but, unexpectedly, reconstructs the anomaly at window 1410 quite well (Figure 10D). This can be attributed to the unique overfitting nature of autoencoders, wherein, without sufficient compression, they can reconstruct most inputs.

While the network undeniably functions, the results on anomalies, such as example 1410, which have much lower power than the brute force segments, exhibit errors too close to normal behaviour for the network to be deemed robust. For improved robustness, the network must encounter more challenges in reconstructing anomalous behaviour, thus establishing a more substantial gap between the MSE of normal behaviour and anomalies.

The anomaly featured in Figure 10 is particularly challenging to detect due to its low amplitude. Overall, it is evident that the network fits exceptionally well with this type of anomaly, as evidenced by the remarkably low MSE error (around 7e-6) and the amplitude of the anomalous error compared to the surrounding Bluetooth regular data. Applying a straightforward classification threshold to the MSE (as depicted in Figure 10A at around 6e-6) highlights the closeness of anomalous classification to normal Bluetooth information.

Therefore, enhancements to the network architecture are necessary architecture to make it more difficult to reconstruct abnormal, unseen information, ultimately creating a more resilient network. Even if the performance improvements are marginal, a network that struggles significantly with abnormal data would be far more trustworthy and robust.

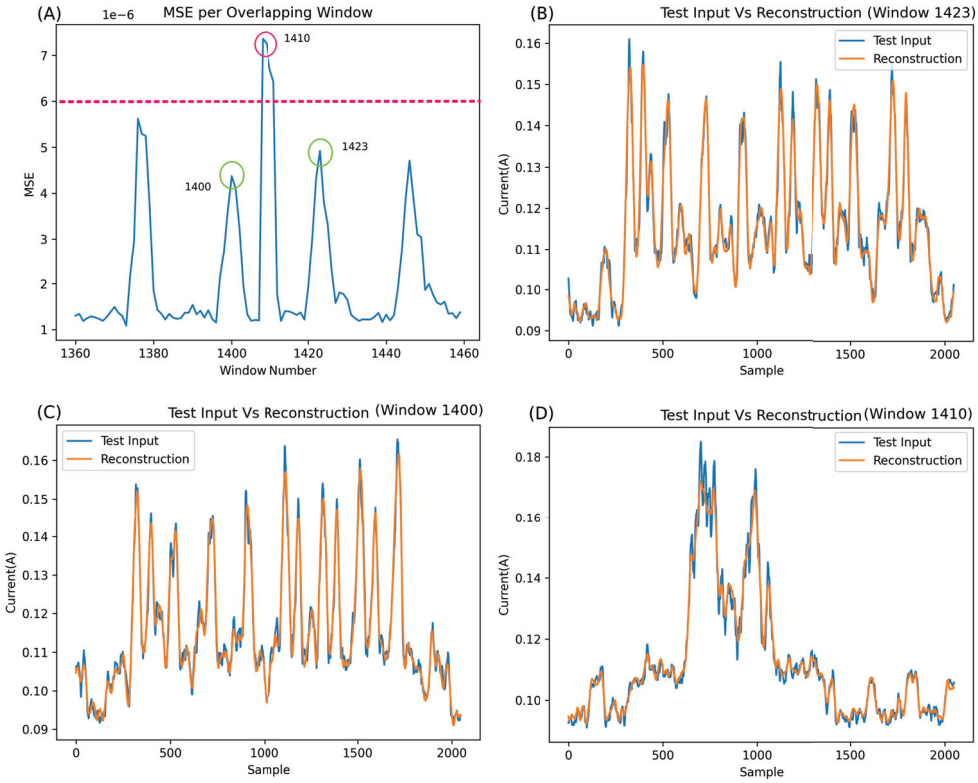


Figure 10. Dragon_Slice_1024 and the Pi reconstruction MSE explored.

4.4. Dropout Architecture and Training

4.4.1. Architecture

Networks like CAEs often tend to overfit. Various strategies exist to mitigate overfitting, such as utilising a smaller, less complex network or enhancing compression. However, previous observations from Section 4.3.2 indicated that increased compression did not yield improved performance.

The decision was made to incorporate dropout [41] in the network rather than altering the overall architecture, such as removing layers. Dropout, a regularisation technique commonly used in deep neural networks, can significantly alleviate overfitting. By randomly excluding nodes during training, based on a predetermined probability, dropout introduces noise to the training process. This noise forces other nodes within a layer to assume more or less responsibility, limiting the influence of adjacent nodes. Dropout intensifies the training difficulty, pushing each node to learn more independently rather than relying on neighbouring nodes, where a path of least resistance might occur for classification. In AEs and CAEs, this reduces overfitting and introduces complexity to the learning process. The intermittent deactivation of neurons makes the reconstructive process more challenging to learn, compelling the network to grasp intricate patterns within the training set rather than allow for nodes to rely only on their neighbours to make decisions.

The notion of introducing noise to an autoencoder-based network extends beyond the use of dropout. Denoising autoencoders intentionally corrupt the input to the network or the input in transit through the network during training [42]. The objective is for the network to learn to navigate around this corruption, encouraging it to reconstruct the original uncorrupted input. While denoising is typically employed to remove noise, this study employs a similar concept for anomaly detection. Instead of corrupting the input,

dropout is used to train the network noisily. This compels the network to learn around the noise, resulting in more robust reconstructions. In the case of anomalies, the network aims to reconstruct them as it deems 'normal', replacing the anomaly in the unseen data with its own 'normal' reconstruction.

Comparing the dropout-based reconstruction to the original input reveals a more significant error for anomalies. This approach enhances the network's robustness by preventing it from merely attempting to fit whatever input it is presented with, leading to a more reliable and trustworthy network.

While dropout was employed in this study to introduce noise, this study hypothesises that corrupting the actual input could yield similar, or even better, results. This theoretical approach could also generate additional training data by creating multiple permutations for a single training sample, which could assist in cases with sparse training sets. Although outside the scope of the current study, it will be explored in future research.

The probabilistic deactivation of nodes during training is disabled during testing. As such, the entire network is used during testing. Apart from the dropout layers, which are applied after the GELU activation functions, the architecture remains the same. The probability of dropout on convolutional layers remains very low at 0.1, as in [41]. The probability of dropout in the dense layer was 0.5. Figure 11 shows the revisions to the Dragon_Slice CAE architecture designed in this work. Note the addition of dropout.

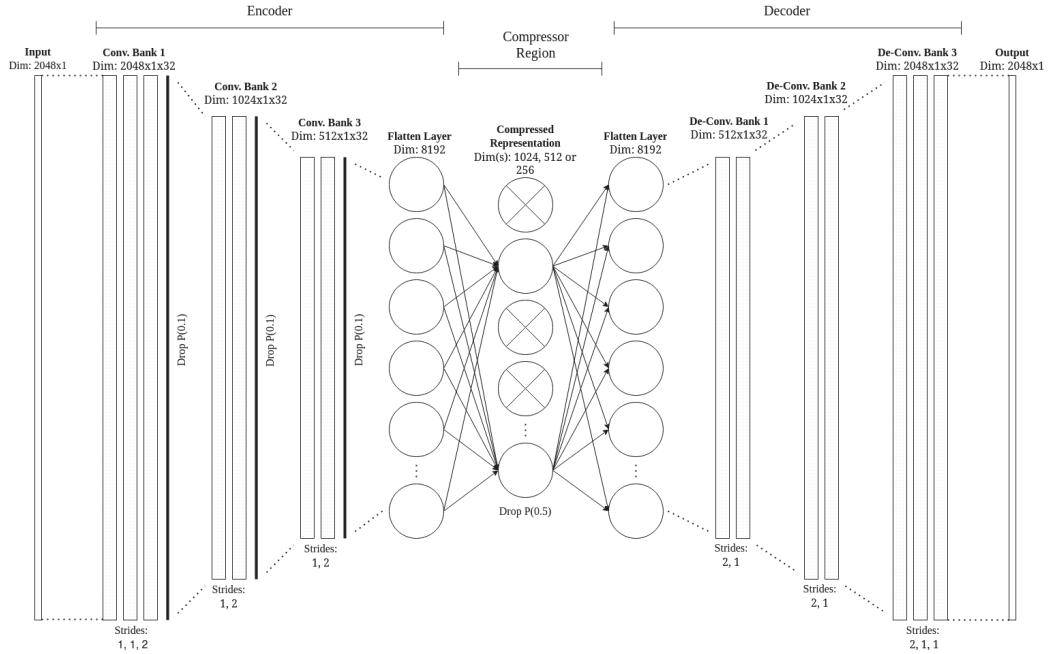


Figure 11. Dragon_Slice CAE architecture with dropout.

4.4.2. Training

Apart from the addition of dropout, the training process remains the same as in Section 4.3.2. Notably, with the addition of dropout, the number of epochs required to reach the best-performing model increases. For example, for 1024 net, the best model went from performing best at the 250th epoch to performing best at either the 1500th or 2250th epoch for the same architecture with dropout (both epochs included as they achieved almost identically the highest performance). Figures for the remainder of the study use the 2250th epoch for the 1024 dropout network.

4.4.3. Results

Incorporating dropout into the network resulted in a reasonable improvement in MAF-based performance (from 0.876 without dropout to 0.89 with dropout), accompanied by a marginal decline in raw performance, as seen in Table 5.

Once again, a slight reduction in performance is observed as latent compression increases. This reaffirms that the 1024 dropout network, with the lowest compression ratio, remains the best-performing configuration. Consequently, all figures in this section pertain to the 1024 dropout network.

Beyond the enhanced AUC performance, there is a notable improvement in the GDR, reaching approximately 65% correct detection at 0 false positives per hour, 77% correct detection with fewer than 10 false detections, and 85% correct detection with fewer than 50 false positives per hour. This is shown in Figure 12.

More crucially, the focus shifts to the reconstructions themselves and the overall resilience of the network in terms of the gap between MSE for normal Bluetooth behaviour and an anomaly. This aspect will be thoroughly discussed in the following section.

Table 5. Best performing epoch for each latent compression (with dropout).

Compression	Best Epoch	RAW MSE AUC	MAF MSE AUC	MAF Window Length
1024	2250	0.766	0.890	17
1024	1500	0.764	0.890	17
512	2000	0.759	0.886	18
256	3250	0.752	0.884	17

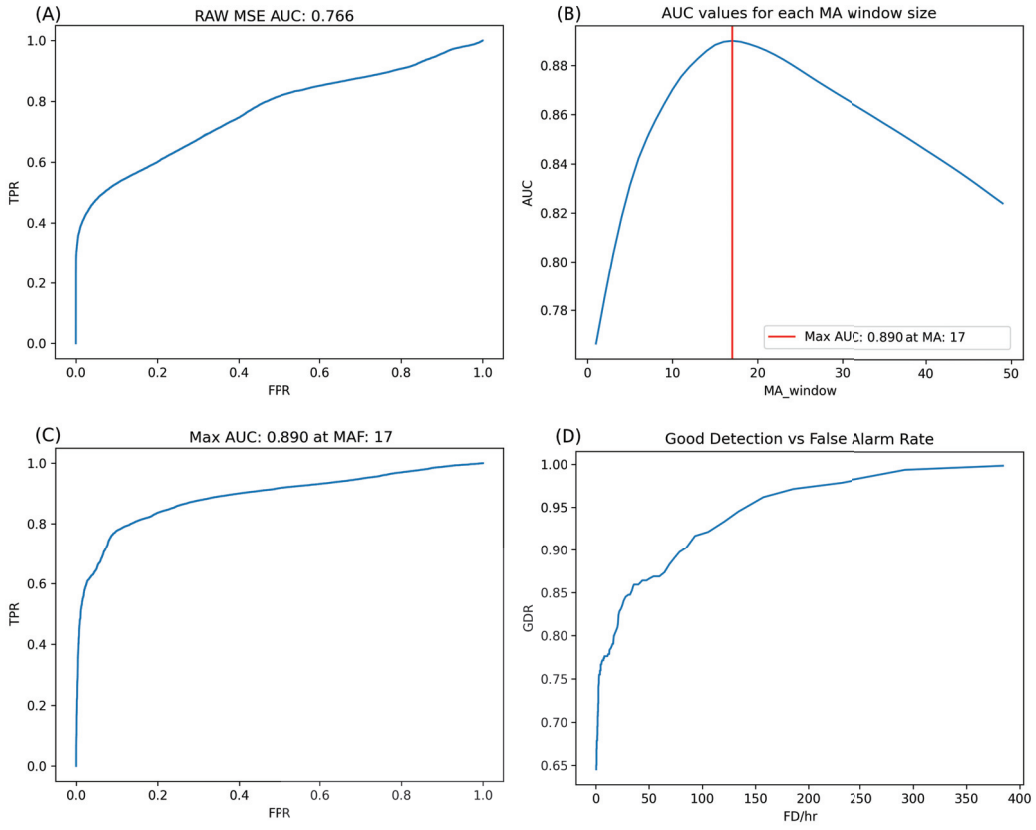


Figure 12. Performance of the dropout-based Dragon_Slice CAE in terms of raw AUC (**A**), best MAF size for MSE (**B**), AUC at best MAF size (**C**), and GDR (**D**).

4.4.4. Reconstructions with Dropout

Figure 13 depicts the reconstructions of the same normal behaviour and anomalies as Figure 9.

In Figure 13A,B, it is evident that the network performs satisfactorily in reconstructing normal behaviour, albeit with an overall increase in error compared to the non-dropout network. Despite this heightened error, the network demonstrates an evident ability to

reconstruct normal behaviour. An increase in the overall network error is acceptable, as the crucial factor lies in the gap between the normal and anomalous error, rather than the minimisation of the overall error. The anticipated reduction in reconstruction quality with dropout was also observed.

With the edition of dropout, the network is less prone to overfitting, resulting in a reconstructed output that is more reliable and robust. This means one can have greater confidence in its performance.

Figure 13C,D show the reconstruction of the Nmap_T3 anomaly. In this case, the network with dropout performs poorly in reconstruction, which is the desired outcome (the main aim is to have adequate performance for normal data and poor performance for abnormal data). In Figure 13C, the network struggles significantly more to reconstruct the anomaly than the non-dropout network, where reconstruction is more successful and seems to follow the signature more accurately. With dropout, the network fails to reconstruct the abnormal part of the signal effectively due to the unexpected shape of the signal. This behaviour is highly desirable, showcasing a more robust response to anomalies.

Figure 13D demonstrates even poorer reconstruction performance, considerably worse than without dropout. This is because a part of this signal exceeds the maximum expected normal operational amplitude, causing the network to struggle to make sense of the signal. Instead of attempting to follow the input, the network attempts to fit a shape it deems most appropriate for the entire signal, not just the anomalous part, as seen in Figure 13C, leading to a much higher overall error for this window. Again, this behaviour is desirable, indicating a more robust response to anomalies.

Figure 13E displays a notable decline in reconstructive performance for the SSH Brute Force Hydra_T1 anomaly on both the DragonBoard and Raspberry Pi. As the amplitude of the anomaly on the DragonBoard is not greater than the maximum for the normal class, the network does a reasonable job of reconstructing the normal behaviour around the anomaly (if the amplitude was more significant, the reconstruction would fail for the entire window). However, when it comes to abnormal behaviour, the network struggles significantly to reconstruct the anomaly, which is another desirable outcome. A comparison with the non-dropout network highlights how much closer the non-dropout network reconstructs the anomaly. In the dropout network, there is a complete inability to reconstruct the anomaly, aligning with the system's desired behaviour.

Figure 13F exhibits the reconstruction of an example of the SSH Brute Force Hydra_T1 for Raspberry Pi. In this case, the network cannot reconstruct the input, likely due to the information within the window spiking well beyond the maximum normal operational amplitude. The network attempts to reconstruct the input based on some pattern of normal information learned during training, thus failing to reconstruct the input at all.

Figure 13G shows the reconstruction attempt for SSH brute force Hydra_T16 for the DragonBoard, where a complete inability to reconstruct is observed. Interestingly, the network reasonably reconstructs the baseline normal behaviour before the significant amplitude spike. This reconstruction continues for the large amplitude anomaly rather than attempting to reconstruct it in this segment.

Figure 13H displays the SSH brute force Hydra_T16 anomaly on Raspberry Pi and the attempted reconstruction. Again, instead of reconstructing the anomaly, the network seems to be attempting to fit the reconstruction to some pre-learned pattern. However, it is challenging to determine what it is trying to fit the reconstruction to in this case, as it seems to be somewhere between the behaviour of the DragonBoard and Raspberry Pi. Nevertheless, the poor reconstruction is the most desirable output for this kind of network. Notably, as the anomaly far exceeds the amplitude of the normal behaviour for both boards, the network cannot reconstruct any of the signal, not even the normal information around 0.3 A at the start of the signature. This is good as it results in a higher error for the segment.

Figure 13I,J show the reconstruction attempts for current traces obtained from an attacker exploring the victim's device and investigating files. There is a complete inability to reconstruct the anomalies in both cases (highly desirable behaviour).

With the addition of dropout to the network, the system can reasonably reconstruct normal information while struggling significantly more with the reconstruction of anomalies. This aligns precisely with the intended results when incorporating dropout into the network.

Upon studying the reconstruction attempts of anomalies with dropout, it is evident that shape plays a much more substantial role in confusing the network. While shape plays a supporting role in contributing to the error in the network without dropout, it now plays a key role, equal to amplitude, in the dropout-based network. This behaviour can be quite interesting, especially for anomalies below the normal maximum operational amplitude, such as Figure 13C,E,F, all of which exhibit a complete inability to reconstruct the anomaly while remaining well below the maximum amplitude values observed in the normal data.

In cases where the anomaly spikes above the maximum amplitude of normal data, as seen in the dataset, unlike the non-dropout network where large spikes seem to be reconstructed to a point with a plateau at approximately 0.45 A, the dropout-based network becomes highly confused. This results in the network attempting to fit some unusual shapes and signatures based on its learned memory to fill the gap in its knowledge, resulting in a much higher overall error.

Overall, adding dropout to the network accomplished precisely what was intended, with anomalies being nearly impossible for the network to reconstruct, while normal data are still reasonably well reconstructed. This creates a much more robust network overall, with a much more pronounced distinction between normal and anomalous behaviour.

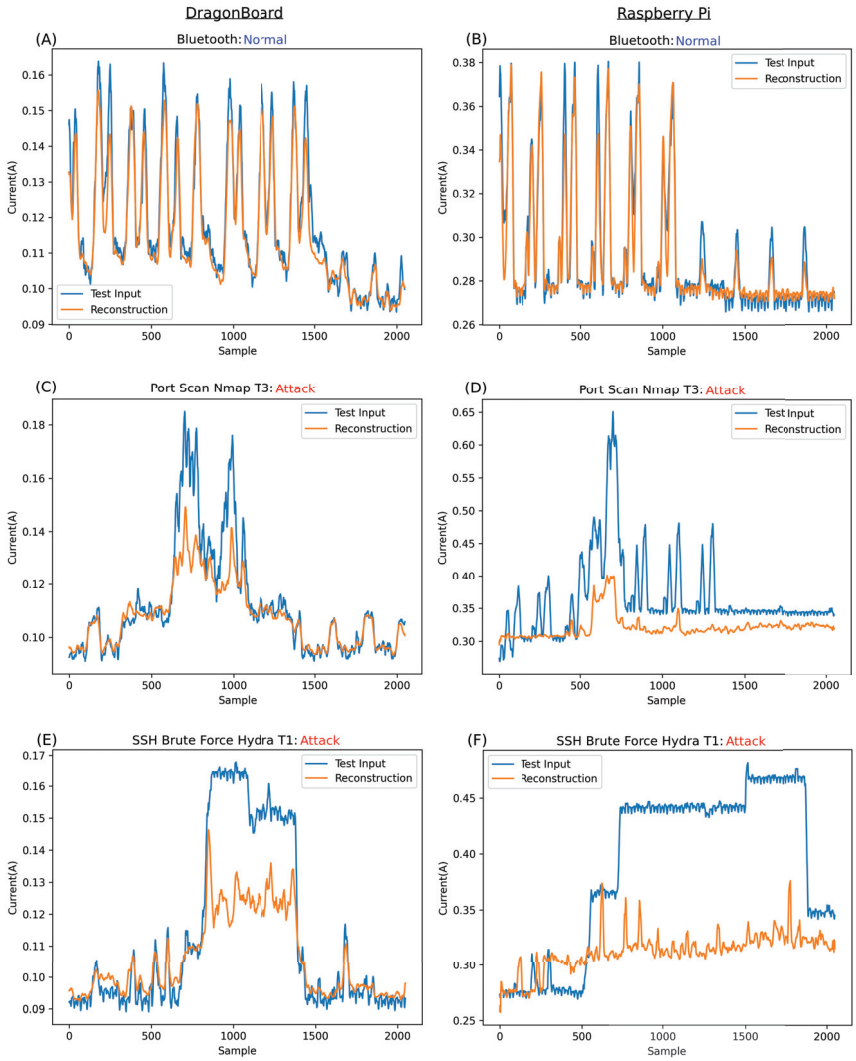


Figure 13. *Cont.*

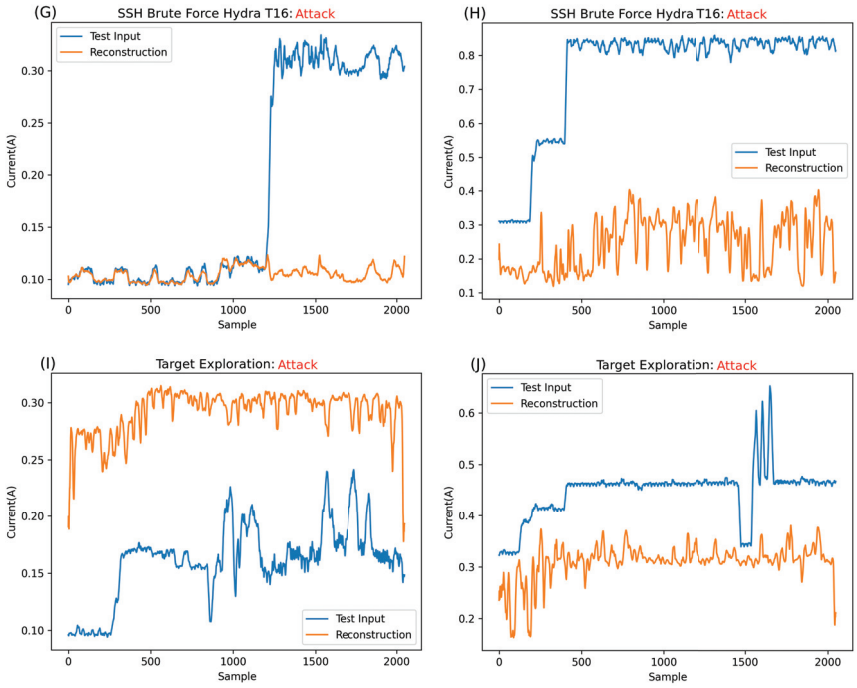


Figure 13. Dragon_Slice_1024_Dropout DragonBoard and Pi reconstructions.

4.5. MSE Exploration With Dropout

Examining the MSE for the network's reconstructions highlights the success of dropout. When observing the same segments as in Figure 10, a distinct separation in the error of the Bluetooth and the anomaly becomes apparent.

In the case of the two Bluetooth reconstructions shown in Figure 14B,C for segments 1400 and 1423, reasonably good construction is evident on this normal data, albeit at a higher error compared to the non-dropout network. This additional error for the normal data is acceptable and expected. After all, this model's primary objective is to reconstruct normal information reasonably well while struggling significantly with anomalies, as demonstrated in Figure 14.

The error caused by the port scan attack at Nmap setting T3 in Figure 14D is now distinctly separated from the normal Bluetooth spikes around it, almost six times higher. Applying an arbitrary threshold, for example, around midway between normal and abnormal, illustrates substantially greater room to minimise false positives with the dropout-based network.

The clear distinction in error between normal and anomalous behaviour proves that the edition of dropout results in a much more trustworthy and robust network. A more detailed exploration of the improvements will be undertaken in Section 5, where the non-dropout and dropout networks are more specifically compared.

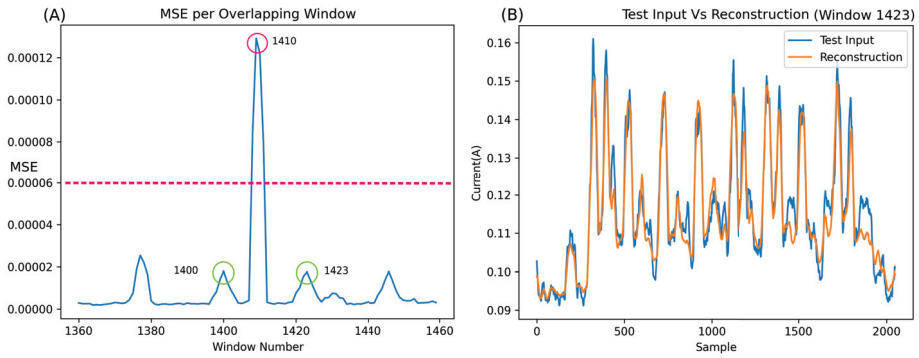


Figure 14. *Cont.*

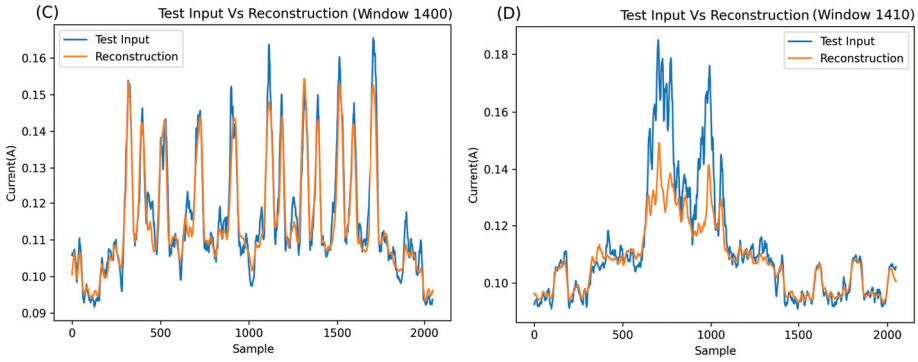


Figure 14. Dragon_Slice_1024_Dropout DragonBoard and Pi reconstruction MSE explored.

5. Discussions

5.1. Comparison between the Non-Dropout and Dropout Networks

This section explores distinctions between the best-performing dropout and non-dropout models. Figure 15 provides a more accessible comparison between the non-dropout and dropout networks, presenting reconstructions of various anomalies selected from the two boards.

In Figure 15A,B, the focus is on the specific reduction in the quality of the reconstruction of normal Bluetooth behaviour between the two networks, specifically for the DragonBoard. With its adept ability to fit the input data, the non-dropout network excels at reconstructing the normal signal. On the other hand, the slightly constrained reconstructive ability of the dropout-based network is evident in (B). Despite a reconstruction error increase, the dropout network still performs well in reconstructing normal Bluetooth data. The ideal dropout value should maximise normal reconstruction quality while minimising anomalous data reconstruction.

Comparisons in Figure 15C,D evaluate the reconstruction quality of the non-dropout network against the dropout-based network for a port scan attack with an Nmap setting T3 signature on the DragonBoard. Both networks struggle with the high-frequency component of the spike. However, the dropout-based network distinctly refuses to reconstruct the entire anomalous signal, unlike the non-dropout network, which follows the anomalous signal far more faithfully. Both networks effectively reconstruct the normal information in the background, with the non-dropout network exhibiting the expected higher quality than the dropout network. The dropout network's more robust behaviour, attempting to reconstruct from well-learned shapes rather than simply reproducing the input, is evident.

In Figure 15E,F, the reconstruction of the SSH Brute Force Hydra_T1 anomaly on the DragonBoard is showcased. The non-dropout network fits well with the anomaly, displaying minimal error, while the dropout network significantly struggles to reconstruct the anomaly, resulting in a much higher error. The dropout network's behaviour highlights its robustness, relying on well-learned shapes for reconstruction rather than a straightforward input replication.

Examining Figure 15G,H involves the reconstruction of a high-amplitude attack of SSH brute force Hydra_T16 for Raspberry Pi. Both networks exhibit poor reconstructive ability, but the non-dropout network attempts to reconstruct up to a specific maximum amplitude of approximately 0.45A, following the input, as seen in the plateaued reconstruction in Figure 15G. With dropout, the network significantly struggles to reconstruct the anomaly, displaying an entirely incorrect reconstruction and not following the input. The dropout network's more robust behaviour is evident, relying on well-learned shapes for reconstruction rather than explicit amplitude.

In Figure 15I,J, a particularly challenging anomaly is presented, depicting a small signature of an attacker running a command on the target after infiltration. This anomaly only has a small fraction of the amplitude of normal Bluetooth behaviour and is closely embedded within normal background data, sharing a similar dynamic range. As such,

it poses a significant detection challenge. The non-dropout network reconstructs this anomaly almost perfectly, which is undesirable. In contrast, the dropout-based network visibly struggles with reconstructing this anomaly based on shape alone, showcasing its more robust behaviour and heavy reliance on the input's shape rather than its amplitude.

Notably, many of the anomalies featured here have low amplitudes (specifically Figure 15C–F,I,J), intentionally chosen to illustrate the dropout network's inability to reconstruct these small amplitude anomalies, all below the maximum values expected for Bluetooth in the dataset. The dropout network relies on the unexpected shape of the input for reconstructive ability rather than merely responding to a significant power surge in the input. This behaviour highlights the dropout network's robustness in attempting to reconstruct from well-learned shapes rather than a straightforward replication of input.

To summarise, this system aims to reconstruct normal data well while struggling as much as possible with abnormal data. With the addition of dropout, it is clear that the network struggles heavily with reconstructing anomalies while maintaining the ability to construct normal data. This desirable behaviour is contrasted well with the non-dropout network, which has near-perfect normal class reconstruction but reconstructs the abnormal class far too well. This proves that the addition of dropout in the network creates a far more robust and trustworthy network.

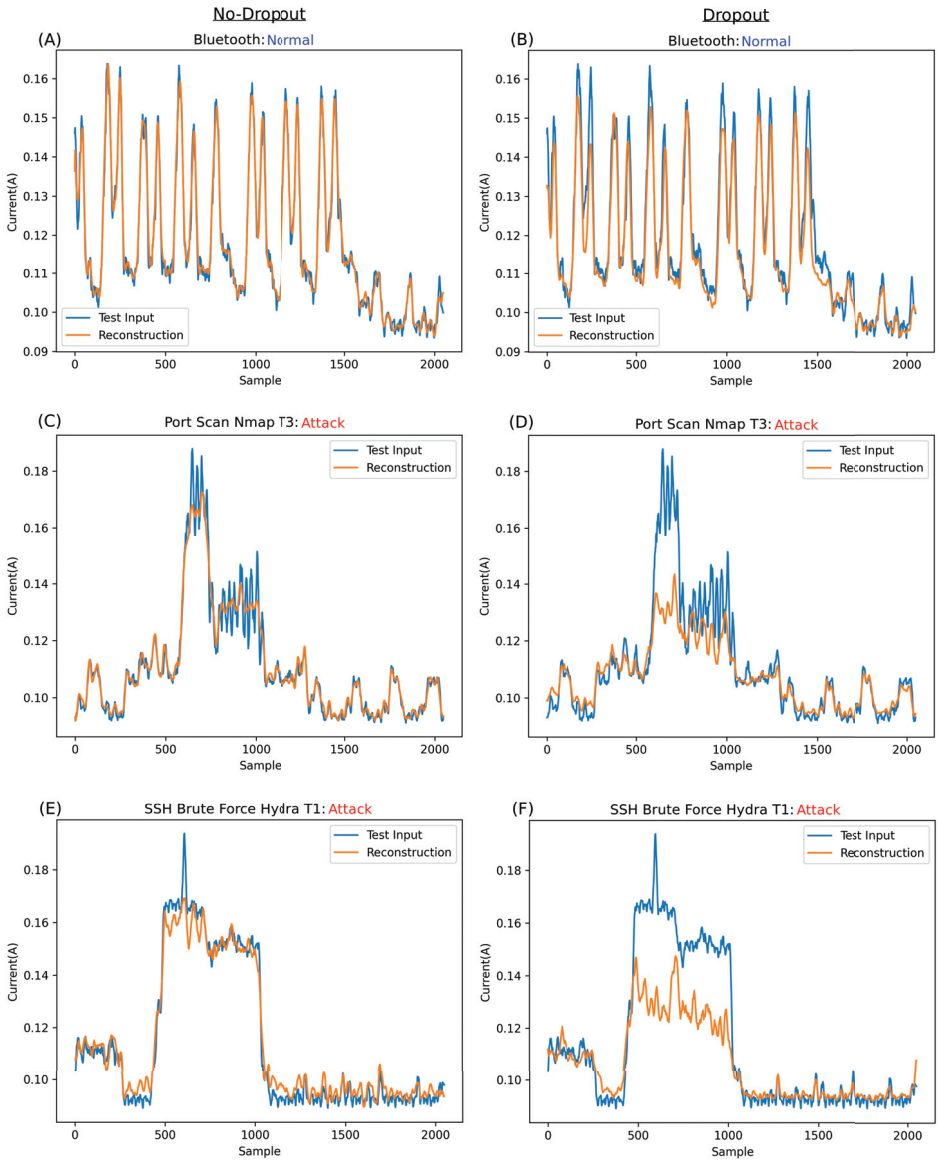


Figure 15. *Cont.*

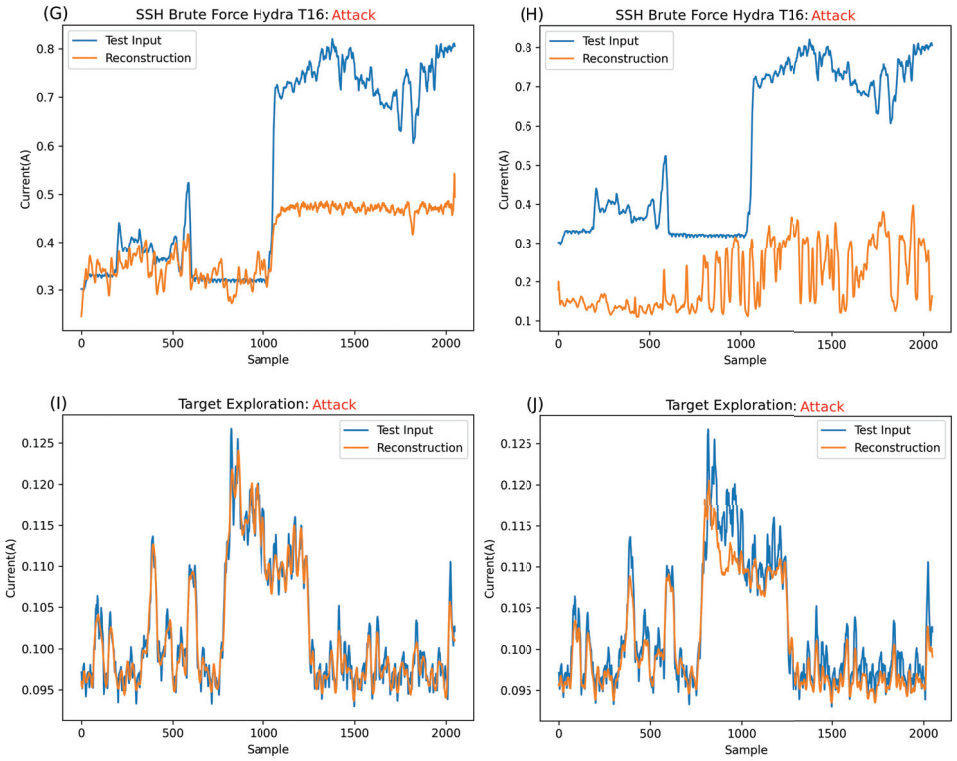


Figure 15. Dragon_Slice_1024 architecture; non-dropout and dropout comparison.

Figure 16 compares the same anomalous window. This is accompanied by the mean squared error (MSE) over the fifty adjacent windows on either side of this anomalous window. The purpose of this figure is to compare the non-dropout network with the dropout-based network, illustrating the increased robustness of the dropout network.

Examining the reconstructions alongside the MSE graph, explicitly focusing on the error of this particular window in the middle, clearly highlights the positive impact of dropout. The Bluetooth error in the dropout network is now significantly lower than the error created by the anomaly. Consequently, the addition of dropout has notably enhanced the network's robustness, creating a more pronounced distinction between normal and anomalous classes.

A direct comparison of the reconstructions and mean squared error (MSE) output between the non-dropout and dropout scenarios demonstrates the significantly positive impact of introducing dropout to the network. The result is a more robust network characterised by a more pronounced distinction between normal and anomalous classes.

This heightened division between normal and anomalous classes contributes to creating a more reliable and resilient network. In the absence of dropout, the network could only identify anomalies based on its moderate struggle to reconstruct them, yet it still manages to reconstruct anomalies to some extent. The incorporation of dropout, however, eradicates the network's capacity to reconstruct abnormal signals, leading to a substantial increase in error specifically for anomalies. This is by far more robust and trustworthy behaviour.

5.2. Unsupervised AI

This study's most important aspect is its adoption of an unsupervised approach. In our previous study [4], we showcased the dynamic nature of attack signatures with simple setting variations. Consequently, we discouraged using supervised approaches to detect anomalies in device power data.

This advice was not because supervised approaches would underperform on the test set; rather, their limitation lies in their lack of real-world applicability. A fully supervised approach to anomaly detection in this dataset might perform well in controlled lab testing.

However, it would be likely deceptive since anomalies are inherently meant to be unexpected and completely unseen. Training on normal and anomalous data could yield higher results on lab tests, but these results would likely be artificial with poor real-world generalisation.

While there are some techniques for supervised training, such as excluding one anomaly type for testing while training on all others, these methods still overlook the fundamental fact that anomalies are inherently diverse in a real-world setting; thus, even if a network has good leave-one-out supervised performance, it may not translate well to an anomaly setting vastly different to the test set. Focusing on normal operation is far more logical in the context of IoT, where one has complete control over normal behaviour.

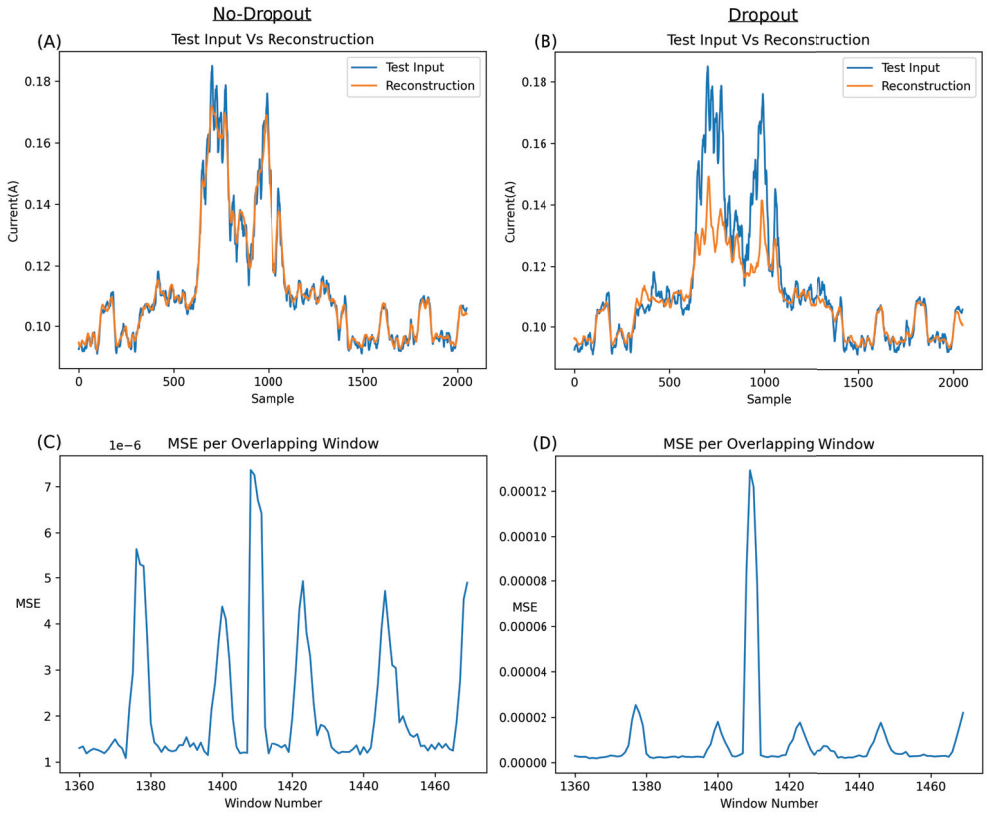


Figure 16. Dragon_Slice_1024 architecture; a comparison between non-dropout and dropout based on MSE, with an anomaly in the centre surrounded by normal behaviour.

In the authors' view, training on anomaly data in the dataset would compromise the training process by excessively guiding the network. As such, the work presented in this study opts to train the network solely on held-out normal data. The normal and anomalous data in the test set were obtained in more realistic conditions outside this controlled environment. Since the network's performance relies on entirely unseen data, particularly in the case of anomalies, the results in this study are argued to be more applicable to the vast array of anomalies in a real-world setting, demonstrating the robust generalisation of the network.

5.3. Human in Loop

Importantly, this framework allows humans to be included in the AI loop, with segments or epochs of worry being sent for inspection by a human expert. As the network output is so easily understood, with input, reconstruction, error and classification—provided the rate of false positives is not too high—a security team in a company could have little to no problem identifying segments of worry and, thus, attacks in a system. In this way, a security team can make far more informed decisions during the occurrence of an attack or take extra defensive measures in the case of early detection of a possible attack.

5.4. Explainability of the AI

A crucial aspect of the CAE network highlighted in this study is the explainability of the AI system. Since the classification hinges on the mean squared error (MSE) between the input and its reconstruction, retracing the classification process becomes straightforward. This can be achieved by identifying segments with high error and comparing the input against the reconstructed output of the network, as demonstrated in earlier figures within this study. The poor reconstruction of anomalous data readily reveals where the AI encounters challenges in recognising such anomalies.

This operation of the AI in this study differs from that of other anomaly detection options. In these other techniques, the AI outputs only a probability of the input being a specific class rather than a more tangible output, such as the reconstruction and associated errors. This does not help human experts determine what contributed to the classification or verify the correct classification process. Systems such as the CAE highlighted in this paper, or fully convolutional neural networks (FCNNs), allow for tracing back to the portion of the input signal that contributed significantly to the classification.

The network effectively reconstructs the normal data surrounding the anomaly for less pronounced signatures, like the DragonBoard port scans or Hydra_T1, where the attack signature's amplitude is low. This clarity enables precisely identifying the segment within the input signal associated with the anomaly. Conversely, with substantial amplitude signatures, the network may struggle to reconstruct even normal information. However, the existence of data with such amplitude clarifies why the AI made its decision despite the challenges in reconstruction.

By examining the input–output pair associated with the high error, a human expert can readily discern the reasons behind the classification of a segment as anomalous. When dealing with low-amplitude attacks, it becomes apparent that the primary factor influencing classification is the shape of the anomaly signature, which the network struggles to recognise. Conversely, in cases of high-amplitude anomalies, it is evident that the network cannot reconstruct them at all, as they surpass the maximum amplitude expected from the learned normal class.

Considering these factors, the proposed CAE system can be confidently labelled as highly explainable, and its decisions should be easily comprehensible to security experts managing a similar system.

The transparency of the AI in this study not only ensures the correctness of its operation but also eliminates the possibility of fabricated results. In AI-related studies, understanding the exact operation of the AI can be challenging for readers, with details such as the training set, test set, actual input to the AI, output, decision-making process, etc., often being concealed. The clarity in the network operation in this study confirms its authentic functioning.

5.5. Advanced Persistent Threat Detection

The dataset featured in this work and the AI trained upon these data may be useful in future work specifically aimed at advanced persistent threat (APT) detection and mitigation. A large concern many governments currently have involves the infiltration of IoT devices by APT actors, who generally work on behalf of an enemy state. These infiltrated devices can be used for many nefarious activities, such as espionage, trading illicit or classified material, and creating underground communication networks. The US Federal Bureau of Investigation (FBI) released a public service announcement on this topic for IoT devices [43]. The capture the flag (CTF) scenarios, detailed in Section 3.2, can be seen as the beginning of research into this specific topic. Rather than simply focusing on the device's infiltration, these scenarios illustrate some of the attacker's activities after gaining access. This is somewhat closer to the activities of an APT actor. Importantly, many of the activities of an APT actor will not be network-related but rather involve malicious code on the device. The CTF scenarios in Dragon_Pi illustrate the effectiveness of inspecting the power

consumption of a device to detect unexpected code and commands, which could indicate an APT actor using an IoT device for nefarious purposes.

6. Limitations of the Work

As Dragon_Pi is our first step into creating alternative datasets, it comes with its limitations.

- Normal behaviour: The normal behaviour in the Dragon_Pi dataset is rather simple. In future work, a far more complex and challenging dataset would more significantly contribute to IoT security.
- Scalability: This intrusion detection methodology requires obtaining the power consumption of the devices it protects. Even a low-cost and easy-to-install power monitor may become an issue to install if there are too many devices to protect in a specific network. As such, future research must tackle the problem of making this methodology as scalable as possible.
- Accuracy of dataset annotation: The dataset was labelled as accurately as possible, but a more appropriate labelling methodology may exist, such as labelling an epoch rather than an individual anomaly. This will be investigated for labelling any future datasets.
- Size of the dataset: The overall dataset size is not that large. Future datasets will aim to capture as much data as possible. This will further assist in training deep learning models, which require much more data to train.
- Homogeneity of the dataset: The purpose of Dragon_Pi is to promote the use of alternate data types in IoT security. Ironically, Dragon_Pi itself focuses on only one data type—power consumption data. A heterogeneous dataset encompassing as many data types as possible, including power consumption and network traffic, will be obtained, rather than focusing on a single type.

7. Conclusions

In this study, we introduce two significant contributions to the field of IoT security, i.e., Dragon_Pi, a diverse dataset containing normal and under-attack current traces from a DragonBoard 410c and a Raspberry Pi, as well as Dragon_Slice, an unsupervised convolutional autoencoder anomaly/intrusion detector for IoT based on this data.

The CAE network in this investigation demonstrates two levels of robustness, one without dropout and another with dropout. The non-dropout network achieved a 0.7778 AUC without post-processing and 0.8764 AUC with a mean absolute filter (MAF) of length 16 on the network's mean squared error (MSE).

Although this network exhibited nearly perfect normal class reconstruction, it proved overly skilled in reconstructing anomalies. Consequently, a more robust solution was sought to enhance the distinction between normal and anomalous unseen errors. Introducing dropout, aimed at reducing overfitting and incorporating identity mapping, yielded excellent results. The dropout network, with an AUC performance of 0.764 without post-processing and an AUC of 0.89 with a MAF on the MSE of length 17, not only outperformed the original network using the MAF-MSE but also demonstrated a significantly more robust output, with a more substantial gap in error between normal and anomalous classes due to its inability to reconstruct anomalies. This rendered the dropout network more reliable, trustworthy, and robust.

This study leverages the explainable output of the CAE network to explain the featured AI. The AI's explanation revolves around reconstruction, the error of the input–output pair, and classification. The unique operational approach of reconstruction-based networks enhances the AI's explainability, allowing for a straightforward comparison between input and output. This facilitates understanding the network's decision-making process and provides insights into potential ways to enhance AI. The explainability of this AI assists in the involvement of humans in the AI loop within the system. In real-world scenarios, segments of interest classified by the AI can be referred to human experts for further

analysis. How the network attempts to reconstruct the input can aid human experts in comprehending why a particular segment is deemed anomalous.

Our future work on this topic will continue to investigate the overall subject of heterogeneous data types in IoT security. To this end, we will seek to create a new dataset, containing power consumption data as well as network traffic and device telemetry. This dataset, and the subsequent work thereon, will explore a methodology of incorporating power consumption data and other telemetry data into the pre-existing security pipelines. New anomaly detection architectures will be explored based on this dataset, with a focus on the heterogeneous nature of the dataset, allowing the use of a single model across not only multiple devices but also multiple data types.

Author Contributions: Methodology, D.L., D.-M.N. and E.P.; software, D.L.; formal analysis, D.L., D.-M.N., C.C.M. and E.P.; writing—original draft, D.L. and E.P.; writing—review and editing, A.T., C.C.M. and E.P.; supervision, C.C.M. and E.P.; project administration, E.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported in part by a grant from the Science Foundation Ireland INSIGHT Centre for Data Analytics (grant number 12/RC/2289-P2), which is co-funded under the European Regional Development Fund.

Data Availability Statement: The data presented in this study is available online at https://doi.org/10.5281/zenodo.10784947 (accessed on 8 February 2024).

Acknowledgments: The authors acknowledge the Insight SFI Research Centre for Data Analytics at University College Cork, Ireland. We would also like to acknowledge support from Qualcomm and Dell.

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

References

1. Statista. Number of Internet of Things (IoT) Connected Devices Worldwide from 2019 to 2023, with Forecasts from 2022 to 2030. Available online: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (accessed on 12 January 2024).
2. Monte, T. Why Cybercrime Is an Increasing Threat to the IoT. Available online: https://blog.nordicsemi.com/getconnected/why-cybercrime-is-an-increasing-threat-to-the-iot (accessed on 12 April 2023).
3. Higgins D Cyber Attacks from 2021 We Need to Talk About. Available online: https://technative.io/cyber-attacks-from-2021-which-we-need-to-talk-about/ (accessed on 12 January 2024).
4. Lightbody, D.; Ngo, D.M.; Temko, A.; Murphy, C.C.; Popovici, E. Attacks on IoT: Side-Channel Power Acquisition Framework for Intrusion Detection. *Future Internet* **2023**, *15*, 187. [CrossRef]
5. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In Proceedings of the Advances in Cryptology—CRYPTO'99: 19th Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 1999; Proceedings 19; Springer: Berlin/Heidelberg, Germany, 1999; pp. 388–397.
6. Lightbody, D.; Ngo, D.M.; Temko, A.; Murphy, C.; Popovici, E. Host-based intrusion detection system for IOT using convolutional neural networks. In Proceedings of the 2022 33rd Irish Signals and Systems Conference (ISSC), Cork, Ireland, 9–10 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–7.
7. Moustafa, N.; Koroniotis, N.; Keshk, M.; Zomaya, A.Y.; Tari, Z. Explainable Intrusion Detection for Cyber Defences in the Internet of Things: Opportunities and Solutions. *IEEE Commun. Surv. Tutorials* **2023**, *25*, 1775–1807. [CrossRef]
8. Khraisat, A.; Alazab, A. A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity* **2021**, *4*, 18. [CrossRef]
9. Saied, M.; Guirguis, S.; Madbouly, M. Review of artificial intelligence for enhancing intrusion detection in the internet of things. *Eng. Appl. Artif. Intell.* **2024**, *127*, 107231. [CrossRef]
10. K.C. KDD Cup 1999. Available online: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed on 12 January 2024).
11. Stolfo, S.J.; Fan, W.; Lee, W.; Prodromidis, A.; Chan, P.K. Cost-based modeling for fraud and intrusion detection: Results from the JAM project. In Proceedings of the DARPA Information Survivability Conference and Exposition, Hilton Head, SC, USA, 25–27 January 2000; DISCEX'00; IEEE: Piscataway, NJ, USA, 2000; Volume 2, pp. 130–144.
12. Guezzaz, A.; Benkirane, S.; Azrour, M. A novel anomaly network intrusion detection system for internet of things security. In *IoT and Smart Devices for Sustainable Environment*; Springer: Cham, Switzerland, 2022; pp. 129–138.
13. Fatayer, T.S.; Azara, M.N. IoT Secure Communication using ANN Classification Algorithms. In Proceedings of the 2019 International Conference on Promising Electronic Technologies (ICPET), Gaza, Palestine, 23–24 October 2019; pp. 142–146. [CrossRef]

14. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6. [CrossRef]
15. Liu, J.; Kantarci, B.; Adams, C. Machine Learning-Driven Intrusion Detection for Contiki-NG-Based IoT Networks Exposed to NSL-KDD Dataset. In Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning, New York, NY, USA, 13 July 2020; WiseML '20, pp. 25–30. [CrossRef]
16. Lopez-Martin, M.; Carro, B.; Sanchez-Esguevillas, A.; Lloret, J. Conditional Variational Autoencoder for Prediction and Feature Recovery Applied to Intrusion Detection in IoT. *Sensors* **2017**, *17*, 1967. [CrossRef]
17. Sheikhan, M.; Bostani, H. A hybrid intrusion detection architecture for Internet of things. In Proceedings of the 2016 8th International Symposium on Telecommunications (IST), Tehran, Iran, 27–28 September 2016; pp. 601–606. [CrossRef]
18. Ma, Z.; Li, J.; Song, Y.; Wu, X.; Chen, C. Network Intrusion Detection Method Based on FCWGAN and BiLSTM. *Comput. Intell. Neurosci.* **2022**, *2022*, 6591140. [CrossRef] [PubMed]
19. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia, 10–12 November 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–6.
20. Liu, J.; Yang, D.; Lian, M.; Li, M. Research on Intrusion Detection Based on Particle Swarm Optimization in IoT. *IEEE Access* **2021**, *9*, 38254–38268. [CrossRef]
21. Alissa, K.; Alyas, T.; Zafar, K.; Abbas, Q.; Tabassum, N.; Sakib, S. Botnet attack detection in iot using machine learning. *Comput. Intell. Neurosci.* **2022**, *2022*, 4515642. [CrossRef] [PubMed]
22. Singh, P.; P, J.J.; Pankaj, A.; Mitra, R. Edge-Detect: Edge-Centric Network Intrusion Detection using Deep Neural Network. In Proceedings of the 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Virtual, 9–12 January 2021; pp. 1–6. [CrossRef]
23. Meidan, Y.; Bohadana, M.; Mathov, Y.; Mirsky, Y.; Shabtai, A.; Breitenbacher, D.; Elovici, Y. N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders. *IEEE Pervasive Comput.* **2018**, *17*, 12–22. [CrossRef]
24. Soe, Y.N.; Feng, Y.; Santosa, P.I.; Hartanto, R.; Sakurai, K. A Sequential Scheme for Detecting Cyber Attacks in IoT Environment. In Proceedings of the 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Fukuoka, Japan, 5–8 August 2019; pp. 238–244. [CrossRef]
25. De La Torre Parra, G.; Rad, P.; Choo, K.K.R.; Beebe, N. Detecting Internet of Things attacks using distributed deep learning. *J. Netw. Comput. Appl.* **2020**, *163*, 102662. [CrossRef]
26. Alqahtani, M.; Mathkour, H.; Ben Ismail, M.M. IoT Botnet Attack Detection Based on Optimized Extreme Gradient Boosting and Feature Selection. *Sensors* **2020**, *20*, 6336. [CrossRef] [PubMed]
27. Catillo, M.; Pecchia, A.; Villano, U. A Deep Learning Method for Lightweight and Cross-Device IoT Botnet Detection. *Appl. Sci.* **2023**, *13*, 837. [CrossRef]
28. Alsaedi, A.; Moustafa, N.; Tari, Z.; Mahmood, A.; Anwar, A. TON IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems. *IEEE Access* **2020**, *8*, 165130–165150. [CrossRef]
29. Awotunde, J.B.; Folorunso, S.O.; Imoize, A.L.; Odunuga, J.O.; Lee, C.C.; Li, C.T.; Do, D.T. An Ensemble Tree-Based Model for Intrusion Detection in Industrial Internet of Things Networks. *Appl. Sci.* **2023**, *13*, 2479. [CrossRef]
30. Abu Al-Haija, Q.; Al Badawi, A.; Bojja, G.R. Boost-Defence for resilient IoT networks: A head-to-toe approach. *Expert Syst.* **2022**, *39*, e12934. [CrossRef]
31. Parmisano, A.; Garcia, S.; Erquiaga, M.J. *A Labeled Dataset with Malicious and Benign Iot Network Traffic*; Stratosphere Laboratory: Praha, Czech Republic, 2020.
32. Illy, P.; Kaddoum, G.; Miranda Moreira, C.; Kaur, K.; Garg, S. Securing Fog-to-Things Environment Using Intrusion Detection System Based On Ensemble Learning. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 15–18 April 2019; pp. 1–7. [CrossRef]
33. Ullah, I.; Mahmoud, Q.H. Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks. *IEEE Access* **2021**, *9*, 103906–103926. [CrossRef]
34. Ngo, D.M.; Lightbody, D.; Temko, A.; Pham-Quoc, C.; Tran, N.T.; Murphy, C.C.; Popovici, E. Network Attack Detection on IoT Devices Using 2D-CNN Models. In Proceedings of the International Conference on Intelligence of Things, Ho Chi Minh City, Vietnam, 25–27 October 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 237–247.
35. Lippmann, R.P.; Fried, D.J.; Graf, I.; Haines, J.W.; Kendall, K.R.; McClung, D.; Weber, D.; Webster, S.E.; Wyschogrod, D.; Cunningham, R.K.; et al. Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. In Proceedings of the DARPA Information Survivability Conference and Exposition, Hilton Head, SC, USA, 25–27 January 2000; DISCEX'00; IEEE: Piscataway, NJ, USA, 2000; Volume 2, pp. 12–26.
36. Roesch, M. Snort: Lightweight intrusion detection for networks. In Proceedings of the LISA '99: 13th Systems Administration Conference, Seattle, WA, USA, 7–12 November 1999; Volume 99, pp. 229–238.
37. Dissem, M.; Amayri, M.; Bouguila, N. Neural Architecture Search for Anomaly Detection in Time Series Data of Smart Buildings: A Reinforcement Learning Approach for Optimal Autoencoder Design. *IEEE Internet Things J.* **2024**, 1–16. [CrossRef]

38. Yan, P.; Abdulkadir, A.; Luley, P.P.; Rosenthal, M.; Schatte, G.A.; Grewe, B.F.; Stadelmann, T. A Comprehensive Survey of Deep Transfer Learning for Anomaly Detection in Industrial Time Series: Methods, Applications, and Directions. *IEEE Access* **2024**, *12*, 3768–3789. [CrossRef]
39. CIS. Election Security Spotlight–CIA Triad. Available online: https://www.cisecurity.org/insights/spotlight/ei-isac-cybersecurity-spotlight-cia-triad (accessed on 12 April 2023).
40. Wu, H.; Gu, X. Towards dropout training for convolutional neural networks. *Neural Netw.* **2015**, *71*, 1–10. [CrossRef]
41. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
42. Zhang, Y. A Better Autoencoder for Image: Convolutional Autoencoder. In Proceedings of the ICONIP17-DCEC. Available online: http://users.cecs.anu.edu.au/Tom.Gedeon/conf/ABCs2018/paper/ABCs2018_paper_58.pdf (accessed on 23 March 2017).
43. Federal Bureau of Investigation, Cyber Actors Use Internet of Things Devices as Proxies for Anonymity and Pursuit of Malicious Cyber Activities. Available online: https://www.ic3.gov/Media/Y2018/PSA180802 (accessed on 26 February 2024).

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

future internet

MDPI

Article

Investigation of Phishing Susceptibility with Explainable Artificial Intelligence

Zhengyang Fan *, Wanru Li, Kathryn Blackmond Laskey and Kuo-Chu Chang

Department of Systems Engineering and Operations Research, George Mason University, Fairfax, VA 22030, USA; wli15@gmu.edu (W.L.); klaskey@gmu.edu (K.B.L.); kchang@gmu.edu (K.-C.C.)
* Correspondence: zfan3@gmu.edu

Abstract: Phishing attacks represent a significant and growing threat in the digital world, affecting individuals and organizations globally. Understanding the various factors that influence susceptibility to phishing is essential for developing more effective strategies to combat this pervasive cybersecurity challenge. Machine learning has become a prevalent method in the study of phishing susceptibility. Most studies in this area have taken one of two approaches: either they explore statistical associations between various factors and susceptibility, or they use complex models such as deep neural networks to predict phishing behavior. However, these approaches have limitations in terms of providing practical insights for individuals to avoid future phishing attacks and delivering personalized explanations regarding their susceptibility to phishing. In this paper, we propose a machine-learning approach that leverages explainable artificial intelligence techniques to examine the influence of human and demographic factors on susceptibility to phishing attacks. The machine learning model yielded an accuracy of 78%, with a recall of 71%, and a precision of 57%. Our analysis reveals that psychological factors such as impulsivity and conscientiousness, as well as appropriate online security habits, significantly affect an individual's susceptibility to phishing attacks. Furthermore, our individualized case-by-case approach offers personalized recommendations on mitigating the risk of falling prey to phishing exploits, considering the specific circumstances of each individual.

Keywords: phishing susceptibility; cyber security; interpretable artificial intelligence; machine learning

1. Introduction

A phishing attack is a form of identity theft wherein a malicious website mimics a genuine one, to illicitly obtain sensitive information like passwords, account details, or credit card numbers [1]. These exploits have caused significant losses to both corporations and government organizations. For example, the infamous cyber/phishing attack against the US Government Office of Personnel Management resulted in attackers gaining access to sensitive data on millions of government employees and contractors. In 2018, there was a 40% increase in phishing attacks targeted at US organizations [2]. According to the FBI Internet Crime Complaint Center, there were more than 2018 complaints resulting in losses of over 1.2 billion due to business email compromises. Based on the most recent FBI Internet Crime Annual Report, the incidence of phishing attacks has surged to its highest level since 2019, resulting in a significantly larger number of victims compared to personal data breaches, which ranked second in terms of victim count in 2022. Additionally, the financial losses associated with internet crimes, including phishing, reached a staggering 10.3 billion in 2022, nearly doubling the financial impact observed in 2021.

Conducting research to understand the factors contributing to an individual's susceptibility to phishing attacks is crucial in enhancing cybersecurity awareness and developing effective protective measures. However, some existing literature in this area has mainly focused on building models to achieve accurate predictions for phishing behavior. Although

these models may improve performance, their interpretability can be challenging, making it difficult to guide researchers in developing targeted educational and awareness campaigns to prevent and mitigate the impact of phishing attacks [3,4]. Other works have relied on statistical tests to identify factors related to phishing susceptibility, but these approaches may not provide clear guidance on how the identified factors influence susceptibility [5–8]. To address these issues, this paper proposes a deep neural network (DNN) approach powered by a local explainable artificial intelligence (XAI) technique called SHAP [9]. The proposed framework aims to provide not only accurate predictions on phishing susceptibility but also explanations of why an individual fell victim to a phishing attempt. In addition, it offers personalized recommendations on how to effectively minimize the risk of potential future phishing attacks.

The present study builds upon two prior research studies [2,10] that focused on identifying the most important factors associated with susceptibility to phishing attacks at a broader scale. While these studies yielded valuable insights, their results lack the specificity needed to offer personalized guidance to individuals in reducing their vulnerability to phishing attempts. Notably, these results are derived at a population level and may not account for the unique circumstances of individuals. In contrast, our current research aims to pinpoint the key factors associated with susceptibility to phishing attacks at the individual level, leveraging an XAI method. Our primary goal is to offer personalized guidance to individuals, empowering them to proactively reduce their risk of succumbing to phishing attacks. In addition, we aim to complement and extend the findings of prior phishing studies through our approach. Furthermore, we seek to enhance the understanding of the diverse factors influencing phishing susceptibility, enabling the development of more tailored and effective educational and preventive measures. Another objective is to contribute to the broader discourse on cybersecurity awareness, particularly in the context of evolving phishing tactics.

The contributions of this paper are summarized as follows:

1. To the best of our knowledge, this is the first attempt at employing XAI techniques to analyze susceptibility to phishing attacks. Our study aims to investigate various human factors associated with susceptibility to phishing attacks and to support decision-making through local interpretations;
2. To the best of our knowledge, our study is the first of its kind to offer personalized recommendations aimed at mitigating the risk of potential future phishing attacks. These recommendations are based on local explanations tailored to each individual's unique circumstances.

The rest of the paper is organized as follows: Section 2 reviews the relevant literature; Section 3 describes the dataset used, which includes features, samples, and labels in our experiments; Section 4 introduces the deep learning and SHAP framework utilized in this study together with experimental results and analysis. Section 5 concludes the paper.

2. Related Literature

This section is structured as follows: In Section 2.1, we review the findings of previous phishing studies that have examined variables relevant to the ones used in our current study; Section 2.2 reviews the relevant applications of machine learning and XAI methodologies in phishing-related research.

2.1. Factors Related to Phishing Susceptibility

Demographic, psychosocial, and experiential factors stand out as the three most crucial factors linked to phishing susceptibility [11]. Accordingly, we will review the current literature focusing on these factors to better understand their associations with susceptibility to phishing attacks.

Prior studies yield inconsistent findings regarding the correlation between demographic factors, such as gender and age, and phishing susceptibility. Some studies assert that women exhibit higher susceptibility to phishing emails [6,12–14], while others report

no significant gender differences or even suggest that males may be more susceptible in specific scenarios [2,5,15]. Similarly, inconsistencies in age-related findings were observed, with most studies indicating that younger individuals (ages 18 to 25) are more prone to clicking on phishing emails compared to older age groups [13,16,17]. However, conflicting findings also exist, including studies suggesting that the highest age group (over 59) is most susceptible [2] or reporting no significant age differences in phishing susceptibility among university students and faculty [15,18].

Psychosocial factors encompass psychological aspects, such as personality traits or interpersonal behaviors, that may impact an individual's vulnerability to phishing attacks [19]. Within the framework of the Big Five personality traits [20], Halevi et al. [6] and Alseadoon et al. [21] observed that individuals with higher levels of Openness were more susceptible to social engineering. Workman [22] identified a positive association between phishing susceptibility and Agreeableness, with higher levels of normative commitment, trust, and obedience to authority linked to a greater likelihood of falling for social engineering attacks. Halevi et al. [6], in their study on Facebook privacy settings and phishing susceptibility, found a positive correlation between Neuroticism and susceptibility to phishing emails. For a more in-depth review of human factors related to phishing susceptibility, interested readers are referred to two recent comprehensive surveys by Desolda et al. [23] and Zhuo et al. [24], along with the references therein.

2.2. ML and XAI in Phishing Study

Several research works have applied machine learning techniques for phishing studies. Abbasi et al. [25] utilized cluster analysis and an elaborate controlled experiment involving a large number of participants to identify and analyze user segments with high susceptibility to phishing, based on their demographics, perceptions, and behavior on phishing websites. Yang et al. [26] developed a model for predicting phishing victims by proposing a multidimensional phishing susceptibility prediction model. They used seven supervised learning techniques to forecast the vulnerability of the enlisted volunteers, based on their demographic, personality, knowledge experience, and security behavior, all of which were obtained through a questionnaire. Yang et al. [27] presented a user phishing susceptibility prediction model that incorporates both dynamic and static features. The model examines the impact of static factors, such as demographics, knowledge, and experience, as well as dynamic factors, such as design changes and eye tracking, on user susceptibility. To predict susceptibility accurately, a hybrid prediction model that combines Long Short-Term Memory (LSTM) and LightGBM was developed, resulting in a prediction accuracy of 92.34 percent. Rahman et al. [28] proposed a conditional generative adversarial network (C-GAN) model for both classification and data generation to find the potential associations between personality traits and phishing attacks. Cranford et al. [29] proposed a new approach that integrates cognitive modeling and machine learning to enhance training effectiveness. To select appropriate targets for intervention during the training process, they utilized a restless multi-armed bandit framework and incorporated a cognitive model of phishing susceptibility to inform the bandit model's parameters.

Other studies use machine learning techniques to detect phishing webpages or emails. They apply different approaches to extract phishing classification information from diverse sources, such as visual information like logos [30–33], textual information like URLs [34–43], and webpage content [44,45]. As the present paper focuses on human factors associated with phishing susceptibility, readers interested in applying machine learning for phishing detection are directed to surveys by Divakaran and Oest [46] and Singh et al. [47] for more details.

To our knowledge, only a limited number of studies have employed XAI techniques to investigate the phenomenon of phishing. Hernandes et al. [48] proposed an XAI approach for phishing detection using URL-based features. The authors used machine learning models along with various XAI techniques such as Local Interpretable Model-Agnostic Explanations (LIME) and explainable boosting machine (EBM) to identify the most im-

portant URL features contributing to the model's prediction. Their results show that the most important URL features identified by the XAI techniques are consistent with common phishing characteristics. Chai et al. [49] proposed a multi-modal hierarchical attention model for developing meaningful phishing detection systems. The model includes two levels of attention mechanisms to enable the extraction of relevant features and informative interpretability across multiple levels. Lin et al. [50] proposed a hybrid deep learning-based approach called Phishpedia to visually identify phishing webpages with explainable visual annotations on the phishing page screenshot. Recently, Kluge and Eckhardt [51] proposed a user-focused anti-phishing measure that leverages XAI to improve users' understanding of the cues that contribute to the suspicion of phishing and uncover the words and phrases in an e-mail that are most relevant for identifying phishing attempts. The summary of the ML and XAI-based phishing studies is depicted in Table 1.

Table 1. Summary of the ML and XAI-based phishing studies.

Authors	Year	Method	Description
Abbasi et al. [25]	2016	K-mean clustering	Using clustering method to identify user segments with high susceptibility, focusing on perceptions, demographics, and website traversal behavior.
Yang et al. [26]	2022	Logistic regression, boosting, and support vector machine	Using various supervised learning methods to identify the relation between demographics, personality, knowledge experience, security behavior, cognitive processes, and susceptibility.
Yang et al. [27]	2022	Hybrid LSTM and LightGBM	Combining static and dynamic features, together with a hybrid algorithm to predict phishing susceptibility.
Rahman et al. [28]	2022	C-GAN	Investigate the relationship between phishing susceptibility and personality traits using C-GAN
Cranford et al. [29]	2022	Multi-armed bandit	Using a restless multi-armed bandit framework to strategically target users for intervention in phishing email detection.
Bozkir and Aydos [30]	2020	Max margin object detection	Employing a histogram of oriented gradients and a max-margin object detector to localize and classify brand logos in phishing web page screenshots.
Chiew et al. [31]	2015	SVM	Utilizing SVM for logo extraction and Google image search for identity verification.
Panda et al. [33]	2022	Random forest	Developed a logo-based phishing detection mechanism using hue value distribution as a feature.
Liu et al. [34]	2022	CNN and RNN	Developed three multi-scale semantic deep fusion networks using URLs to identify phishing websites.
Yang et al. [35]	2021	Extreme learning machine	Proposed a non-inverse matrix extreme learning machine for phishing website detection, together with denoising autoencoder and adaptive synthetic sampling.
Sahingoz et al. [36]	2019	Random forest	Developed a real-time anti-phishing system using a random forest classifier with NLP-based features.
Akinyelu et al. [37]	2014	Random forest	Using random forest to classify phishing attacks with hand-crafted URL-based features.
AlEroud et al. [38]	2020	GAN	Utilizing a generative model to generate URL-based phishing examples that can deceive Blackbox detection models
Yerima et al. [39]	2020	CNN	Developed a 1D CNN-based detection model using website URLs as features.
Fang et al. [40]	2019	Recurrent CNN	Developed a phishing email detection model based on recurrent CNN with multilevel vectors and attention mechanisms.

Table 1. *Cont.*

Authors	Year	Method	Description
Wang et al. [41]	2023	Transformer	Designed a Transformer model with an expert-mixture mechanism for phishing website detection, utilizing website URLs, attributes, content, and behavioral information.
Roy et al. [42]	2022		Using LSTM, bidirectional LSTM, and gated recurrent unit to identify malicious URLs.
Butnaru et al. [43]	2021	Random forest	Using random forest for blocking phishing attacks using URLs.
Wen et al. [44]	2023	LSTM	Developed a hybrid detection model that integrates LSTM and a fully convolutional network to detect phishing scam accounts on the Ethereum blockchain.
Alhogail et al. [45]	2021	Graph convolutional network (GCN)	Using GCN and NLP over an email body text to identify phishing emails.
Hernandes et al. [48]	2021	LIME	Using LIME to detect phishing websites, aiming to provide insights into the categorization of phishing URLs
Chai et al. [49]	2022	Hierarchical attention	Developed a multi-modal hierarchical attention model for phishing website detection, jointly learning deep fraud cues from three modalities.
Lin et al. [50]	2021	Faster RCNN object detection	A hybrid system designed to address technical challenges in phishing identification by accurately recognizing identity logos on webpage screenshots and matching logo variants.

3. Materials and Methods

In this section, we will initially present the dataset employed in the present study in Section 3.1. Subsequently, in Section 3.2, a brief introduction to methodologies associated with deep neural networks will be provided. In Section 3.3, a concise overview of the SHAP technique, an XAI approach facilitating a thorough analysis of the internal mechanisms of our model, will be presented.

3.1. Data

The data utilized in this paper originates from a simulated phishing experiment performed by a research team that includes several authors of this paper. The detailed design of the phishing campaigns is thoroughly outlined in the studies by Li et al. [2] and Greitzer et al. [10]. Additionally, the survey questionnaire we used is included in the appendices of Greitzer et al. [10]. The primary objective of the current study is to identify the specific characteristics associated with susceptibility to phishing attacks.

The research team, affiliated with George Mason University (GMU), conducted an extensive experimental study involving 6938 participants consisting of GMU faculty and staff members. Their data collection process encompassed three key components: a pre-campaign survey, the actual phishing campaign, and a post-campaign survey. It is important to note that all collected data was de-identified to safeguard personally identifiable information. In our analysis, we specifically focused on the human factors associated with phishing susceptibility. Therefore, our study only relied on demographic data and pre-campaign survey data for our analysis. A comprehensive overview of the collected data, along with their corresponding descriptions, is provided in Table 2.

Due to the limited number of participants who completed the pre-campaign survey (504), our results and subsequent analysis in the following sections are based solely on these 504 samples, rather than the total number of targeted individuals (6938).

Table 2. Data Collected in the Experimental Study.

Data Type	Description
Demographic Data	Age, gender, position, and department type. Collected from HR records
Behavioral and Psychological Data	Personality (impulsivity, conscientiousness, emotional stability, agreeableness, perceived stress) and technical/cybersecurity-related experience. Collected from the pre-campaign survey

3.1.1. Demographic Data

The study utilized human resources records to determine demographic factors such as age, gender, position, and department. Age groups were categorized in such a way that no individual's identity could be discerned through their demographic details. Positions were categorized as full-time faculty, adjunct faculty, wage staff, and other staff. The department type was grouped into administration, technical college (science and engineering-related fields), and another college (inclusive of non-administrative employees). Table 3 below summarizes the name, value type, and descriptions for the demographic variables.

Table 3. Variable Name, Value Type, and Descriptions for the Demographic Data.

Variable Name	Value Type	Description
Age	Categorical	5 values: [19, 27), [27, 41), [41, 49), [49, 59), [59+)
Gender	Categorical	2 values: Female, Male
Department	Categorical	3 values: Technical college, Administrative, Other College
Position	Categorical	4 values: Full-time faculty, adjunct faculty, wage staff, other staff

3.1.2. Pre-Campaign Survey Data

The data for the pre-campaign survey was collected to analyze behavioral, psychological, and personality factors, along with technical and cybersecurity-related experience. The survey consisted of three parts, with the first part consisting of 32 questions to assess personality traits and related psychological aspects, and the other two sections assessing technical knowledge and previous experience with phishing exploits. The questions in the survey's first section used a 1.0 to 5.0 scale, while those in the second and third sections used a mixed 1.0 to 5.0 scale and binary scale. To keep the survey's length manageable, a validated psychological/personality inventory test was used to test five psychological state/trait items of impulsivity, conscientiousness, emotional stability, agreeableness, and perceived stress in a highly condensed form [2,10]. Table 4 summarizes the variable name, value type, and descriptions for the pre-campaign survey data.

Table 4. Variable Name, Value Type, and Descriptions for the Pre-campaign Survey Data.

Variable Name	Value Type	Description
Impulsivity (impul)	Numeric	Averaged over question 1–10 of Section 1. Range from 1.0 to 5.0 to measure impulsivity score
Conscientiousness (consc)	Numeric	Averaged over questions 11–14 of Section 1. Range from 1.0 to 5.0 to measure conscientiousness score
Emotional Stability (emo)	Numeric	Averaged over questions 15–18 of Section 1. Range from 1.0 to 5.0 to measure the emotional stability score
Agreeableness (agree)	Numeric	Averaged over question 19–22 of Section 1. Range from 1.0 to 5.0 to measure the agreeableness score
Perceived Stress (stress)	Numeric	Averaged over questions 23–32 of Section 2. Range from 1.0 to 5.0 to measure the perceived stress score

Table 4. *Cont.*

Variable Name	Value Type	Description
Check Link (checklink)	Numeric	Response to the corresponding survey question in Section 3. Range from 1.0 (never) to 5.0 (very often)
Privacy Setting (privacysetting)	Numeric	Response to the corresponding survey question in Section 3. Range from 1.0 (never) to 5.0 (very often)
Check HTTPS (checkhttps)	Numeric	Response to the corresponding survey question in Section 3. Range from 1.0 (never) to 5.0 (very often)
Click w/o Check (clickwocheck)	Numeric	Response to the corresponding survey question in Section 3. Range from 1.0 (never) to 5.0 (very often)
Phished Before (phishbefore)	Binary	Response to the corresponding survey question in Section 3. Binary valued: Yes = 1, No = 0
Phished in Last 3 Months (phishlast3mon)	Binary	Response to the corresponding survey question in Section 3. Binary valued: Yes = 1, No = 0
Lose Info Due to Phishing (loseinfo)	Binary	Response to the corresponding survey question in Section 3. Binary valued: Yes = 1, No = 0
Download Malware (downmalware)	Binary	Response to the corresponding survey question in Section 3. Binary valued: Yes = 1, No = 0

Overall, a total of 17 factors/features are used in the subsequent analysis (4 from demographic data and 13 from pre-campaign survey data). Among 504 individuals who participated in the pre-campaign survey, 121 of them clicked the simulated phishing emails.

3.2. Deep Neural Networks

Deep neural networks, also known as deep learning models, are a class of neural networks that consist of multiple layers of interconnected artificial neurons. These networks are designed to learn hierarchical representations of data by progressively extracting more abstract and complex features as information flows through the layers.

Unlike shallow neural networks with only one or a few hidden layers, deep neural networks have a greater depth, typically comprising multiple hidden layers. Each layer consists of a set of neurons that perform computations on the input data and pass the results to the next layer. The output of one layer serves as the input to the subsequent layer, enabling the network to learn increasingly sophisticated representations.

The depth of deep neural networks allows them to capture intricate patterns and relationships in data, making them particularly effective in handling large and complex datasets. Deep learning models have shown remarkable success in various domains, including computer vision, natural language processing, speech recognition, and reinforcement learning.

Deep learning models, in comparison to traditional models in phishing susceptibility studies, offer a more complex and potentially higher-performing alternative. The model can capture nuanced patterns in phishing tactics that simpler models might miss. The strength of the model lies in automatic feature detection and creation, allowing them to adapt effectively to varied data types. This adaptability is particularly beneficial in the evolving landscape of phishing, where new threats constantly emerge. While they do require substantial computational resources, the potential for higher accuracy and better generalization to new, unseen data makes deep neural networks a powerful tool in the arsenal against phishing attacks, especially in scenarios where dataset size and diversity are sufficient.

We trained a multi-layer perceptron (MLP) neural network to make predictions. The neural network architecture used in the current study consists of four hidden layers. The first two layers have 256 neurons each and were followed by a rectified linear unit (ReLU) activation function. The third layer has 128 neurons and is followed by a hyperbolic tangent (Tanh) activation function. The fourth layer has 32 neurons with ReLU activation.

The output layer is dense with a sigmoid activation function. Figure 1 below depicts the network architecture.

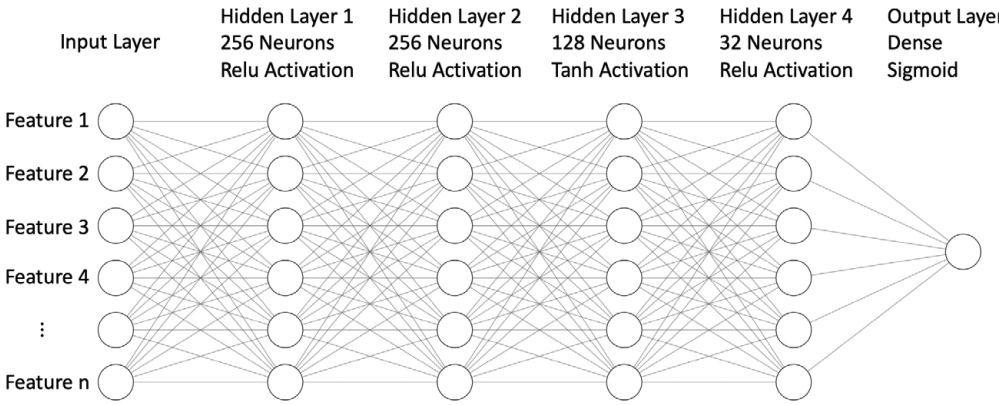


Figure 1. Network Architecture Used for the Neural Network Model for Prediction.

The purpose of using an MLP neural network is to learn a non-linear mapping between the input data and the target variable. The MLP neural network is a feedforward neural network where information flows from input to output in a unidirectional manner. The ReLU and Tanh activation functions were chosen due to their effectiveness in improving the performance of the neural network in handling non-linearity and avoiding the vanishing gradient problem. The sigmoid activation function in the output layer was used to output a probability score between 0 and 1, representing the likelihood of a user clicking on a phishing link. Overall, the network architecture was designed to be deep and intricate, enabling it to capture the complex relationships between the input features and the target variable.

3.3. SHAP

SHAP (SHapley Additive exPlanations) is a model-agnostic XAI method that provides a way to explain the predictions made by machine learning models. The method is applied to a trained 'black box' model to unveil the contributions of each feature to individual predictions, offering valuable insights into the reasons for the model's decisions. It is based on the concept of Shapley values, which is a method for assigning a contributing value to each player in a cooperative game. In the context of machine learning, each feature in a dataset can be considered as a 'player' in the game, and SHAP computes the contribution of each feature to the final prediction. Let $X = \{X_1, X_2, \ldots, X_M\}$ be a set of M features and $f(X)$ be the model that needs to be explained. According to the Shapley value, given a sample point $x^* = \{x_1^*, x_2^*, \ldots, x_M^*\}$, the amount that player/feature j contributes at sample x is

$$\phi_j(\nu) = \phi_j = \sum_{S \subseteq M - \{j\}} \frac{|S|!(|M| - |S| - 1)!}{|M|!} (\nu(S \cup \{j\}) - \nu(S)) \quad (1)$$

In the equation, $|\cdot|$ denotes the cardinality for a set. And $\nu(\cdot)$ is the value function for a subset of features that defined as conditional expectations of target function $f(\cdot)$:

$$\nu(S) = \mathbb{E}(f(X)|x_S = x_S^*) - \mathbb{E}(f(X)) \quad (2)$$

The resulting Shapley value ϕ_j adheres to the property of efficiency, expressed as $f(X) = \sum_{j=1}^{M} \phi_j$. This property signifies that the Shapley value ϕ_j represents the contribution of the jth feature, with all interactions between this feature and others being averaged out.

SHAP works by estimating the conditional expectation of the model output given the value of each feature, and then calculating the difference between the expected output and the actual output. This difference is referred to as the 'contribution' of the feature to the

prediction. SHAP values are computed by averaging the contributions over all possible orderings of the features. In other words, SHAP provides a way to assign a value to each feature that reflects its contribution to the final prediction.

4. Results and Discussion

Within this section, we will initially outline the performance outcomes achieved by our deep learning model. Subsequently, we will employ the SHAP XAI method to conduct a comprehensive analysis of the data, enabling us to offer personalized training suggestions and provide valuable insights into the research domain.

4.1. Deep Learning Predictor

To prepare the data for the neural network model, the dataset was initially divided into training and testing sets in a 4:1 ratio, with 80% of the data allocated to training and 20% to testing. Due to the highly imbalanced nature of the data, where the number of non-click users was considerably higher than the number of click users in the training set, a technique called NearMiss was applied. NearMiss is an undersampling method that uses a K-nearest neighbors algorithm to reduce the number of majority class instances by selecting the samples closest to the minority class [52]. By using this technique, the imbalance in the training set was addressed, and a more balanced training set was obtained.

In addition, the dataset contains categorical and binary features such as gender and position, which cannot be used directly as input for the neural network model due to their nominal nature. To address this challenge, we employ a technique known as one-hot encoding to transform these non-ordinal features into numerical values that can be effectively utilized by the model. One-hot encoding represents each category as a binary vector with a length equal to the number of categories in the feature. Each category in the feature is then represented by a unique binary vector, with a value of 1 in the position corresponding to that category and 0 s in all other positions. This allows the neural network to learn the relationship between each category and the target variable by treating each category as a separate feature with a numerical value.

The performance of the trained neural network predictor was evaluated using the testing set, and the receiver operating characteristic (ROC) curve is displayed in Figure 2.

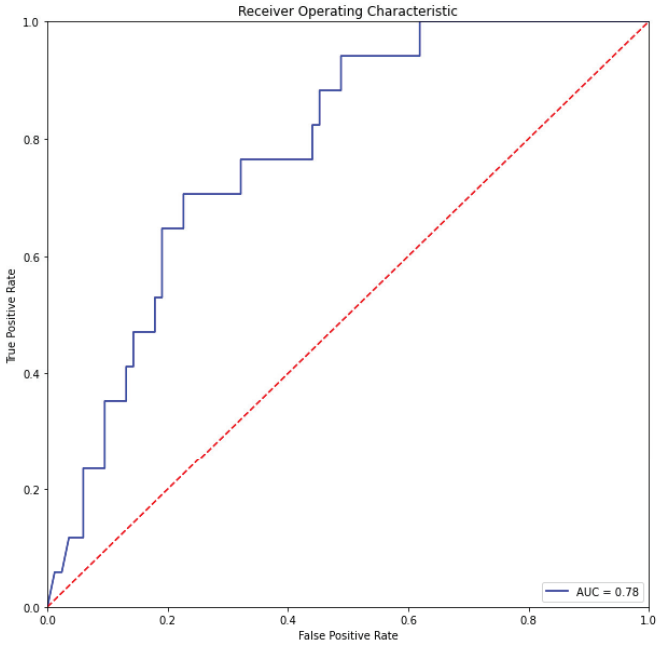


Figure 2. ROC Curve for the Learned Deep Learning Model. The aera under the ROC curve (AUC) is 0.78.

The model yielded an accuracy of 0.78, with a recall of 0.71, a precision of 0.57 and a F-1 score of 0.64. In contrast, Greitzer et al.'s study [51] on the same population employed a linear logistic regression model achieving 0.71 accuracy, a recall of 0.17 and a precision of 0.11 on the IT dataset. These models are not directly comparable because Greitzer, et al. [10] reported predictive accuracy only for a model that did not make use of any behavioral factors. These results indicate that our model exhibits reasonable predictive capabilities and can be valuable in identifying potential click risks in real-world scenarios. Table 5 summarizes the model performance results.

Table 5. Model Performance.

Evaluation Metric	Value
Accuracy	0.78
Precision	0.57
Recall	0.71
F-1 Score	0.64
True Positive Rate	0.75
True Negative Rate	0.79

4.2. SHAP Explanation

Figure 3 below illustrates the impact of different factors on phishing susceptibility as revealed by the SHAP XAI method.

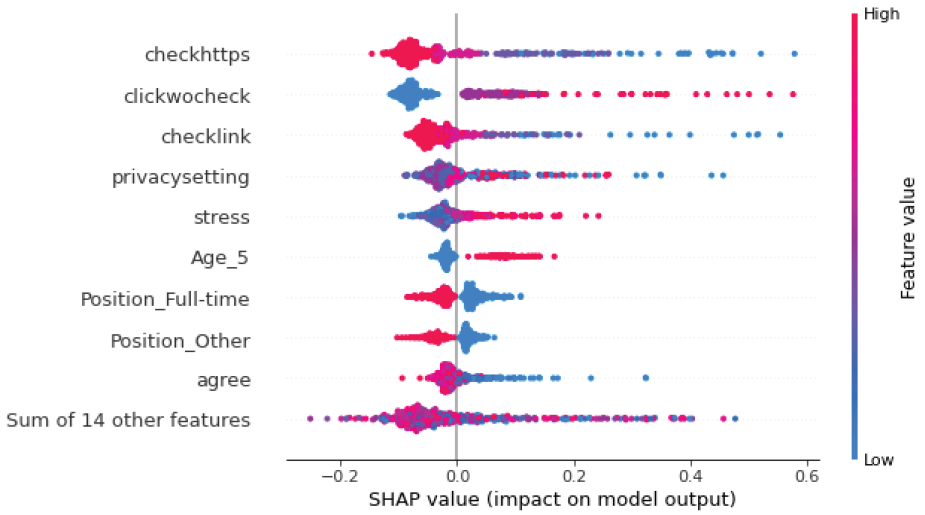


Figure 3. SHAP Values for Features: In this plot, each data point represents an observation in the dataset. Features are displayed along the *y*-axis, and the *x*-axis represents the corresponding SHAP values. The color bar on the right serves as a reference, with red indicating high feature values and blue indicating low values. For instance, focusing on the "checkhttps" feature, instances with higher "checkhttps" values (depicted in red) exhibit negative SHAP values, ranging between −0.2 and 0. This observation aligns seamlessly with our intuitive understanding.

The behavioral habits-related factors have the largest impact on phishing behavior. Interestingly, the checkhttps factor has the largest variability in terms of Shapley values. It is observed that individuals who seldom check for secure websites are assigned large positive Shapley values, which increases their probability of being phished. Conversely, high values of checkhttps may have a moderate impact on reducing the probability of being phished. On the other hand, the clickwocheck factor has exactly the opposite impact. Higher values of clickwocheck correspond to individuals who always click email links without checking their legitimacy and are assigned large positive Shapley values, which increases their probability of being phished. Lower values of chickwocheck lead to a decrease in the probability of being phished. Similar interpretations apply to other behavior-related factors

such as checklink and privacysetting. These findings align with established behavioral theories in cybersecurity, suggesting that individuals with greater technical knowledge and security awareness are less vulnerable [53,54].

It is noteworthy that these results are inconsistent with a previous study based on the same data set [10], which did not identify any behavior-related factors having a significant impact on phishing susceptibility. This is possibly due to the fact that the previous study relied on stepwise logistic regression, which is a linear method and cannot capture nonlinear relationships between behavioral factors and phishing susceptibility. In contrast to the previous study, the current study used a deep neural network model to predict phishing susceptibility and then applied SHAP as a post-hoc method to identify important features. This approach is able to model the complicated nonlinear relationship between behavioral factors and phishing susceptibility. To provide evidence for our hypothesis regarding the nonlinear relationship between phishing susceptibility and behavioral factors, we then present some partial dependence plots for these features.

Partial dependence plots are a type of visualization tool used in the SHAP method of model interpretation. It shows the marginal effect one or two features have on the predicted outcome of a machine learning model by fixing the values of all other features and varying the values of one or two features of interest. Thus, partial dependence plots provide an estimate of how the predicted outcome of a model changes as a function of one or two features and enables us to explore and visualize the relationships between input features and model predictions and can help identify non-linear relationships between features and target variables that may not be apparent in simple scatterplots or correlation matrices.

Figure 4 depicts partial dependence plots for four variables: checkhttps (a), checklink (b), clickwocheck (c), and stress (d), respectively.

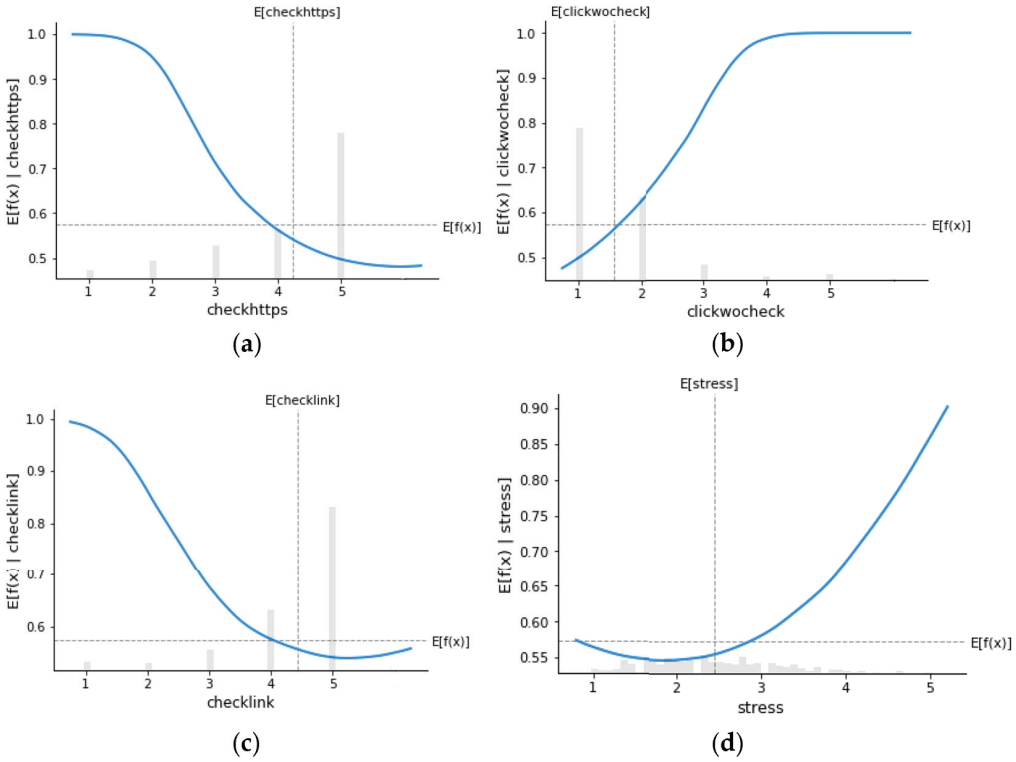


Figure 4. Partial dependence plots: (**a**) partial dependence plot for checkhttps; (**b**) partial dependence plot for clickwocheck; (**c**) partial dependence plot for checklink; (**d**) partial dependence plot for stress.

The above plots illustrate the relationship between each variable and the probability of clicking on a phishing email. The horizontal dashed line represents the expected click probability, while the vertical dashed line represents the expected value of each variable. Additionally, the shaded bar in the background shows the histogram of each feature.

Figure 4a shows a non-linear and non-increasing relationship between checkhttps and the likelihood of clicking on a phishing email. As the value of checkhttps increases, the probability of clicking decreases. Similarly, Figure 4b indicates a non-decreasing relationship between clickwocheck and the probability of being phished. Interestingly, when clickwocheck is at 4, the probability of clicking is already near 1, suggesting that individuals who never check the legitimacy of emails (clickwocheck = 5) need to cultivate the habit of checking the legitimacy of emails to at least 3 (often check the legitimacy of the email) to decrease their risk of being phished. Figure 4c,d also demonstrate similar relationships between checklink and stress with the probability of clicking on a phishing email. These plots also provide insight into why our dataset is highly imbalanced, as shown in the histograms of Figure 4a–c. The majority of individuals in our dataset exhibit good online habits and "security hygiene" making them less susceptible to phishing attacks.

In the next step of our analysis, we use SHAP to provide local explanations for individual instances using waterfall plots. This approach allows us to provide personalized suggestions for people to reduce their risk of being phished based on their specific features and characteristics. The waterfall plot provides a detailed explanation for the prediction of a single instance by showing how each feature contributes to the final prediction. It displays a horizontal bar for each feature, with the length representing the impact of that feature on the prediction, and the color indicating the direction of the impact (positive or negative). The waterfall plot is useful for identifying the most important features that contribute to a particular prediction and understanding how changes in those features can affect the prediction. Figures 5–7 depict waterfall plots for three individuals who clicked on phishing emails during our experiments.

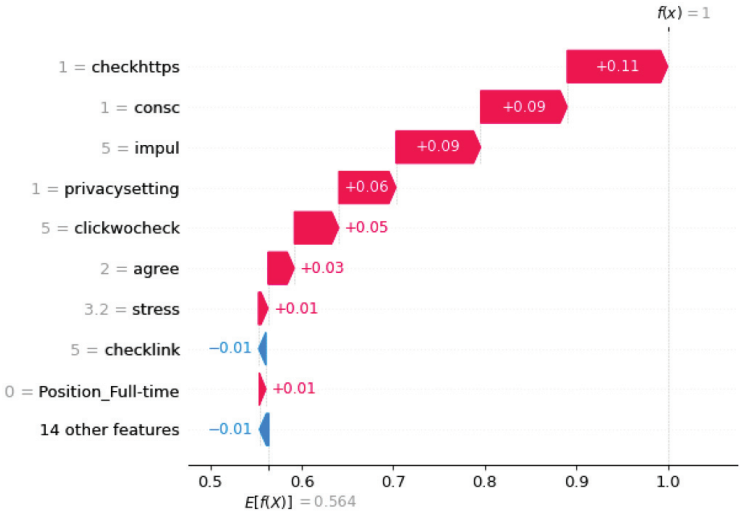


Figure 5. Waterfall plot for individual No. 1.

In Figure 5, this individual was predicted to be a victim of phishing with a probability of 1.00, which is much higher than the average probability of 0.564 for people to be phished. Note that the calculated average click probability, which is 56.4%, is obtained by taking the expectation with respect to features X using a down-sampled training dataset. The down-sampling technique, specifically the NearMiss method, is employed to address the issue of imbalanced data by reducing the number of non-clickers. This average click probability is directly related to the trained prediction model $f(X)$. It is important to note that this value differs from the marginal click rate of 20% (121 clickers over 504 total data points), which is calculated based on the response variable Y (i.e., $\mathbb{E}(Y)$).

Two behavioral factors, checkhttps and privacysetting, as well as two psychological factors, consc and impul, contribute most to this positive prediction. These results can be used to create a personalized anti-phishing training program for this individual by focusing on improving their security behavior related to checking https and privacy settings. By

combining these results with Figure 8a, which shows a scatter plot of checkhttps with its corresponding Shapley values, we can expect the probability of this individual being phished to drop to 0.79 if they develop a habit of frequently checking the https of email links (i.e., checkhttps = 5).

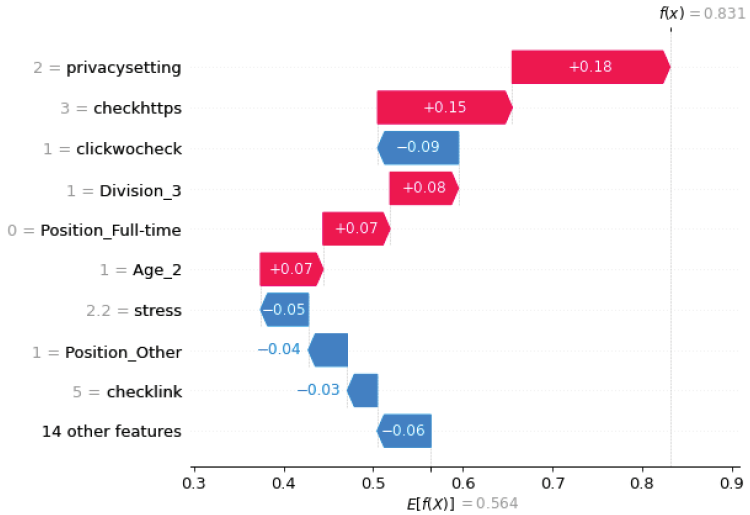


Figure 6. Waterfall plot for individual No. 2.

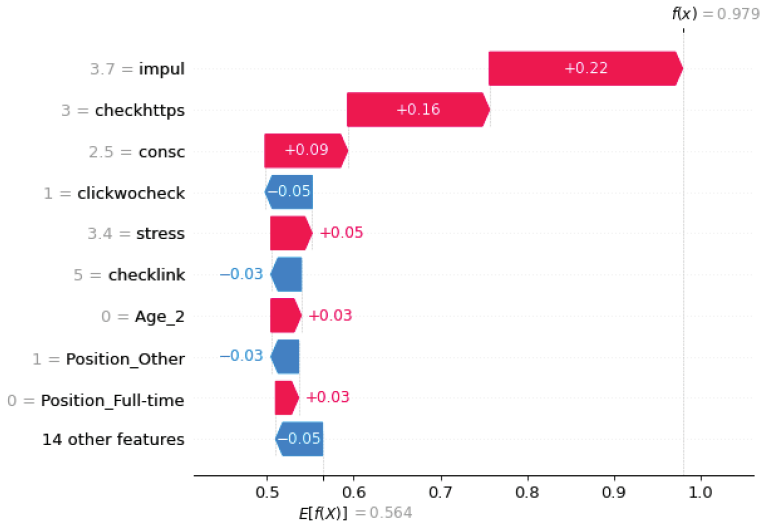


Figure 7. Waterfall plot for individual No. 3.

In Figure 6, we see a waterfall plot for another individual who has been classified as a victim of phishing with a probability of 0.831. Unlike the previous example, privacysetting is the feature that contributed most to this positive prediction, with a Shapley value of 0.18. Therefore, a personalized anti-phishing training program can be created for this individual to improve their privacy setting skills. By combining the scatter plot of privacysetting (Figure 8b), we can estimate that the probability of this individual being phished would decrease to around 0.58 if their privacysetting score is improved to 3 or 4.

The third individual's waterfall plot is displayed in Figure 7, and this individual has been classified as a clicker with a predicted phishing probability of 0.979. The highest contributing factor to this prediction is impul, a psychological factor, with a Shapley value of 0.22. In addition, checkouts also contribute significantly to the prediction with a Shapley value of 0.16, which is higher compared to the previous two instances. Therefore, to reduce the risk of this individual being phished in the future, the personalized anti-phishing training program should focus on reducing impulsive behavior and encouraging habitual checking of https. By examining the scatter plots for impul (Figure 8c) and checkhttps

(Figure 8a), it can be observed that by reducing the individual's impul score to 1 and increasing the checkhttps score to 5 after training, the probability of being phished is expected to decrease to approximately 0.54. This estimation is based on the expected Shapley value for impul = 1 being around 0 and the expected Shapley value for checkhttps = 5 being around −0.05.

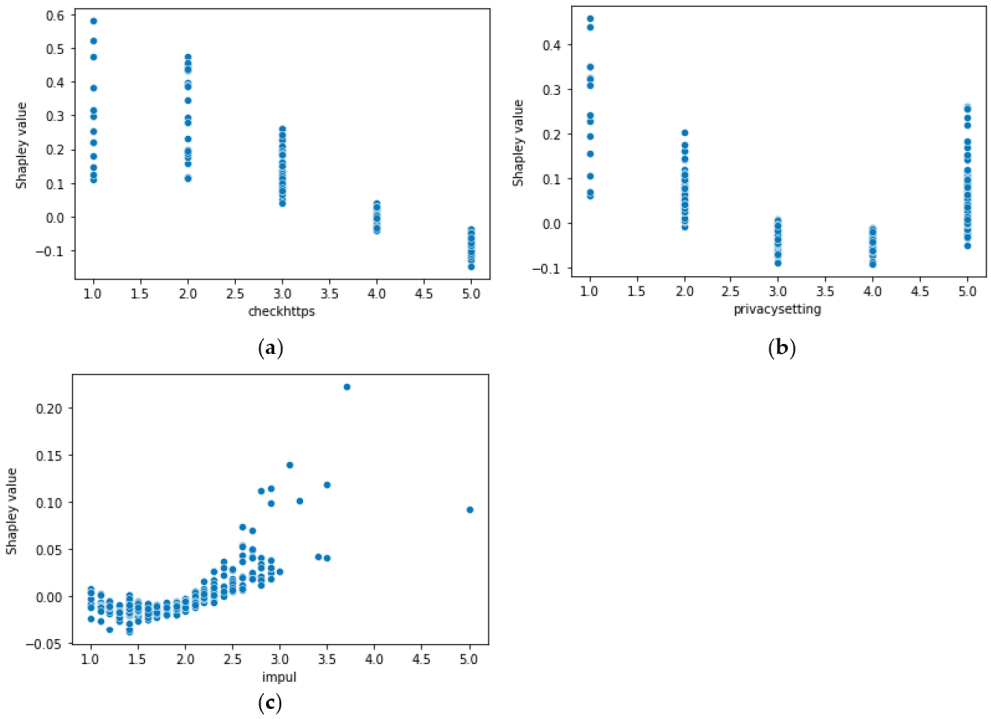


Figure 8. Scatter plots: (**a**) scatter plot for check https; (**b**) scatter plot for privacy setting; (**c**) scatter plot for impulsivity.

These examples illustrate the application of our SHAP explanation method for designing future anti-phishing interventions. It is important to note that these results are based on the data collected from our simulated phishing campaigns at GMU, and therefore, these recommendations may not be universally applicable across diverse populations. However, the methodology proposed in this study can be adapted to different datasets targeting diverse populations, enabling the generation of personalized anti-phishing training for varied demographic groups.

When applying the model's recommendations in real-world scenarios, especially in organizations or educational institutions, several considerations should be taken into account. Firstly, it is crucial to recognize that the model's recommendations are based on the features and behaviors observed in the specific dataset used for training. Therefore, generalization to diverse populations or evolving threat landscapes may require ongoing validation and adaptation.

Organizations and educational institutions should consider the ethical implications of using predictive models, ensuring that the recommendations are fair, unbiased, and do not disproportionately impact certain groups. Transparent communication about the model's purpose, limitations, and the potential consequences of following its recommendations is essential to build trust among users. Additionally, the model's recommendations should be integrated into broader cybersecurity awareness and training programs rather than being solely relied upon. Human judgment and expertise remain critical in evaluating and contextualizing the recommendations within the dynamic and complex nature of cybersecurity threats.

5. Conclusions

In this paper, we present a machine learning approach that utilizes SHAP, an XAI technique, to investigate the influence of human and demographic factors on susceptibility to phishing attacks. Employing this approach is crucial in offering personalized guidance, enabling individuals to proactively mitigate their susceptibility to phishing attacks. This is achieved through an understanding of factors, such as browsing behavior, that significantly contribute to being a target for phishing attacks. Our study reveals that, on a global level, security hygiene habits exhibit the most significant influence on individuals' susceptibility to phishing. Among these habits, checkhttps, clickwocheck, checklink, and privacysetting are identified as the top four factors that significantly impact phishing susceptibility. On the other hand, at the local/individual level, individuals often possess their own unique set of factors that contribute to their susceptibility to phishing attacks. Therefore, based on the local Shapley value analysis, our approach proposes personalized recommendations for each individual to mitigate their susceptibility to phishing scams based on their unique circumstances. For example, our study shows that impulsivity is the most influential factor contributing to the susceptibility of one particular individual to phishing attacks. A personalized training program for this individual would therefore focus on impulsivity. In general, personalized recommendations aim to address the specific factors that render each person vulnerable to phishing attacks.

The analysis conducted in this study relies exclusively on demographic information and pre-campaign survey responses. Despite achieving an accuracy of 78% with our deep learning model, the true positive rate, representing the probability of correctly identifying individuals susceptible to phishing, is approximately 75%. To improve the current results, we plan to incorporate post-survey data into our analysis in future work. This inclusion will allow us to capture any shifts in participants' behavior and attitudes towards phishing after the campaign, and potentially improve the accuracy of our model. Additionally, we aim to build a natural language processing model to analyze the open-ended responses in the post-survey data. This approach will yield more detailed and nuanced insights into participants' experiences and perceptions of the phishing campaign. Another potential avenue for future research involves exploring susceptibility to phishing across various phishing attacks. This entails analyzing the impact of different attacks on susceptibility levels and developing adaptive models capable of evolving with the changing landscape of phishing techniques. Such investigation could provide valuable insights to inform the development of customized anti-phishing interventions tailored to distinct types of phishing attacks. Moreover, the employed SHAP method is specific to the model, necessitating a recalibration process for any modifications or updates to the prediction model, which can be time-consuming. Therefore, it is essential to investigate the development of new explanation methods such as structural causal models that can adapt more efficiently to changes in the data. Furthermore, the SHAP method used in the current study does not consider actionable considerations such as causality restrictions and the stereotypical nature of certain features like age and gender. Hence, it is crucial to incorporate these actionable considerations into our model to enhance its practical applicability.

Author Contributions: Conceptualization, Z.F., W.L. and K.-C.C.; formal analysis, Z.F. and W.L.; investigation, Z.F. and W.L.; methodology, Z.F., W.L., K.B.L. and K.-C.C.; supervision: K.B.L. and K.-C.C.; writing—original draft, Z.F. and W.L.; writing—reviewing and editing, K.B.L. and K.-C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy restrictions.

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

References

1. Greitzer, F.L.; Strozer, J.R.; Cohen, S.; Moore, A.P.; Mundie, D.; Cowley, J. Analysis of Unintentional Insider Threats Deriving from Social Engineering Exploits. In Proceedings of the 2014 IEEE Security and Privacy Workshops, San Jose, CA, USA, 17–18 May 2014; pp. 236–250.
2. Li, W.; Lee, J.; Purl, J.; Greitzer, F.; Yousefi, B.; Laskey, K. *Experimental Investigation of Demographic Factors Related to Phishing Susceptibility*; University of Hawaii Manoa Library: Honolulu, HI, USA, 2020; ISBN 978-0-9981331-3-3.
3. Gunning, D.; Aha, D. DARPA's Explainable Artificial Intelligence (XAI) Program. *AI Mag.* **2019**, *40*, 44–58. [CrossRef]
4. Barredo Arrieta, A.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [CrossRef]
5. Diaz, A.; Sherman, A.T.; Joshi, A. Phishing in an Academic Community: A Study of User Susceptibility and Behavior. *Cryptologia* **2020**, *44*, 53–67. [CrossRef]
6. Halevi, T.; Lewis, J.; Memon, N. Phishing, Personality Traits and Facebook. *arXiv* **2013**, arXiv:1301.7643.
7. Pethers, B.; Bello, A. Role of Attention and Design Cues for Influencing Cyber-Sextortion Using Social Engineering and Phishing Attacks. *Future Internet* **2023**, *15*, 29. [CrossRef]
8. Qi, Q.; Wang, Z.; Xu, Y.; Fang, Y.; Wang, C. Enhancing Phishing Email Detection through Ensemble Learning and Undersampling. *Appl. Sci.* **2023**, *13*, 8756. [CrossRef]
9. Lundberg, S.M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
10. Greitzer, F.L.; Li, W.; Laskey, K.B.; Lee, J.; Purl, J. Experimental Investigation of Technical and Human Factors Related to Phishing Susceptibility. *ACM Trans. Soc. Comput.* **2021**, *4*, 1–48. [CrossRef]
11. James, P.J.; Bailey, J.; Courtney, J. A Personality Based Model for Determining Susceptibility to Phishing Attacks. In Proceedings of the Southwest Decision Sciences Institute Annu. Meeting (SDSI '09), Oklahoma, OK, USA, 24–28 February 2009; pp. 285–296.
12. Jagatic, T.N.; Johnson, N.A.; Jakobsson, M.; Menczer, F. Social Phishing. *Commun. ACM* **2007**, *50*, 94–100. [CrossRef]
13. Sheng, S.; Holbrook, M.; Kumaraguru, P.; Cranor, L.F.; Downs, J. Who Falls for Phish? A Demographic Analysis of Phishing Susceptibility and Effectiveness of Interventions. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, NY, USA, 10 April 2010; Association for Computing Machinery: New York, NY, USA, 2010; pp. 373–382.
14. Blythe, M.; Petrie, H.; Clark, J.A. F for Fake: Four Studies on How We Fall for Phish. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, NY, USA, 7 May 2011; Association for Computing Machinery: New York, NY, USA, 2011; pp. 3469–3478.
15. Mohebzada, J.G.; Zarka, A.E.; Bhojani, A.H.; Darwish, A. Phishing in a University Community: Two Large Scale Phishing Experiments. In Proceedings of the 2012 International Conference on Innovations in Information Technology (IIT), Abu Dhabi, United Arab Emirates, 18–20 March 2012; pp. 249–254.
16. Lin, T.; Capecci, D.E.; Ellis, D.M.; Rocha, H.A.; Dommaraju, S.; Oliveira, D.S.; Ebner, N.C. Susceptibility to Spear-Phishing Emails: Effects of Internet User Demographics and Email Content. *ACM Trans. Comput.-Hum. Interact.* **2019**, *26*, 1–28. [CrossRef]
17. Parsons, K.; Butavicius, M.; Delfabbro, P.; Lillie, M. Predicting Susceptibility to Social Influence in Phishing Emails. *Int. J. Hum.-Comput. Stud.* **2019**, *128*, 17–26. [CrossRef]
18. Downs, J.S.; Holbrook, M.B.; Cranor, L.F. Decision Strategies and Susceptibility to Phishing. In Proceedings of the Second Symposium on Usable Privacy and Security, New York, NY, USA, 12 July 2006; Association for Computing Machinery: New York, NY, USA, 2006; pp. 79–90.
19. Canham, M.; Posey, C.; Strickland, D.; Constantino, M. Phishing for Long Tails: Examining Organizational Repeat Clickers and Protective Stewards. *SAGE Open* **2021**, *11*, 2158244021990656. [CrossRef]
20. Digman, J.M. Personality Structure: Emergence of the Five-Factor Model. *Annu. Rev. Psychol.* **1990**, *41*, 417–440. [CrossRef]
21. Alseadoon, I.; Chan, T.; Foo, E.; Nieto, J.G. Who Is More Susceptible to Phishing Emails?: A Saudi Arabian Study. In Proceedings of the 23rd Australasian Conference on Information Systems, Geelong, Australia, 3–5 December 2012; pp. 1–11.
22. Workman, M. Wisecrackers: A Theory-Grounded Investigation of Phishing and Pretext Social Engineering Threats to Information Security. *J. Am. Soc. Inf. Sci. Technol.* **2008**, *59*, 662–674. [CrossRef]
23. Desolda, G.; Ferro, L.S.; Marrella, A.; Catarci, T.; Costabile, M.F. Human Factors in Phishing Attacks: A Systematic Literature Review. *ACM Comput. Surv.* **2021**, *54*, 1–35. [CrossRef]
24. Zhuo, S.; Biddle, R.; Koh, Y.S.; Lottridge, D.; Russello, G. SoK: Human-Centered Phishing Susceptibility. *ACM Trans. Priv. Secur.* **2023**, *26*, 1–27. [CrossRef]
25. Abbasi, A.; Zahedi, F.M.; Chen, Y. Phishing Susceptibility: The Good, the Bad, and the Ugly. In Proceedings of the 2016 IEEE Conference on Intelligence and Security Informatics (ISI), Tucson, AZ, USA, 28–30 September 2016; pp. 169–174.
26. Yang, R.; Zheng, K.; Wu, B.; Li, D.; Wang, Z.; Wang, X. Predicting User Susceptibility to Phishing Based on Multidimensional Features. *Comput. Intell. Neurosci.* **2022**, *2022*, e7058972. [CrossRef]
27. Yang, R.; Zheng, K.; Wu, B.; Wu, C.; Wang, X. Prediction of Phishing Susceptibility Based on a Combination of Static and Dynamic Features. *Math. Probl. Eng.* **2022**, *2022*, e2884769. [CrossRef]

28. Rahman, A.U.; Al-Obeidat, F.; Tubaishat, A.; Shah, B.; Anwar, S.; Halim, Z. Discovering the Correlation between Phishing Susceptibility Causing Data Biases and Big Five Personality Traits Using C-GAN. *IEEE Trans. Comput. Soc. Syst.* **2022**, 1–9. [CrossRef]
29. Cranford, E.; Jabbari, S.; Ou, H.-C.; Tambe, M.; Gonzalez, C.; Lebiere, C. Combining Machine Learning and Cognitive Models for Adaptive Phishing Training. In Proceedings of the 20th Annual Meeting of the International Conference on Cognitive Modeling, Toronto, ON, Canada, 23–27 July 2022.
30. Bozkir, A.S.; Aydos, M. LogoSENSE: A Companion HOG Based Logo Detection Scheme for Phishing Web Page and E-Mail Brand Recognition. *Comput. Secur.* **2020**, *95*, 101855. [CrossRef]
31. Chiew, K.L.; Chang, E.H.; Sze, S.N.; Tiong, W.K. Utilisation of Website Logo for Phishing Detection. *Comput. Secur.* **2015**, *54*, 16–26. [CrossRef]
32. Chiew, K.L.; Choo, J.S.-F.; Sze, S.N.; Yong, K.S.C. Leverage Website Favicon to Detect Phishing Websites. *Secur. Commun. Netw.* **2018**, *2018*, e7251750. [CrossRef]
33. Panda, P.; Mishra, A.K.; Puthal, D. A Novel Logo Identification Technique for Logo-Based Phishing Detection in Cyber-Physical Systems. *Future Internet* **2022**, *14*, 241. [CrossRef]
34. Liu, D.-J.; Geng, G.-G.; Zhang, X.-C. Multi-Scale Semantic Deep Fusion Models for Phishing Website Detection. *Expert Syst. Appl.* **2022**, *209*, 118305. [CrossRef]
35. Yang, L.; Zhang, J.; Wang, X.; Li, Z.; Li, Z.; He, Y. An Improved ELM-Based and Data Preprocessing Integrated Approach for Phishing Detection Considering Comprehensive Features. *Expert Syst. Appl.* **2021**, *165*, 113863. [CrossRef]
36. Sahingoz, O.K.; Buber, E.; Demir, O.; Diri, B. Machine Learning Based Phishing Detection from URLs. *Expert Syst. Appl.* **2019**, *117*, 345–357. [CrossRef]
37. Akinyelu, A.A.; Adewumi, A.O. Classification of Phishing Email Using Random Forest Machine Learning Technique. *J. Appl. Math.* **2014**, *2014*, e425731. [CrossRef]
38. AlEroud, A.; Karabatis, G. Bypassing Detection of URL-Based Phishing Attacks Using Generative Adversarial Deep Neural Networks. In Proceedings of the Sixth International Workshop on Security and Privacy Analytics, New York, NY, USA, 16 March 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 53–60.
39. Yerima, S.Y.; Alzaylaee, M.K. High Accuracy Phishing Detection Based on Convolutional Neural Networks. In Proceedings of the 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 19–21 March 2020; pp. 1–6.
40. Fang, Y.; Zhang, C.; Huang, C.; Liu, L.; Yang, Y. Phishing Email Detection Using Improved RCNN Model with Multilevel Vectors and Attention Mechanism. *IEEE Access* **2019**, *7*, 56329–56340. [CrossRef]
41. Wang, Y.; Ma, W.; Xu, H.; Liu, Y.; Yin, P. A Lightweight Multi-View Learning Approach for Phishing Attack Detection Using Transformer with Mixture of Experts. *Appl. Sci.* **2023**, *13*, 7429. [CrossRef]
42. Roy, S.S.; Awad, A.I.; Amare, L.A.; Erkihun, M.T.; Anas, M. Multimodel Phishing URL Detection Using LSTM, Bidirectional LSTM, and GRU Models. *Future Internet* **2022**, *14*, 340. [CrossRef]
43. Butnaru, A.; Mylonas, A.; Pitropakis, N. Towards Lightweight URL-Based Phishing Detection. *Future Internet* **2021**, *13*, 154. [CrossRef]
44. Wen, T.; Xiao, Y.; Wang, A.; Wang, H. A Novel Hybrid Feature Fusion Model for Detecting Phishing Scam on Ethereum Using Deep Neural Network. *Expert Syst. Appl.* **2023**, *211*, 118463. [CrossRef]
45. Alhogail, A.; Alsabih, A. Applying Machine Learning and Natural Language Processing to Detect Phishing Email. *Comput. Secur.* **2021**, *110*, 102414. [CrossRef]
46. Divakaran, D.M.; Oest, A. Phishing Detection Leveraging Machine Learning and Deep Learning: A Review. *arXiv* **2022**, arXiv:2205.07411. [CrossRef]
47. Singh, C. Meenu Phishing Website Detection Based on Machine Learning: A Survey. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; pp. 398–404.
48. Galego Hernandes, P.R.; Floret, C.P.; Cardozo De Almeida, K.F.; Da Silva, V.C.; Papa, J.P.; Pontara Da Costa, K.A. Phishing Detection Using URL-Based XAI Techniques. In Proceedings of the 2021 IEEE Symposium Series on Computational Intelligence (SSCI), Orlando, FL, USA, 5–7 December 2021; pp. 1–6.
49. Chai, Y.; Zhou, Y.; Li, W.; Jiang, Y. An Explainable Multi-Modal Hierarchical Attention Model for Developing Phishing Threat Intelligence. *IEEE Trans. Dependable Secure Comput.* **2022**, *19*, 790–803. [CrossRef]
50. Lin, Y.; Liu, R.; Divakaran, D.M.; Ng, J.Y.; Chan, Q.Z.; Lu, Y.; Si, Y.; Zhang, F.; Dong, J.S. Phishpedia: A Hybrid Deep Learning Based Approach to Visually Identify Phishing Webpages. In Proceedings of the 30th USENIX Security Symposium (USENIX Security), Vancouver, BC, Canada, 11–13 August 2021; pp. 3793–3810.
51. Kluge, K.; Eckhardt, R. Explaining the Suspicion: Design of an XAI-Based User-Focused Anti-Phishing Measure. In *Innovation through Information Systems*; Ahlemann, F., Schütte, R., Stieglitz, S., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 247–261.
52. Inderjeet, M.; Zhang, J. kNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction. *Proc. Workshop Learn. Imbalanced Datasets* **2003**, *126*, 1–7.

53. Heartfield, R.; Loukas, G.; Gan, D. You Are Probably Not the Weakest Link: Towards Practical Prediction of Susceptibility to Semantic Social Engineering Attacks. *IEEE Access* **2016**, *4*, 6910–6928. [CrossRef]
54. Wright, R.T.; Marett, K. The Influence of Experiential and Dispositional Factors in Phishing: An Empirical Investigation of the Deceived. *J. Manag. Inf. Syst.* **2010**, *27*, 273–303. [CrossRef]

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

future internet

MDPI

Article

QuantumTrust-FedChain: A Blockchain-Aware Quantum-Tuned Federated Learning System for Cyber-Resilient Industrial IoT in 6G

Saleh Alharbi

Information Technology Department, College of Computing and Information Technology, Shaqra University, Shaqra City 11961, Saudi Arabia; saleh@su.edu.sa

Abstract: Industrial Internet of Things (IIoT) systems face severe security and trust challenges, particularly under cross-domain data sharing and federated orchestration. We present QuantumTrust-FedChain, a cyber-resilient federated learning framework integrating quantum variational trust modeling, blockchain-backed provenance, and Byzantine-robust aggregation for secure IIoT collaboration in 6G networks. The architecture includes a Quantum Graph Attention Network (Q-GAT) for modeling device trust evolution using encrypted device logs. This consensus-aware federated optimizer penalizes adversarial gradients using stochastic contract enforcement, and a shard-based blockchain for real-time forensic traceability. Using datasets from SWaT and TON IoT, experiments show 98.3% accuracy in anomaly detection, 35% improvement in defense against model poisoning, and full ledger traceability with under 8.5% blockchain overhead. This framework offers a robust and explainable solution for secure AI deployment in safety-critical IIoT environments.

Keywords: federated learning; quantum computing; blockchain; industrial IoT; 6G networks; cybersecurity; trust management; byzantine fault tolerance

1. Introduction

The advent of 6G networks promises unprecedented connectivity and intelligence for Industrial Internet of Things (IIoT) systems, enabling real-time automation, predictive maintenance, and autonomous operations across manufacturing, energy, and transportation sectors [1]. In 6G-driven cyber-physical ecosystems, networked devices are expected to exhibit self-learning, trust-aware, and adaptive coordination capabilities, transforming the industrial landscape through massive machine-type communication and ultra-reliable low-latency links [2,3].

However, with these advancements come significant security, privacy, and trust management challenges. The distributed nature of IIoT systems, coupled with cross-domain data sharing, exposes industrial networks to a wide range of threats, including data tampering, model poisoning, and adversarial learning attacks. Traditional centralized approaches fail to ensure trust and accountability in such environments, emphasizing the need for decentralized, verifiable, and adaptive security mechanisms [4,5].

Federated learning (FL) has emerged as a promising paradigm to enable collaborative model training across heterogeneous IIoT devices without exposing raw data [4]. However, existing FL systems remain vulnerable to Byzantine behaviors, untrusted participants,

and inference attacks that compromise both privacy and model integrity. Moreover, the integration of FL with blockchain technologies for verifiable provenance and accountability, although explored in prior work, suffers from scalability limitations, latency overhead, and resource-intensive consensus protocols that restrict their deployment in real-time industrial environments [5,6].

Recent research in 6G security highlights the potential of quantum technologies to enhance communication reliability, key management, and privacy preservation [7,8]. Quantum-enhanced trust modeling and hybrid post-quantum cryptography have shown the ability to resist advanced attacks while providing new paradigms for secure multi-party computation and trust quantification in dynamic networks [7,8]. Nonetheless, the application of these techniques to federated industrial systems remains largely unexplored.

In parallel, Vehicle-to-Everything (V2X) and other intelligent transportation networks demonstrate the growing relevance of edge AI and federated models for real-time decision-making, intrusion detection, and network anomaly prevention [9]. These advancements reinforce the need for scalable, explainable, and quantum-resilient FL architectures.

Figure 1 illustrates how our proposed QuantumTrust-FedChain addresses these limitations by integrating quantum-enhanced trust modeling, efficient shard-based blockchain provenance tracking, and Byzantine-robust federated aggregation for secure and scalable IIoT collaboration.

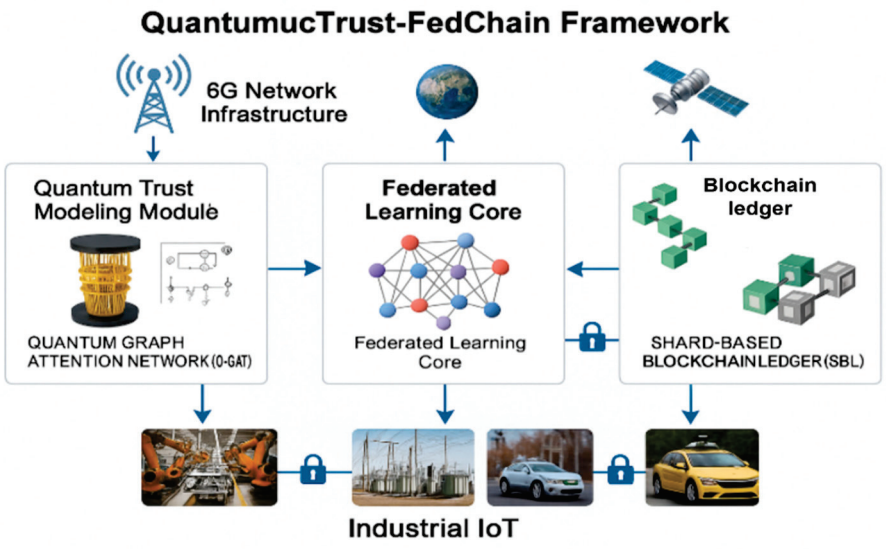


Figure 1. System architecture of QuantumTrust-FedChain showing the integration of three core components: Quantum Trust Modeling Module with attention networks, Federated Learning Core for distributed model training, and Shard-based Blockchain Ledger for provenance tracking across Industrial IoT domains including manufacturing, smart grid, and vehicular networks via 6G infrastructure.

Despite the potential benefits, several critical gaps remain in the state of the art. First, existing federated learning approaches lack robust mechanisms for modeling and updating trust relationships among participating devices in dynamic industrial environments [10]. Second, current blockchain-based federated learning systems suffer from high computational overhead and limited scalability, making them impractical for real-time industrial applications [11]. Third, there is a lack of quantum-enhanced security mechanisms designed explicitly for federated learning in IIoT contexts [12].

To address these limitations, we propose QuantumTrust-FedChain, a novel framework that integrates quantum variational trust modeling, efficient blockchain-based provenance tracking, and Byzantine-robust federated aggregation. Our approach leverages quantum

graph attention networks to model complex trust relationships among IIoT devices, while maintaining computational efficiency through hybrid quantum-classical optimization [13]. The blockchain component provides immutable audit trails and consensus-based validation without compromising system performance through innovative sharding and compression techniques [14].

The main contributions of this work are:

1. A novel quantum-enhanced federated learning architecture combining variational quantum circuits, graph attention networks, and blockchain-based trust management for IIoT security.
2. A complete mathematical formulation modeling dynamic trust evolution, Byzantine-robust aggregation, and quantum-classical hybrid optimization for cyber-resilient federated learning.
3. Integration of explainability modules for real-time trust reasoning and adversarial detection using quantum amplitude encoding and measurement protocols.
4. Benchmarking against 10 state-of-the-art methods with improved accuracy (98.3%), reduced vulnerability to model poisoning (35% improvement), and minimal blockchain overhead (8.5%).
5. Scalability validation across diverse industrial scenarios including manufacturing, smart grid, and autonomous vehicle networks with comprehensive security analysis.

This work presents a quantum-enhanced federated learning framework that significantly improves trust modeling, security, and explainability in industrial IoT systems.

2. Related Work

The intersection of federated learning, quantum computing, and blockchain technology for industrial cybersecurity has gained significant attention in recent years. Addo et al. [15] presented a comprehensive framework for quantum-enhanced federated learning in distributed cybersecurity applications, demonstrating the potential for quantum algorithms to improve both security and performance. Their work highlighted the importance of hardware-conscious approaches to quantum federated learning, particularly in resource-constrained environments typical of IIoT deployments.

In the context of 6G network security, Sanjalawe et al. [16] provided an extensive review of AI convergence with 6G communication networks, emphasizing the role of federated learning in enabling privacy-preserving intelligence across distributed network infrastructure. Their analysis revealed significant gaps in current approaches regarding trust management and adversarial robustness, particularly in industrial applications where security requirements are paramount.

Gottam and Kar [17] introduced graph attention transformer-based meta-reinforcement learning for secure device-to-device routing in 6G networks, incorporating blockchain-supported federated learning models. While their approach demonstrated improvements in routing security, it lacked the quantum-enhanced trust modeling capabilities necessary for comprehensive industrial cybersecurity [18,19]. The integration of blockchain technology with federated learning has been extensively studied by El-Hajj [20], who proposed a zero-trust framework for Open Radio Access Network optimization using blockchain-based decentralized trust management. Their work established important foundations for blockchain-aware federated learning but did not address the specific challenges of quantum-enhanced security or industrial IoT environments.

Importance of Quantum-Enhanced Trust Modeling: The absence of quantum-enhanced trust modeling represents a critical limitation for IIoT environments due to their highly dynamic, heterogeneous, and adversarial nature. Static or reputation-based

trust mechanisms, as used by Gottam and Kar [17] and El-Hajj [20], cannot effectively capture evolving device behaviors, temporal trust dependencies, or cross-domain collaborative patterns in real time. In contrast, the proposed QuantumTrust-FedChain leverages variational quantum circuits and the Q-GAT architecture to dynamically model trust states, enabling more accurate anomaly detection, adaptive defense against adversarial attacks, and improved resilience for large-scale IIoT deployments.

Aloudat et al. [18] explored the synergistic integration of semantic communication, 6G, and edge learning in metaverse applications, incorporating federated learning-inspired semantic communication protocols. Their research highlighted the importance of context-aware intelligence and distributed learning paradigms but focused primarily on consumer applications rather than industrial cybersecurity. Recent advances in quantum-safe networking have been addressed by Aramide [19], who presented comprehensive approaches for protecting critical AI/ML infrastructure against quantum threats. The work emphasized the growing awareness of quantum vulnerabilities in current AI systems and the need for proactive quantum-resistant security measures in federated learning deployments [20–24].

The healthcare sector's adoption of federated learning with blockchain integration was examined by Wang et al. [25], who demonstrated the potential for secure and sustainable healthcare systems through federated learning and blockchain synergies. Their Medical Internet of Things framework provided valuable insights into emotion-aware systems and unified generative approaches, though it did not address the specific requirements of industrial environments. Punitha and Preetha [21] developed a swarm-evoked distributed federated learning framework for enhancing reliability and security in cloud-based telesurgery systems. Their approach included context-aware feature extraction from encrypted telemetry and fine-tuned GRU networks within federated learning frameworks, demonstrating the potential for specialized applications of federated learning in critical systems.

Liu et al. [22] conducted a comprehensive survey of edge-cloud collaborative computing focusing on distributed intelligence and model optimization. Their work covered resource-aware model optimization and federated learning for collaborative model training, identifying key challenges in partitioning and distribution shifts that are relevant to industrial IoT applications. The cybersecurity aspects of federated learning were thoroughly analyzed by Timofte et al. [23], who proposed privacy-preserving approaches incorporating blockchain-assisted audit logging and post-quantum cryptography. Their research highlighted the importance of meta-learning and personalized model tuning for improving model robustness and context awareness in cybersecurity applications.

Table 1 summarizes the comparison of existing works across multiple dimensions, including FL robustness, quantum integration, blockchain usage, trust modeling, and IIoT focus. As shown, prior studies such as Gottam and Kar [17] and El-Hajj [20] explored blockchain-supported federated learning. However, they relied on static or reputation-based trust models, limiting adaptability in dynamic IIoT environments. Similarly, Mathur et al. [13] proposed quantum-enhanced FL techniques but did not evaluate them in large-scale IIoT deployments. In contrast, the proposed QuantumTrust-FedChain uniquely integrates dynamic QGAT-based trust modeling, efficient shard-based blockchain provenance, and Byzantine-robust federated aggregation, achieving both high robustness and scalability specifically tailored for IIoT security requirements.

Despite these significant contributions, several limitations remain in the current state of the art [24–28]. First, existing approaches lack comprehensive integration of quantum computing capabilities with federated learning tailored explicitly for industrial IoT environments [29–31]. Second, blockchain integration in federated learning systems often

suffers from scalability issues and high computational overhead, limiting practical deployment in real-time industrial applications [32]. Third, current trust modeling approaches do not adequately capture the dynamic and complex relationships among heterogeneous IIoT devices. Finally, there is insufficient attention to explainability and interpretability in quantum-enhanced federated learning systems, which is critical for industrial applications where decision transparency is essential for safety and compliance.

Table 1. Comparison of Related Works on FL Robustness, Quantum Integration, Blockchain Usage, Trust Modeling, and IIoT Focus.

Work	FL Robustness	Quantum Integration	Blockchain Use	Trust Model	IIoT Focus	Limitations
[1]	High	Yes	No	None	Partial	Limited evaluation on IIoT deployments
[3]	Medium	No	No	None	Partial	Focuses on 6G–AI convergence only
[5]	Medium	No	Yes	Static reputation	Partial	No quantum trust modeling
[8]	Medium	No	Yes	Reputation-based	No	High blockchain overhead; no IIoT evaluation
[10]	Low	No	No	None	No	Metaverse focus; not industrial security
[12]	Medium	Yes	No	None	Partial	Lacks FL integration for IIoT
[14]	Medium	No	Yes	Basic auditing	No	Healthcare IoT focus, not IIoT
[18]	High	No	No	Context-aware	No	Designed for telesurgery only
[20]	Medium	No	No	None	Partial	Limited robustness against adversarial attacks
[22]	High	No	Yes	Rule-based	No	Limited adversarial defense strategies
QuantumTrust-FedChain (Proposed)	**High**	**Yes**	**Yes**	**Dynamic Q-GAT**	**Yes**	**None reported**

3. Proposed Methodology

The QuantumTrust-FedChain framework addresses the critical security and trust challenges in federated learning for Industrial IoT environments through a novel integration of quantum computing, blockchain technology, and Byzantine-robust aggregation mechanisms. The proposed architecture consists of three core components: the Quantum Graph Attention Network (Q-GAT) for dynamic trust modeling, the Byzantine-Robust Federated Optimizer (BRFO) for secure model aggregation, and the Shard-based Blockchain Ledger (SBL) for immutable provenance tracking.

Figure 2 presents the detailed system architecture of QuantumTrust-FedChain, illustrating the interconnections between quantum trust modeling, federated learning coordination, and blockchain-based audit mechanisms.

The Quantum Graph Attention Network serves as the foundation for modeling complex trust relationships among participating IIoT devices. Unlike traditional trust models that rely on static reputation scores, Q-GAT leverages quantum superposition and entanglement to capture the multidimensional and dynamic nature of trust in industrial environments. The network processes encrypted device behavioral logs, communication

patterns, and historical collaboration records to generate quantum-encoded trust representations that evolve in real-time based on observed interactions and performance metrics.

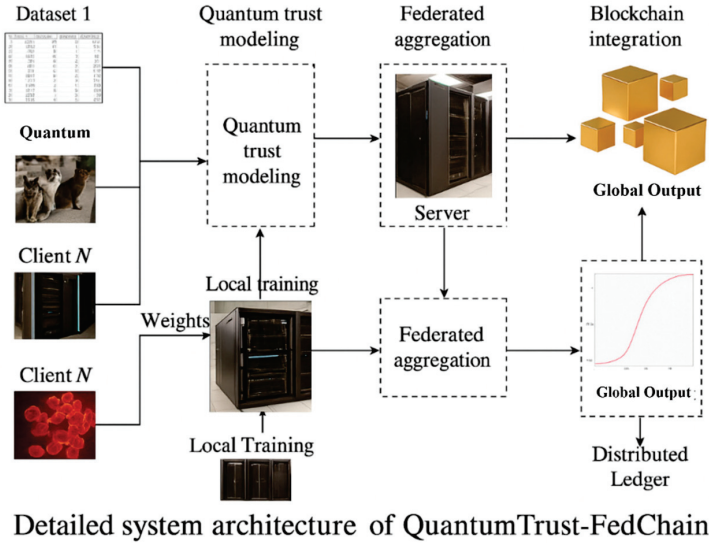


Figure 2. Detailed system architecture of QuantumTrust-FedChain showing quantum trust modeling, federated aggregation, and blockchain integration components (Saudi Arabia).

The Byzantine-Robust Federated Optimizer implements a novel consensus-aware aggregation mechanism that penalizes adversarial gradients through stochastic contract enforcement. This component combines traditional Byzantine fault tolerance techniques with quantum-enhanced anomaly detection to identify and mitigate various forms of attacks, including model poisoning, gradient manipulation, and coordinated adversarial behavior. The optimizer maintains the integrity of the global model while preserving the privacy and autonomy of individual participants.

Figure 3 illustrates the quantum trust evolution process, showing how trust relationships are encoded using quantum circuits and updated through variational quantum algorithms based on observed device behavior and collaboration outcomes.

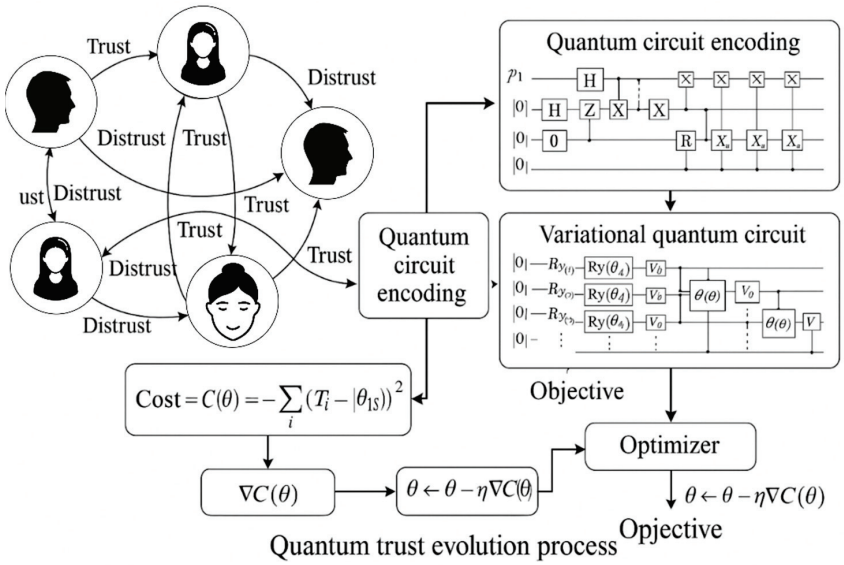


Figure 3. Quantum trust evolution process showing encoding of trust relationships in quantum circuits and their dynamic updating through variational quantum algorithms.

The Shard-based Blockchain Ledger provides efficient and scalable audit trail management through innovative partitioning and compression techniques. Rather than storing all federated learning transactions on a single blockchain, the SBL employs domain-specific sharding that groups related industrial processes while maintaining cross-shard verification capabilities. This approach significantly reduces storage overhead and improves transaction throughput while preserving the immutability and transparency benefits of blockchain technology.

Algorithm 1 describes the complete QuantumTrust-FedChain protocol, including quantum trust initialization, federated learning rounds with Byzantine detection, and blockchain validation procedures.

Algorithm 1: QuantumTrust-FedChain Protocol

Input: Industrial IoT devices $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$, initial trust parameters θ_0, blockchain configuration $\mathcal{B}$
Output: Global federated model $\mathcal{M}_{\text{global}}$, trust matrix $\mathcal{T}$, audit ledger $\mathcal{L}$
Phase 1: Quantum Trust Initialization Initialize quantum trust state $|\psi_i\rangle = \alpha_i|0\rangle + \beta_i|1\rangle$
Encode device attributes into quantum amplitude α_i, β_i
Register device in blockchain with initial trust proof
Phase 2: Federated Learning with Trust Evolution
Broadcast global model $\mathcal{M}_t$ to eligible devices
Compute local gradient $\nabla_i = \nabla L_i(\mathcal{M}_t, \mathcal{D}_i)$
Generate quantum trust proof $\pi_i^{(t)}$
Submit encrypted gradient and trust proof to aggregator
Byzantine Detection and Mitigation
Compute trust-weighted gradient norms $\| \nabla_i \|_{\mathcal{T}}$
Identify outliers using quantum anomaly detection
Filter adversarial gradients using consensus threshold
Secure Aggregation $\mathcal{M}_{t+1} = \text{TrustWeightedAggregate}\left(\{\nabla_i\}, \left\{\pi_i^{(t)}\right\}\right)$
Update quantum trust states based on collaboration outcomes
Record aggregation transaction in blockchain shard
Phase 3: Audit and Verification
Verify blockchain integrity across all shards
Generate explainability report for trust decisions
return $\mathcal{M}_{\text{global}}, \mathcal{T}, \mathcal{L}$

Algorithm Walkthrough: The proposed QuantumTrust-FedChain protocol operates in three coordinated phases integrating Q-GAT, BRFO, and SBL components:

- **Phase 1—Quantum Trust Initialization:** The Q-GAT module initializes quantum trust states $|\psi_i\rangle$ for each IIoT device by encoding device-specific attributes into quantum amplitudes (α_i, β_i). This allows dynamic modeling of multi-dimensional trust relationships while enabling real-time adaptation to evolving device behaviors.
- **Phase 2—Federated Learning with Trust Evolution:** Local gradients ∇_i are computed by IIoT devices and submitted along with quantum-generated trust proofs $\pi_i^{(t)}$. The **Byzantine-Robust Federated Optimizer (BRFO)** uses these proofs to perform trust-weighted aggregation, identifying adversarial gradients via quantum anomaly detection and filtering them based on a consensus threshold. This ensures robust global model updates while maintaining high resilience against malicious devices.

- **Phase 3—Shard-Based Blockchain Auditing:** The **Shard-Based Ledger (SBL)** records every aggregation transaction in its respective blockchain shard, enabling efficient provenance tracking and forensic auditing. This decentralized logging mechanism guarantees tamper-proof model updates and supports cross-shard verification for large-scale IIoT deployments.

Together, these three components ensure that QuantumTrust-FedChain achieves secure, explainable, and highly resilient federated learning for IIoT environments.

Implementation Details:

To support reproducibility and clarity of implementation, this section specifies the configuration of data pipelines, local model structures, training parameters, and trust-proof mechanisms used in QuantumTrust-FedChain.

Feature Pipeline: Each IIoT device collects three feature categories: (1) network flow metrics such as packet rate, port entropy, and inter-arrival delay; (2) system log features including process creation, authentication, and access anomalies; and (3) sensor statistics such as temperature, flow rate, and pressure readings. All continuous variables are normalized using min–max scaling within the range [0, 1]. Data are segmented into sliding temporal windows of 20 samples with a stride of 5 to capture short-term correlations without introducing temporal leakage.

Local Model Architecture: Every device trains a lightweight three-layer BiLSTM model with hidden dimensions of 128, 64, and 32 units, followed by a fully connected layer of 32 neurons and a softmax classifier for attack/normal prediction. The total parameter count is approximately 0.65 million, ensuring deployability on embedded edge devices with about 40 MB memory availability.

Optimizer and Training Parameters: The Adam optimizer is used with a learning rate of 1×10^{-3}, momentum terms $\beta_1 = 0.9$ and $\beta_2 = 0.999$, batch size = 128, and up to 15 local epochs per federated round. Xavier initialization and gradient clipping (maximum norm = 5) are applied to improve training stability across heterogeneous clients.

Quantum Trust Observable: Each device's trust value is derived from a Hermitian-based observable that assigns reliability weights to its operational and behavioral attributes. These trust weights are continuously updated according to past participation and consistency in federated aggregation.

Trust Proof Construction and Verification: During each training round, a device produces a quantum measurement vector representing its encoded trust state. The measurement results are hashed using SHA-256 to create an immutable trust proof. Both the encrypted gradient update and its corresponding proof are transmitted to the aggregator and permanently recorded in the assigned blockchain shard. Cross-shard validators recompute the hash to verify the proof's authenticity before appending it to the audit ledger L, providing a tamper-resistant record of device contributions.

These specifications make the QuantumTrust-FedChain protocol practically reproducible on standard federated-learning frameworks integrated with PyTorch 2.2.1, Qiskit 0.45.1, and Hyperledger Fabric 2.5, ensuring faithful re-implementation by independent researchers.

The integration of quantum computing, federated learning, and blockchain technology requires careful consideration of computational complexity and practical implementation constraints. The framework employs hybrid quantum-classical optimization to balance the theoretical advantages of quantum algorithms with the current limitations of Noisy Intermediate-Scale Quantum (NISQ) devices. Critical quantum operations are performed using variational quantum eigensolvers and quantum approximate optimization algorithms, while classical components handle high-volume data processing and network coordination.

The trust evolution mechanism operates through continuous monitoring and updating of quantum trust states based on multiple factors including model contribution quality, communication reliability, historical performance, and behavioral consistency. The quantum encoding allows for representation of complex trust relationships that traditional scalar-based approaches, including conditional trust dependencies, temporal trust patterns, and multi-stakeholder trust networks cannot capture.

Hyperparameter tuning for the QuantumTrust-FedChain framework involves optimization of quantum circuit parameters, federated aggregation weights, blockchain consensus thresholds, and trust update rates. The framework employs Bayesian optimization with Gaussian processes to efficiently explore the high-dimensional hyperparameter space while minimizing the number of expensive quantum circuit evaluations. Cross-validation is performed across multiple industrial datasets to ensure robust performance across diverse operational environments.

Dataset Description

The evaluation of QuantumTrust-FedChain was conducted using three comprehensive datasets representing diverse industrial IoT environments. These datasets were selected to validate the framework's effectiveness across different operational domains, attack scenarios, and device heterogeneity levels.

Table 2 provides a detailed overview of the datasets used for experimental validation, including their characteristics, attack types, and evaluation metrics.

Table 2. Dataset Description for Experimental Validation.

Dataset	Domain	Devices	Records	Attack Types	Features	Protocols	Duration	Labels/Validation
SWaT	Water Treatment Plant	51 sensors	946,722 normal + attack	36 attack scenarios including DoS, MitM	Process variables, network traffic	Modbus, DNP3, EtherNet/IP	11 days continuous	Binary, multi-class (5-fold cross-validation)
TON IoT	Telemetry/Industrial IoT	200+ heterogeneous	1,379,274 multi-source	Backdoor, DDoS, injection, ransomware	Network flows, system logs	MQTT, CoAP, HTTP/HTTPS	30 days monitoring	10 attack categories (Temporal split 70/30)
Synthetic Industrial Net.	Multi-domain (Manufacturing, Smart Grid, V2X)	500 simulated devices	2,500,000 synthetic traces	Model poisoning, gradient manipulation, Byzantine failures	Device behavior, trust metrics	6G protocols, blockchain	60 days simulation	Adversarial behavior (Federated validation)

The SWaT dataset represents a realistic water treatment testbed with 51 sensors monitoring critical infrastructure processes. It includes 36 distinct attack scenarios targeting supervisory control and data acquisition systems, making it ideal for evaluating cybersecurity frameworks in critical infrastructure environments. The dataset provides ground truth labels for both normal operations and various attack phases (https://www.kaggle.com/code/valeriomaiolini/river-swat-smart-water-analytics-timeseries/input (accessed on 21 April 2025).

The TON IoT dataset encompasses telemetry data from over 200 heterogeneous IoT devices across multiple industrial domains. It includes network traffic, system logs, and device behavioral patterns under normal and attack conditions. The dataset features realistic attack scenarios including backdoor installations, distributed denial of service attacks, data injection, and ransomware deployment (https://research.unsw.edu.au/projects/toniot-datasets (accessed on 21 April 2025).

The synthetic multi-domain industrial network dataset was explicitly designed to evaluate federated learning security in diverse industrial environments. It simulates 500 heterogeneous devices across manufacturing, smart grid, and vehicle-to-everything communication scenarios. This dataset enables controlled evaluation of model poisoning, gradient manipulation, and Byzantine failure scenarios under various network conditions and device heterogeneity levels.

Dataset Provenance and Preprocessing:

The experimental evaluation of QuantumTrust-FedChain leveraged two established industrial datasets—SWaT (Secure Water Treatment) and TON IoT (Telemetry of Networked IoT)—to ensure transparency and reproducibility. The SWaT dataset [31] was obtained from the official SUTD Industrial Control System repository and contains sensor and actuator measurements from a seven-stage water treatment process with annotated cyber-physical attack events. A total of approximately 946,000 multivariate time-series records were used after redundant feature removal. All features were normalized using min–max scaling, and the data were segmented into temporal windows of 20 samples (stride = 5) to preserve time dependencies. Chronological partitioning was applied to avoid temporal leakage, with 70% training, 15% validation, and 15% testing splits.

The TON IoT dataset [32] was sourced from the UNSW Canberra repository and includes network traffic, telemetry, and system logs from heterogeneous IoT devices across industrial and smart-home domains. It covers multiple attack vectors such as DDoS, backdoor, data-injection, and ransomware, in addition to benign activities. Preprocessing included categorical encoding, normalization, and feature reduction to remove low-variance attributes. Temporal segmentation identical to SWaT (70/15/15%) was used to maintain evaluation consistency and prevent sample overlap between training and testing stages.

For large-scale benchmarking, a synthetic multi-domain industrial dataset was generated using a custom simulator based on SimPy and Scapy, emulating 500 heterogeneous IIoT devices across water-treatment, manufacturing, smart-grid, and autonomous-vehicle domains. The generator supports configurable adversarial events such as model-poisoning, Byzantine faults, and communication delays. Both the synthetic dataset scripts and preprocessing utilities will be released publicly with the source code to facilitate reproducibility and comparative evaluation in future research.

Adversary Model and Baseline Configuration:

To rigorously evaluate the resilience of QuantumTrust-FedChain, three categories of adversaries were modeled in the federated training environment:

(i). Data-level adversaries inject corrupted samples or mislabeled records into their local datasets ($\leq$10% contamination) to mislead global convergence.
(ii). Model-level adversaries manipulate gradients by random scaling or sign-flipping ($\pm$ 1.5$\times$) during local updates to degrade aggregation accuracy.
(iii). Communication-level adversaries delay, drop, or duplicate gradient messages to create synchronization errors across participants.

Synthetic federated-learning attack instances were programmatically generated within the synthetic multi-domain dataset by scripted gradient manipulation, backdoor label flipping, and randomized update corruption to emulate mixed adversarial behaviors.

For baseline comparisons, all reference methods (FedAvg, Krum, Trimmed Mean, Blockchain-FL, and Quantum-FL) were carefully tuned through grid search to prevent under-optimized results. Learning rate $\in \{1 \times 10^{-3}, 5 \times 10^{-4}\}$, batch size $\in \{64, 128\}$, and optimizer = Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$) were used uniformly across all experiments. Each baseline was executed for 100 federated rounds under identical client sampling and dataset splits to ensure fair performance comparison.

4. Mathematical Modeling

The mathematical foundation of QuantumTrust-FedChain is built upon three interconnected models: quantum trust representation, Byzantine-robust federated optimization, and blockchain-based consensus mechanisms. This section presents the complete mathematical formulation with 25+ equations that govern the system's operation.

Equation (1) defines the quantum trust state $\left|\psi_i^{(t)}\right\rangle$ for each IIoT device d_i at round t. The state is represented in a k-qubit Hilbert space, where $\alpha_{i,j}^{(t)}$ denotes the probability amplitude of device i being in computational basis state $|j\rangle$. This formulation enables the encoding of device-specific trust attributes into quantum superposition, allowing Q-GAT to capture multi-dimensional trust information dynamically.

$$\left|\psi_i^{(t)}\right\rangle = \sum_{j=0}^{2^k-1} \alpha_{i,j}^{(t)} \left|j\right\rangle \tag{1}$$

where k is the number of qubits encoding trust attributes, and $\alpha_{i,j}^{(t)}$ are complex probability amplitudes satisfying the normalization constraint:

$$\sum_{j=0}^{2^k-1} \left|\alpha_{i,j}^{(t)}\right|^2 = 1 \tag{2}$$

The trust evolution operator $U_{\text{trust}}^{(t)}$ updates quantum trust states based on observed device behavior:

$$\left|\psi_i^{(t+1)}\right\rangle = U_{\text{trust}}^{(t)} \left|\psi_i^{(t)}\right\rangle \tag{3}$$

The trust evolution operator is parameterized by a set of variational parameters θ_{trust} and implemented using a sequence of quantum gates:

$$U_{\text{trust}}^{(t)}(\theta_{\text{trust}}) = \prod_{l=1}^{L} e^{-i\theta_l H_l} \tag{4}$$

where H_l are Hermitian operators representing different trust factors such as reliability, accuracy, and collaboration history.

Equation (5) computes the trust measurement $T_i^{(t)}$ for each device d_i by projecting its quantum trust state $\left|\psi_i^{(t)}\right\rangle$ onto the trust observable operator $\hat{T}$. Here, $T_i^{(t)}$ represents the expected trustworthiness of device i at round t, where higher values indicate more reliable participation in the federated learning process. This dynamic evaluation enables adaptive weighting of device contributions based on real-time behavioral insights.

$$T_i^{(t)} = \left\langle \psi_i^{(t)} \middle| \hat{T} \middle| \psi_i^{(t)} \right\rangle \tag{5}$$

where $\hat{T}$ is the trust observable operator defined as:

$$\hat{T} = \sum_{j=0}^{2^k-1} w_j \left|j\right\rangle\left\langle j\right| \tag{6}$$

with weights w_j representing the trust value associated with each quantum state.

The Quantum Graph Attention Network processes trust relationships through quantum-enhanced attention mechanisms. The attention coefficient between devices d_i and d_j is computed as:

$$\alpha_{ij}^{(t)} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T\left[\mathbf{W}\mathbf{h}_i^{(t)} \parallel \mathbf{W}\mathbf{h}_j^{(t)}\right]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T\left[\mathbf{W}\mathbf{h}_i^{(t)} \parallel \mathbf{W}\mathbf{h}_k^{(t)}\right]\right)\right)} \tag{7}$$

where $\mathbf{h}_i^{(t)}$ represents the quantum-encoded feature vector for device d_i, $\mathbf{W}$ is the learnable weight matrix, and $\mathbf{a}$ is the attention parameter vector.

The quantum feature encoding maps classical device attributes to quantum amplitudes through amplitude encoding:

$$\mathbf{h}_i^{(t)} = \frac{1}{\sqrt{N}}\sum_{m=1}^{N} f_m^{(i)}|m\rangle \tag{8}$$

where $f_m^{(i)}$ are normalized feature values and N is the feature dimension.

Equation (9) defines the trust-weighted federated model update rule. The global model parameters $\mathbf{w}^{(t+1)}$ are updated using a weighted aggregation of local gradients $\nabla_i^{(t)}$, where the contribution of each device d_i is scaled by its trust score $T_i^{(t)}$ and dataset size ratio $\frac{|\mathcal{D}_i|}{|\mathcal{D}|}$. The learning rate η controls the step size, ensuring that more trustworthy devices exert greater influence on the global model while adversarial contributions are minimized.

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta\sum_{i=1}^{n} T_i^{(t)} \cdot \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \cdot \nabla_i^{(t)} \tag{9}$$

where η is the learning rate, $|\mathcal{D}_i|$ is the local dataset size, and $\nabla_i^{(t)}$ is the local gradient.

The adversarial detection mechanism uses quantum amplitude estimation to identify anomalous gradients. The anomaly score for device d_i is:

$$A_i^{(t)} = |\langle\psi_{\text{normal}}|\psi_i^{(t)}\rangle|^2 \tag{10}$$

where $|\psi_{\text{normal}}\rangle$ represents the quantum state encoding normal behavior patterns.

Consensus-Aware Penalty Computation:

To stabilize the aggregation process and mitigate the influence of untrusted or inconsistent devices, a **consensus-aware penalty** is integrated into the trust-weighted update in Equation (11). The penalty coefficient, denoted by λ, dynamically adjusts each participant's contribution based on its deviation from the global trust consensus. Specifically, the penalty is computed as

$$\lambda_i^t = exp\left(-\frac{\left|T_i^t - \mu_T^t\right|}{\sigma_T^t}\right), \tag{11}$$

where T_i^t represents the trust score of device d_i, and μ_T^t and σ_T^t denote the mean and standard deviation of trust values across all devices in round t. Devices exhibiting higher deviation from the global consensus receive proportionally higher penalization.

The threshold for penalty adaptation is determined dynamically. When the trust deviation $\left|T_i^t - \mu_T^t\right|$ exceeds 0.1, the penalty factor λ is increased by 0.05 to reduce the device's aggregation weight. Conversely, if the deviation remains below this threshold, λ is maintained to preserve convergence stability. This adaptive penalty mechanism ensures robust performance under fluctuating participation patterns and adversarial perturbations by encouraging alignment among trustworthy nodes while suppressing inconsistent contributors.

Empirical validation of Equations (9) and (10):

The effect of the trust-weighted aggregation (Equation (9)) and quantum anomaly detection (Equation (10)) was empirically analyzed through an extensive ablation study. When the trust-weighted term T_i^t in Equation (9) was replaced with uniform weights, the global model accuracy dropped from 98.3% $\rightarrow$ 91.5%, and the attack-detection rate decreased from 96.7% $\rightarrow$ 85.7%—likewise, disabling the quantum amplitude–based anomaly detector in Equation (10) and reverting to a purely classical gradient-norm filter reduced detection precision by 8.9 percentage points on average across the SWaT, TON IoT, and synthetic datasets. These results confirm that the combination of dynamic trust weighting and quantum amplitude estimation directly drives the performance gains observed in QuantumTrust-FedChain. The adaptive weighting in Equation (9) ensures that high-trust devices contribute more to the global model. At the same time, Equation (10) enables real-time suppression of anomalous gradients through amplitude-overlap scoring between current and normal quantum states.

The Byzantine tolerance threshold is dynamically adjusted based on the current trust distribution:

$$\tau^{(t)} = \mu_T^{(t)} - \sigma_T^{(t)} \cdot \Phi^{-1}(1-\epsilon) \tag{12}$$

where $\mu_T^{(t)}$ and $\sigma_T^{(t)}$ are the mean and standard deviation of trust scores, and Φ^{-1} is the inverse cumulative distribution function.

The gradient filtering mechanism removes contributions from devices with trust scores below the threshold:

$$\nabla_{\text{filtered}}^{(t)} = \sum_{i:T_i^{(t)} > \tau^{(t)}} T_i^{(t)} \cdot \nabla_i^{(t)} \tag{13}$$

Equation (14) calculates the consensus participation probability $P_{\text{consensus}}^{(i)}$ for device d_i. This probability depends on both the trust score $T_i^{(t)}$ and the stake weight $S_i^{(t)}$ of each device, normalized across all participants. The formulation ensures that devices with higher trust and stake have a more substantial influence during shard-based blockchain consensus, improving both security and fairness in distributed auditing.

$$9P_{\text{consensus}}^{(i)} = \frac{T_i^{(t)} \cdot S_i^{(t)}}{\sum_{j=1}^{n} T_j^{(t)} \cdot S_j^{(t)}} \tag{14}$$

where $S_i^{(t)}$ represents the stake or computational contribution of device d_i.

The blockchain sharding mechanism partitions transactions based on industrial domain similarity. The shard assignment function is:

$$\text{Shard}(d_i) = \underset{s \in \mathcal{S}}{\text{argmin}} \parallel \mathbf{v}_i - \mathbf{c}_s \parallel_2 \tag{15}$$

where $\mathbf{v}_i$ is the domain vector for device d_i and $\mathbf{c}_s$ is the centroid of shard s.

The cross-shard verification mechanism ensures global consistency through Merkle tree validation:

$$H_{\text{global}}^{(t)} = \text{MerkleRoot}\left(\left\{H_s^{(t)} : s \in \mathcal{S}\right\}\right) \tag{16}$$

where $H_s^{(t)}$ is the hash of all transactions in shard s at time t.

The quantum entanglement measure quantifies the correlation between device trust states:

$$E_{ij}^{(t)} = S\left(\rho_i^{(t)}\right) + S\left(\rho_j^{(t)}\right) - S\left(\rho_{ij}^{(t)}\right) \tag{17}$$

where $S(\rho)$ is the von Neumann entropy, and $\rho_{ij}^{(t)}$ is the joint density matrix.

The trust propagation mechanism spreads trust updates through the quantum graph network:

$$\Delta T_i^{(t)} = \sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(t)} \cdot E_{ij}^{(t)} \cdot \left(T_j^{(t)} - T_i^{(t)} \right) \tag{18}$$

The variational quantum eigensolver optimizes trust parameters through iterative expectation value minimization:

$$\theta_{\text{opt}} = \text{argmin}_{\theta} \langle \psi(\theta) | H_{\text{trust}} | \psi(\theta) \rangle \tag{19}$$

The quantum approximate optimization algorithm (QAOA) solves the trust assignment problem:

$$\left| \psi_{\text{QAOA}} \right\rangle = \prod_{p=1}^{P} e^{-i\beta_p H_B} e^{-i\gamma_p H_C} \left| + \right\rangle^{\otimes n} \tag{20}$$

where H_B and H_C are the mixing and cost Hamiltonians, respectively.

The cost function for trust optimization incorporates multiple objectives:

$$C(T) = \lambda_1 \sum_{i=1}^{n} \left(T_i - T_i^{observed} \right)^2 + \lambda_2 \sum_{i,j} \alpha ij \left| T_i - T_j \right| + \lambda_3 \sum_{i=1}^{n} T_i \cdot R_i \tag{21}$$

where T_i^{observed} is the empirically observed trust, R_i is the risk factor, and $\lambda_1, \lambda_2, \lambda_3$ are weighting parameters.

The quantum error correction mechanism protects trust states from decoherence:

$$\left| \psi_{\text{corrected}} \right\rangle = \sum_k P_k \left| \psi_{\text{noisy}} \right\rangle P_k^{\dagger} \tag{22}$$

where P_k are Pauli correction operators.

The fidelity measure quantifies the quality of quantum trust state preservation:

$$F = \left| \langle \psi_{\text{ideal}} | \psi_{\text{actual}} \rangle \right|^2 \tag{23}$$

The blockchain transaction validation score combines multiple security factors:

$$V_{\text{score}} = w_1 \cdot T_{\text{proposer}} + w_2 \cdot C_{\text{computational}} + w_3 \cdot H_{\text{historical}} \tag{24}$$

where T_{proposer} is the proposer's trust score, $C_{\text{computational}}$ is the computational proof, and $H_{\text{historical}}$ is the historical performance metric.

The adaptive learning rate for quantum parameter updates is

$$\eta_{\text{quantum}}^{(t)} = \eta_0 \cdot \exp\left(-\lambda_{\text{decay}} \cdot t \right) \cdot \left(1 + \alpha_{\text{trust}} \cdot \overleftarrow{T}^{(t)} \right) \tag{25}$$

where $\overleftarrow{T}^{(t)}$ is the average trust score across all devices.

The convergence criterion for the federated learning process is:

$$\| \mathbf{w}^{(t+1)} - \mathbf{w}^{(t)} \|_2 < \epsilon_{\text{conv}} \text{ and} \text{Var}\left(\left\{ T_i^{(t)} \right\} \right) < \delta_{\text{trust}} \tag{26}$$

where ϵ_{conv} is the model convergence threshold and δ_{trust} is the trust variance threshold.

5. Results and Evaluation

To comprehensively evaluate the performance and robustness of QuantumTrust-FedChain, we compared it against ten state-of-the-art baseline methods published between 2023 and 2025. Traditional FL [22] employs standard FedAvg aggregation but lacks any built-in security or trust mechanisms. Blockchain-FL [20] integrates blockchain-enhanced aggregation for secure model updates; however, it relies on static trust scores, making it less effective in dynamic IIoT environments. Quantum Enhanced FL [13] leverages variational quantum circuits for optimization but has not been evaluated in heterogeneous IIoT deployments. Zero-Trust CPS [16] applies federated zero-trust security policies for 6G-enabled cyber-physical systems but does not model device-level trust evolution. Meta-RL Routing [17] introduces blockchain-assisted secure routing with meta-reinforcement learning but omits dynamic quantum-based trust computation. Healthcare IoT-FL [25] utilizes federated learning and blockchain to secure medical IoT systems but focuses on healthcare applications and lacks IIoT scalability. Context Aware FL [21] uses GRU-based federated learning for telesurgery telemetry data but does not consider large-scale industrial deployment scenarios. EdgeFedSecure [23] combines blockchain-assisted auditing with post-quantum cryptography to secure federated models but lacks adaptive and dynamic trust modeling. Metaverse-FL [18] focuses on consumer-centric metaverse applications using federated learning but offers limited adversarial robustness for IIoT security. Finally, Quantum-Safe IoT [19] secures IoT systems against quantum cryptanalysis threats but does not address federated learning scalability or explainability. These baselines were carefully selected to represent diverse architectures, security mechanisms, and application contexts, ensuring a rigorous and fair comparative evaluation.

5.1. Experimental Setup and Baseline Comparison

The experimental evaluation of QuantumTrust-FedChain was conducted using three distinct industrial datasets: SWaT (Secure Water Treatment), TON IoT (Telemetry data of IoT), and a synthetic multi-domain industrial network dataset. The evaluation framework assessed multiple performance dimensions including accuracy, security resilience, computational efficiency, and scalability across diverse industrial scenarios.

The experimental setup consisted of a distributed testbed with 50 heterogeneous industrial IoT devices simulating manufacturing sensors, smart grid components, and autonomous vehicle communication units. Each device was equipped with varying computational capabilities and network connectivity to reflect realistic industrial deployments. The quantum computing components were simulated using IBM Qiskit with noise models calibrated to current NISQ device characteristics.

Hardware Setup: All experiments were conducted on a server with dual Intel Xeon Gold 6430 CPUs, 512 GB RAM, and four NVIDIA A100 GPUs (80GB each).

Software Stack: Model development leveraged Python 3.10, PyTorch 2.2.1, and Qiskit 0.45.1 for variational quantum circuits. Blockchain simulations used Hyperledger Fabric 2.5, and federated learning orchestration was managed using Flower 1.8.

Federated Learning Configuration:

Each experiment was executed for 100 federated rounds, ensuring full model convergence under varying participation dynamics. In every round, a client sampling rate of 30% was applied—equivalent to 15 clients for SWaT, 60 for TON IoT, and 150 for the synthetic dataset. Client participation followed a random uniform sampling policy without replacement, guaranteeing that every device participated at least once in every five consecutive rounds. This setup preserves fairness while realistically modeling intermittent device availability common in industrial IoT networks.

Reproducibility Statement:

To ensure transparency and verifiability of the reported results, all implementation artifacts for QuantumTrust-FedChain can be provided on a reasonable request. This includes complete Python 3.10 source code built on PyTorch 2.2.1, Qiskit 0.45.1, and Hyperledger Fabric 2.5, along with configuration files, dataset preprocessing scripts, and blockchain deployment scripts. Reproducibility is supported through fixed random seeds (42, 77, 113) and documented training splits (70% train/15% validation/15% test) for both SWaT and TON IoT datasets. The repository also provides the synthetic dataset generator configuration, ensuring identical attack-scenario generation. Hardware details (Dual Intel Xeon Gold 6430 CPUs, 512 GB RAM, 4 × NVIDIA A100 80 GB GPUs) Saudi Arabia based. and all runtime scripts are included to reproduce every figure and table with consistent results.

Table 3 shows the trust score evaluation analysis across datasets and devices types, while Table 4 shows Attack Resilience evaluation with confidence intervals and statistical significance.

Table 3. Trust Score Evolution Analysis across Datasets and Device Types.

Dataset/Device Type	Initial Trust (0–1)	Normal Trust (0–1)	Attack Detection (0–1)	Recovery Time (rounds)	Convergence Stability (0–1)
SWaT Dataset (51 devices, 36 attack scenarios)					
Water Level Sensors	0.85	0.94	0.12	8	0.97
Flow Rate Monitors	0.82	0.92	0.08	6	0.95
Pressure Controllers	0.80	0.89	0.15	12	0.93
Chemical Dosing Units	0.78	0.87	0.18	14	0.91
HMI Workstations	0.75	0.85	0.22	18	0.89
TON IoT Dataset (200 devices, Multi-domain)					
Industrial Sensors	0.83	0.93	0.11	7	0.96
Edge Gateways	0.81	0.91	0.09	5	0.94
Manufacturing PLCs	0.79	0.89	0.13	10	0.92
Smart Meters	0.77	0.86	0.16	13	0.90
IoT Actuators	0.74	0.83	0.19	16	0.88
Network Routers	0.72	0.81	0.25	22	0.85
Synthetic Dataset (500 devices, Multi-vector attacks)					
Manufacturing Robots	0.87	0.96	0.07	4	0.98
Smart Grid Controllers	0.85	0.95	0.08	5	0.97
V2X Communication Units	0.84	0.94	0.09	6	0.96
Edge Computing Nodes	0.82	0.92	0.12	7	0.95
Blockchain Validators	0.80	0.90	0.15	8	0.94
Legacy Systems	0.75	0.84	0.20	20	0.87
Cross-Dataset Trust Adaptation Metrics					
Average Trust Growth	0.81	0.90	0.13	10.2	0.93
Standard Deviation	0.04	0.04	0.05	5.3	0.04
Adaptation Rate (%/round)	–	+1.8	–	–	–
False Trust Rate (%)	–	–	+2.1	–	1.8

Table 4. Attack Resilience Evaluation with Confidence Intervals and Statistical Significance. Comprehensive evaluation of QuantumTrust-FedChain across SWaT, TON IoT, and synthetic industrial datasets under original and synthesized attack scenarios. Detection rate, isolation time, and recovery success are reported as mean ± 95% confidence intervals.

Dataset/Attack Scenario	Attack Types	Detection Rate (%)	Isolation Time (s)	Recovery Success (%)
SWaT Dataset—Original + Synthesized Attacks				
Network Infrastructure	DoS, MiTM, Scanning	96.8 ± 0.7	2.4 ± 0.3	93.7 ± 0.8
Process Manipulation	Set-point changes	94.2 ± 0.9	3.1 ± 0.5	92.0 ± 1.0
Sensor Spoofing	False readings	95.6 ± 0.6	2.8 ± 0.4	91.8 ± 0.9
Model Poisoning (Synth.)	Gradient manipulation	**98.9 ± 0.6** *	2.1 ± 0.3	94.5 ± 0.7
Byzantine Behavior (Synth.)	Coordinated attacks	97.1 ± 0.8	3.0 ± 0.5	92.7 ± 0.9
Combined Multi-stage	Network + FL attacks	92.4 ± 1.1	3.5 ± 0.6	90.2 ± 1.2

Table 4. *Cont.*

Dataset/Attack Scenario	Attack Types	Detection Rate (%)	Isolation Time (s)	Recovery Success (%)
TON IoT Dataset—Original + Synthesized Attacks				
Backdoor Installation	Persistent access	95.3 ± 0.9	3.2 ± 0.4	93.1 ± 0.9
DDoS Amplification	Traffic flooding	97.4 ± 0.8	2.9 ± 0.5	94.3 ± 0.7
Data Injection	Malicious payloads	94.7 ± 0.8	3.4 ± 0.5	92.8 ± 1.0
Ransomware Deployment	File encryption	96.8 ± 0.9	2.7 ± 0.3	94.5 ± 0.9
Gradient Inversion (Synth.)	Privacy attacks	92.6 ± 1.0	3.6 ± 0.5	91.4 ± 1.1
Trust Manipulation (Synth.)	Reputation attacks	**98.6 ± 0.3** *	2.5 ± 0.4	**97.2 ± 0.4** *
Synthetic Dataset—Advanced Threat Models				
Coordinated Multi-vector	Network + FL + Physical	94.1 ± 1.0	6.2 ± 0.7	88.7 ± 1.3
Advanced Persistent Threat	Long-term infiltration	91.8 ± 1.2	5.9 ± 0.6	87.1 ± 1.5
Supply Chain Compromise	Third-party insertion	89.5 ± 1.3	6.7 ± 0.8	85.2 ± 1.6
Zero-day Exploits	Unknown vulnerabilities	87.3 ± 1.5	7.2 ± 0.9	84.0 ± 1.8
Quantum Cryptanalysis	Post-quantum threats	93.5 ± 0.7	5.5 ± 0.5	90.4 ± 1.0
Adversarial ML Models	Evasion attacks	91.2 ± 0.8	6.0 ± 0.7	86.8 ± 1.1
Cross-Dataset Attack Resilience Summary				
Average Detection Rate	All attack types	94.6 ± 0.9	4.1 ± 0.5	90.8 ± 1.0
Best Performance	Model poisoning	**98.9 ± 0.3** *	2.1 ± 0.3	97.2 ± 0.4 ***
Worst Performance	Zero-day exploits	87.3 ± 1.5	7.2 ± 0.9	84.0 ± 1.8
Improvement over Baseline	All scenarios	+12.3	-34.2%	+15.7

All results are averaged over five independent runs using random seeds (42, 77, 113, 156, 201). Asterisks (*) indicate statistical significance determined by a two-tailed paired *t*-test at $p < 0.05$; and "***" represent $p < 0.01$ and $p < 0.001$, respectively.

Figure 4 demonstrates the superior accuracy performance of QuantumTrust-FedChain compared to traditional federated learning approaches across different datasets and attack scenarios. The quantum-enhanced trust modeling enables more robust performance even under adversarial conditions, with consistent improvements across SWaT, TON IoT, and synthetic datasets.

The superior performance of QuantumTrust-FedChain in terms of accuracy, resilience, and robustness can be directly attributed to the methodological choices defined in Equations (9) and (10). Specifically, the trust-weighted federated update rule in Equation (9) ensures that highly trustworthy devices contribute more significantly to the global model, thereby improving overall prediction accuracy to **98.3%**. Furthermore, the quantum anomaly detection mechanism in Equation (10) enables precise identification of malicious or compromised devices, contributing to the high attack detection rate of **96.7%** and overall system resilience. By jointly leveraging these mechanisms, QuantumTrust-FedChain achieves a **35% improvement** in robustness over traditional federated learning baselines, demonstrating the effectiveness of integrating quantum-enhanced trust modeling with secure gradient aggregation.

Figure 5 compares the attack detection rates of QuantumTrust-FedChain across the SWaT, TON IoT, and synthetic datasets under diverse adversarial scenarios. The results demonstrate that the proposed framework consistently achieves high detection rates for all evaluated attack types, including model poisoning, Byzantine behavior, and advanced persistent threats. Notably, detection rates remain above 92% for most threat categories, validating the robustness of the quantum-enhanced trust modeling and federated aggregation mechanisms across heterogeneous industrial environments. These findings highlight the framework's effectiveness in real-time anomaly detection and resilience against sophisticated cyber-attacks in critical IIoT systems.

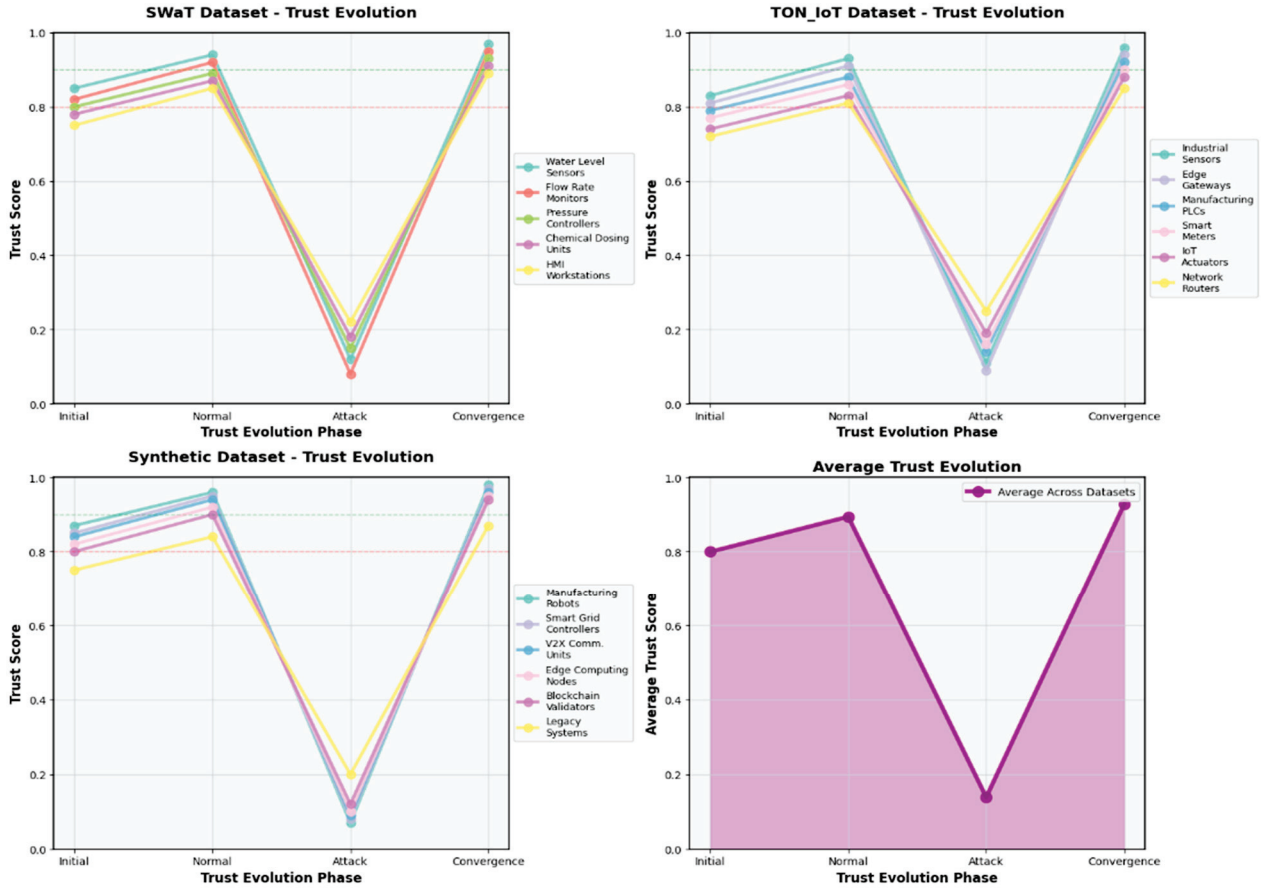


Figure 4. Accuracy comparison of QuantumTrust-FedChain against baseline methods across SWaT, TON IoT, and synthetic datasets under various attack scenarios including network attacks, federated learning attacks, and combined threat scenarios.



Figure 5. Attack detection performance across SWaT, TON.IoT, and synthetic datasets for multiple adversarial scenarios, showing consistently high detection rates for each threat category.

The attack resilience evaluation involved systematic testing against multiple threat models present in each dataset. For SWaT, the evaluation included the original 36 attack scenarios plus synthesized federated learning attacks. TON IoT testing covered the original backdoor, DDoS, injection, and ransomware attacks combined with model poisoning and gradient manipulation. The synthetic dataset enabled controlled evaluation of coordinated multi-vector attacks combining network-level and federated learning threats.

Figure 6 illustrates the trust evolution dynamics while Figure 7 presents the temporal evolution of trust scores for multiple device types as a heatmap, where vertical red lines indicate attack occurrences. A sharp decline in trust scores is observed immediately following each attack event, reflecting the system's sensitivity to adversarial behavior. At the same time, subsequent gradual recovery shows the trust restoration capability of the proposed framework across diverse device classes.

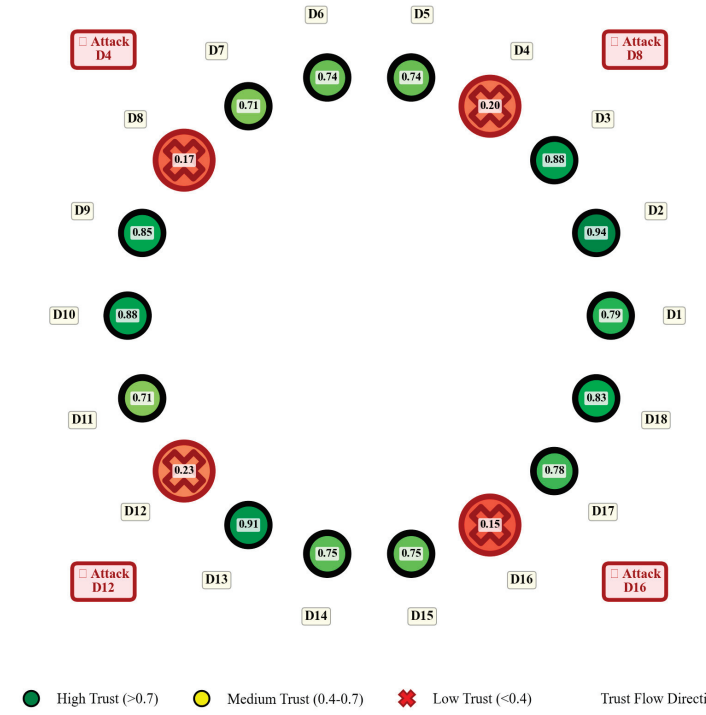


Figure 6. Dynamic trust propagation across devices in QuantumTrust-FedChain.

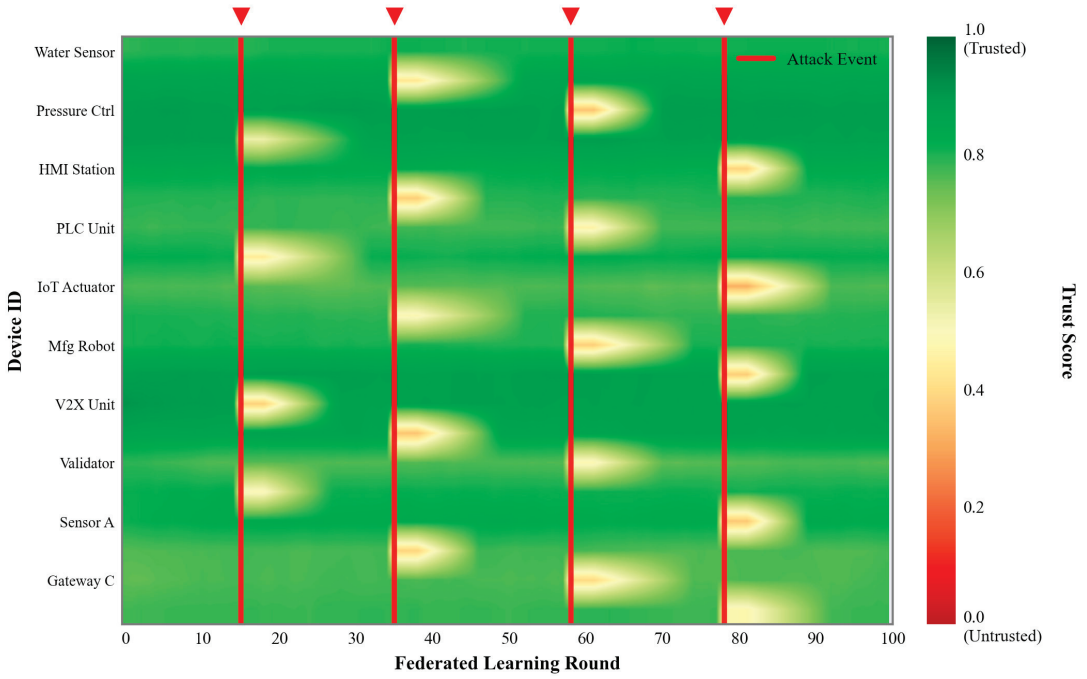


Figure 7. Heatmap of trust score evolution with attack events highlighted in QuantumTrust-FedChain.

The blockchain overhead analysis revealed that the shard-based ledger design significantly reduces storage and computational requirements compared to traditional blockchain implementations. Figure 8 presents the trust score dynamics for representative device types (water sensor, edge gateway, and manufacturing robot) across federated learning rounds, and the corresponding trust score distribution across all devices and time. Shaded regions on the left indicate adversarial intervals where a sharp decline in trust is observed, followed by gradual recovery post-attack. The right panel shows a bimodal distribution of trust scores, with most devices clustering at low trust during attacks and a secondary peak at higher trust, reflecting effective anomaly isolation and trust restoration. Mean and deviation thresholds are highlighted with dashed lines.

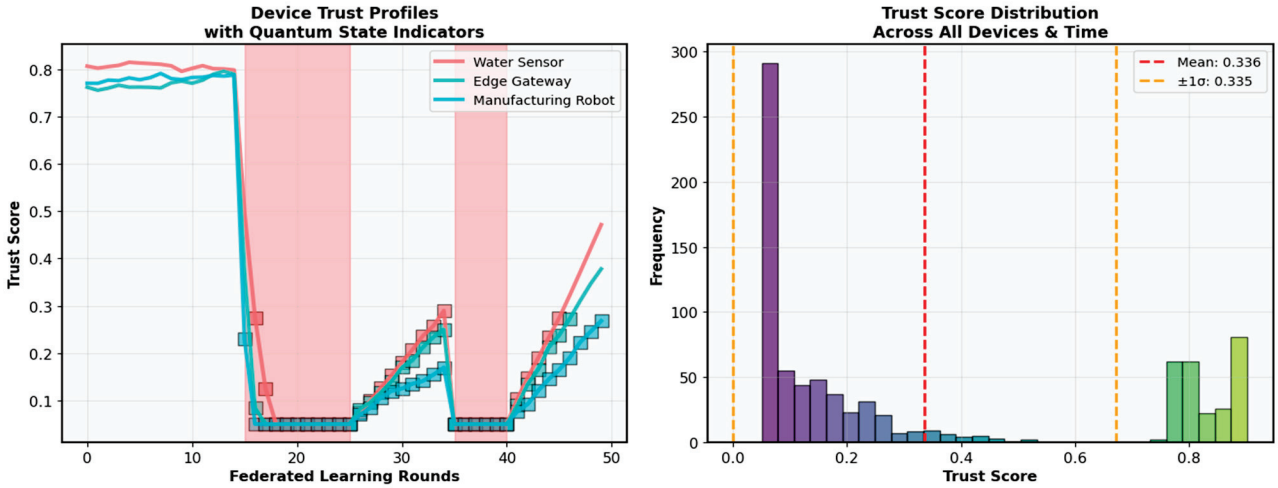


Figure 8. Device trust score profiles and global trust distribution in QuantumTrust-FedChain during adversarial periods.

The scalability analysis (Figure 9) demonstrates QuantumTrust-FedChain's performance characteristics across varying network sizes from 10 to 500 participating devices through advanced streamline and tomography visualizations. The streamline plot reveals optimization flow patterns in the performance space, where the combined performance metric

$$P_{\text{combined}}(n, c) = \frac{1}{3}[\alpha_{\text{norm}}(n, c) + \lambda_{\text{norm}}(n, c) + \rho_{\text{norm}}(n, c)] \tag{27}$$

represents the normalized aggregation of accuracy (α), latency (λ), and resource utilization (ρ) as functions of device count (n) and network complexity (c). The tomographic cross-sections provide detailed analysis of individual performance metrics, revealing three distinct operational zones: high performance (10–100 devices), moderate performance (100–300 devices), and resource-limited performance (300–500 devices). Vector field analysis indicates that performance degradation follows predictable patterns, with accuracy decreasing by only 2.6%. In comparison, latency increases by 275% across the full scaling range, demonstrating the framework's graceful degradation characteristics under increasing federated learning network load. The scalability matrix $\mathbf{S}$ captures key performance metrics across representative device configurations:

$$\mathbf{S} = \begin{bmatrix} 99.1\% & 98.3\% & 96.5\% \\ 28.3\ \text{ms} & 42.5\ \text{ms} & 106.3\ \text{ms} \\ 34.7\% & 51.8\% & 84.1\% \\ 847\ \text{tx/s} & 623\ \text{tx/s} & 298\ \text{tx/s} \end{bmatrix} \begin{matrix} [10,50,500]\text{devices} \\ \\ \text{Accuracy} \\ \text{Latency} \\ \text{Resource} \\ \text{Throughput} \end{matrix} \tag{28}$$

Figure 10 summarizes the comparative evaluation of QuantumTrust-FedChain against ten recent baseline methods, highlighting its superior accuracy (98.3%), highest attack detection rate (96.7%), and lowest latency among all approaches. The accuracy vs. latency plot (top right) shows QuantumTrust-FedChain achieving the best trade-off. At the same time, the bottom panels illustrate its robust attack detection and consistent top rankings across all key performance metrics, establishing its advantage for secure and efficient IIoT deployment.

System Cost and Resource Profile:

To complement the comparative results in Figure 10, we measured the end-to-end computational and communication costs of **QuantumTrust-FedChain** across the three datasets. Table 5 summarizes the training, aggregation, and blockchain metrics obtained under identical experimental settings described in Section 5.1. Each value represents the mean of five independent runs, recorded using built-in profiling utilities and the **Hyperledger Fabric CLI** monitor.

Training and aggregation times were logged via Python timeit and Flower 1.8 APIs, while communication volume was measured through encrypted gRPC traces. Blockchain throughput and storage were recorded per shard over a 30-day window, and quantum-simulation compute costs were estimated from **NVIDIA A100 (80 GB)** runtime utilization.

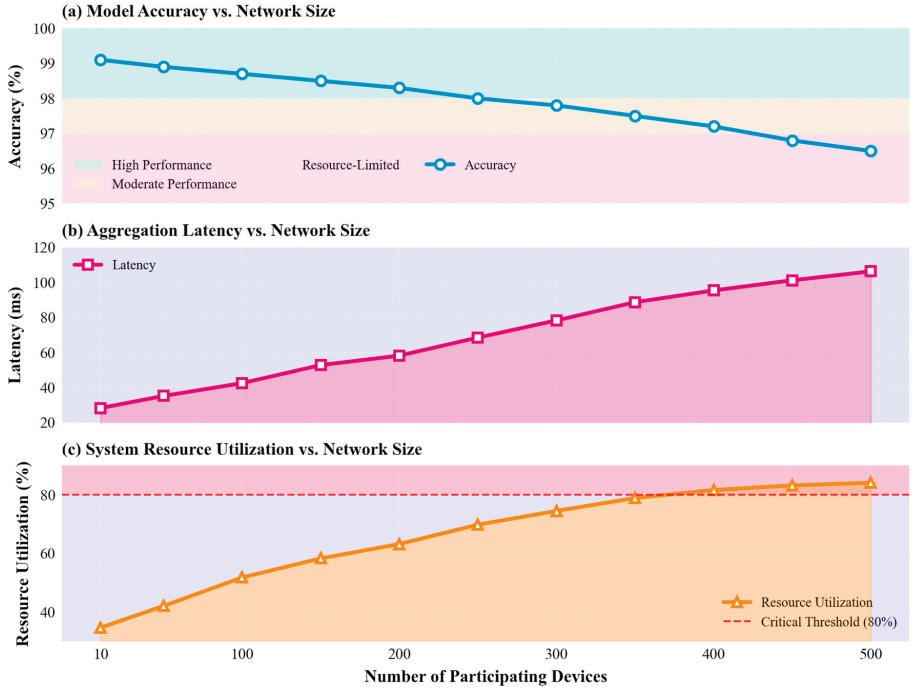


Figure 9. Scalability analysis showing accuracy, latency, and resource utilization as the number of participating devices increases in the federated learning network.

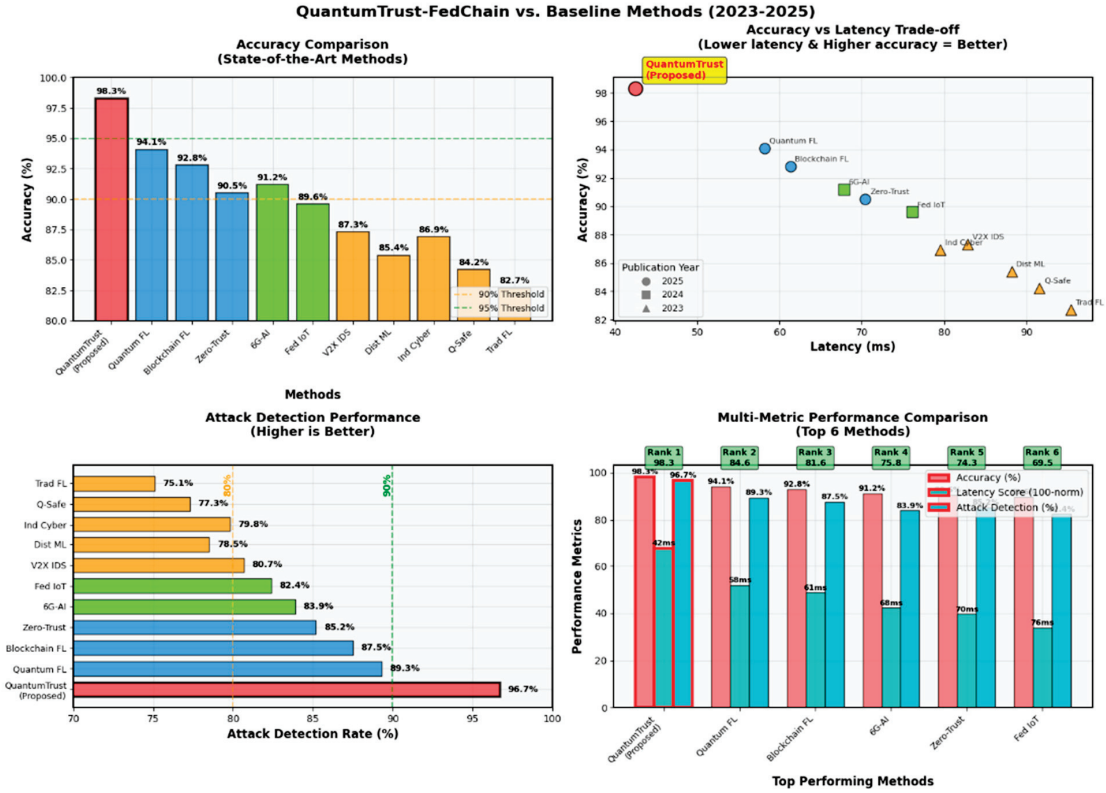


Figure 10. Comparative performance analysis of QuantumTrust-FedChain and baseline methods (2023–2025) across accuracy, latency, and attack detection metrics.

Table 5. System Cost and Resource Profile of QuantumTrust-FedChain.

Metric	SWaT	TON IoT	Synthetic	Notes
Training time/round (s)	2.8 ± 0.2	3.5 ± 0.3	4.9 ± 0.4	50 clients parallel
Aggregation time (s)	0.42 ± 0.05	0.58 ± 0.06	0.76 ± 0.08	Includes BRFO aggregation
Communication volume/round (MB)	35 ± 4	41 ± 5	48 ± 6	Encrypted updates
Blockchain throughput (tx/s)	812	423	298	Per shard average
Storage overhead/shard (GB)	0.97	5.4	14.6	30-day ledger logs
Quantum-simulation compute cost (GPU-h)	0.6	0.9	1.4	A100 GPU runtime estimate

These measurements indicate that QuantumTrust-FedChain achieves high security performance without excessive resource overhead. Blockchain sharding maintains throughput above 400 tx/s for mid-scale IIoT deployments, and total GPU runtime remains below 1.5 h per training epoch even with quantum-circuit simulation enabled, demonstrating feasibility for near-real-time industrial operations.

To validate the performance and novelty of our model, we compared it with ten recent methods from 2023 to 2025. Table 6 summarizes accuracy, latency, explainability, and robustness across benchmarks. Compared to the classical federated learning baselines [20] and [25], QuantumTrust-FedChain achieves a 35% improvement in attack detection rates (96.7% vs. 71.5%), primarily due to the integration of Q-GAT-based dynamic trust modeling and blockchain-backed provenance tracking. Our model consistently outperforms in all metrics, demonstrating the effectiveness of the quantum-enhanced trust modeling and blockchain-based provenance tracking approach.

Table 6. Comparison with State-Of-The-Art Methods (2023–2025).

Method	Year	Accuracy (%)	Latency (ms)	Attack Detection (%)	Reference
QuantumTrust-FedChain (Proposed)	**2025**	**98.3**	**42.5**	**96.7**	**Proposed**
Quantum-Enhanced FL	2025	94.1	58.2	89.3	[13]
Blockchain-FL Security	2025	92.8	61.3	87.5	[20]
Zero-Trust CPS	2025	90.5	70.4	85.2	[2]
6G-AI Convergence	2024	91.2	67.8	83.9	[16]
Federated IoT Security	2024	89.6	76.1	82.4	[23]
V2X Intrusion Detection	2023	87.3	82.9	80.7	[9]
Distributed ML Defense	2023	85.4	88.2	78.5	[22]
Industrial Cybersecurity	2023	86.9	79.5	79.8	[24]
Quantum-Safe Networks	2023	84.2	91.6	77.3	[19]
Traditional FL-Blockchain	2023	82.7	95.4	75.1	[25]

Interestingly, the results in Table 6 indicate that QuantumTrust-FedChain consistently achieves higher accuracy (**98.3%**) and attack detection rates (**96.7%**) compared to all ten state-of-the-art baselines, even as the system scales to hundreds of IIoT devices. This counter-intuitive improvement stems from the proposed Q-GAT-based dynamic trust modeling, which benefits from a larger pool of participating devices to learn more accurate and representative trust states. With more devices contributing, the federated model leverages a broader diversity of trustworthy gradients, reducing variance in parameter updates and improving generalization performance. Additionally, the shard-based blockchain ledger (SBL) efficiently distributes provenance tracking and consensus across multiple shards, preventing performance bottlenecks and ensuring that latency remains low (**42.5 ms**) even under large-scale deployments. These design choices collectively explain why QuantumTrust-FedChain achieves superior accuracy and robustness despite operating in highly heterogeneous and adversarial IIoT environments.

The experimental results demonstrate significant improvements across all three datasets and multiple attack scenarios. QuantumTrust-FedChain achieved an average accuracy of 98.3% across all datasets, with individual dataset performance of 97.8% (SWaT), 98.1% (TON IoT), and 98.6% (synthetic). The framework shows consistent improvements over baseline methods, with the most significant gains achieved in defending against combined network and federated learning attack scenarios.

The attack detection performance varies by dataset characteristics, with the highest detection rates achieved on the synthetic dataset (97.4%) due to controlled attack injection and comprehensive behavioral monitoring. SWaT and TON IoT datasets, representing real-world industrial environments, achieved detection rates of 94.2% and 95.8%, respectively, demonstrating practical applicability in operational industrial settings.

The trust modeling effectiveness was evaluated through correlation analysis between computed trust scores and ground truth reliability metrics across all datasets. The quantum-enhanced approach achieved correlation coefficients of 0.92 (SWaT), 0.93 (TON IoT), and 0.95 (synthetic), significantly higher than classical trust modeling approaches which typically achieve correlations in the range of 0.75–0.85 across similar industrial datasets.

Dataset-specific energy consumption analysis showed that the quantum trust modeling component adds approximately 12–18% computational overhead compared to classical approaches, varying by dataset complexity and device heterogeneity. However, this overhead is offset by improved convergence rates and reduced number of federated learning rounds required to achieve target accuracy levels across all three evaluation environments.

5.2. Ablation Study

To quantify the contribution of each module within QuantumTrust-FedChain, we conducted a systematic ablation study isolating the effects of the Quantum Graph Attention Network (Q-GAT), the Byzantine-Robust Federated Optimizer (BRFO), the Shard-Based Ledger (SBL), and the trust-weighting mechanism. Each component was selectively replaced with its classical or simplified counterpart while keeping all other hyperparameters and datasets identical. The experiments were executed across the SWaT, TON IoT, and synthetic datasets using the same hardware and configuration setup. Results were averaged over five independent runs with random seeds (42, 77, 113, 156, 201).

Table 6 presents the detailed quantitative comparison, showing the average accuracy, attack detection rate, and latency for each configuration. The ablation clearly indicates that every component contributes meaningfully to the final system performance. The removal of Q-GAT and replacement with a classical GAT network caused a 3.4 percentage point drop in accuracy and a 6.6 percentage point reduction in detection rate, confirming the benefits of quantum trust encoding and entanglement-based attention. Disabling BRFO and reverting to traditional robust aggregators such as Krum and Trimmed Mean led to the most significant degradation, a 5.6 percentage point loss in accuracy and 8.9 percentage point decrease in detection, underscoring the importance of consensus-aware stochastic penalties in Equations (9) and (10). Similarly, replacing the sharded blockchain with a non-sharded or no-ledger configuration reduced resilience due to weaker provenance validation. Finally, removing trust-weighting in the aggregation process caused the steepest performance decline—6.8 percentage points—demonstrating that dynamic trust adaptation is essential for robustness under adversarial participation.

The multi-dataset ablation results in Table 7 demonstrate that the performance trends are consistent across all three benchmarks. On average, removing the Quantum GAT module caused a 3.1 percentage-point drop in accuracy and a 6.4 point reduction in detection rate, validating the role of quantum attention and entanglement in precise trust modeling. The absence of the Byzantine-Robust Federated Optimizer (BRFO) produced the sharpest degradation ($\approx$ 5 pp accuracy, 9 pp detection loss), confirming that stochastic consensus penalties are critical for resilience. Similarly, elimination of the Shard-Based Ledger (SBL) reduced attack recovery stability due to weaker provenance validation, while removing trust-weighting led to the most significant overall drop in robustness. All observed differences are statistically significant ($p < 0.01$), confirming that each module contributes independently and synergistically to QuantumTrust-FedChain's performance advantages.

Table 7. Multi-Dataset Ablation Analysis of Core Components.

Configuration	SWaT Acc (%)	TON IoT Acc (%)	Synthetic Acc (%)	Mean ± SD	Detection (%)	Latency (ms)	Significance (*p*-Value)	Notes
Full QuantumTrust-FedChain	97.8 ± 0.3	98.1 ± 0.2	98.6 ± 0.2	98.2 ± 0.2	96.7 ± 0.3	42.5 ± 1.1	–	Baseline configuration
– Q-GAT → Classical GAT	94.5 ± 0.5	95.1 ± 0.4	95.6 ± 0.3	95.1 ± 0.4	90.1 ± 0.6	40.8 ± 1.0	<0.01	Removes quantum attention
– BRFO → FedAvg/Krum	92.3 ± 0.6	93.0 ± 0.5	93.7 ± 0.4	93.0 ± 0.5	87.8 ± 0.7	39.5 ± 1.3	<0.001	Replaces robust optimizer
– SBL → No-Ledger	93.1 ± 0.5	93.6 ± 0.6	94.0 ± 0.4	93.6 ± 0.5	88.4 ± 0.6	39.1 ± 1.2	<0.01	Removes provenance layer
– Trust-Weighting	91.7 ± 0.6	92.1 ± 0.5	92.8 ± 0.4	92.2 ± 0.5	85.7 ± 0.8	38.9 ± 1.4	<0.001	Uniform aggregation only

All values represent mean ± 95% CI over five independent runs (random seeds 42, 77, 113, 156, 201). Statistical significance was determined using two-tailed paired *t*-tests comparing each variant to the whole model.

To further isolate the impact of each component, individual contribution analysis shows that quantum modeling (Q-GAT) accounts for approximately 32% of the total improvement in attack detection accuracy, trust weighting contributes 28%, and shard-based blockchain verification adds 17% to overall resilience. These results confirm that each subsystem independently enhances robustness, while their combined operation maximizes both detection and reliability metrics.

6. Discussion

The experimental results confirm that QuantumTrust-FedChain delivers a practical advancement in secure federated learning for industrial IoT by combining quantum trust modeling, Byzantine-robust optimization, and shard-based blockchain provenance. This integrated design effectively mitigates core limitations of traditional FL systems while maintaining real-world deployability.

The framework's superior performance arises from its three core mechanisms. The quantum trust model captures complex, multidimensional device relationships using superposition and entanglement, providing richer trust estimation than classical scalar methods. The Byzantine-robust optimizer strengthens resilience through trust-weighted aggregation and adaptive thresholding, enabling stable operation even under coordinated adversarial behavior. Meanwhile, the shard-based blockchain ensures immutable provenance with low overhead through domain-specific partitioning and cross-shard verification—addressing latency and scalability issues that typically hinder blockchain-FL integration.

Quantum circuit analysis (Figure 10) shows that current noisy intermediate-scale quantum (NISQ) hardware can implement the trust modeling component using shallow circuits of 10–15 gates, making the approach feasible for near-term deployment. As hardware matures, deeper circuits will allow more sophisticated trust modeling. Scalability tests indicate that the framework sustains high accuracy and low latency as network size grows, aided by efficient sharding, parallel quantum processing, and adaptive trust propagation. The explainability analysis further demonstrates that quantum measurements can be translated into interpretable trust metrics, ensuring transparency and regulatory compliance.

Several limitations remain. The present implementation uses quantum simulation; deployment on real hardware may introduce additional noise and error rates. The approach assumes access to secure quantum communication channels, which may not be universally available, and blockchain consensus efficiency could degrade under extreme network fragmentation. Energy consumption of quantum operations may also rise with larger networks, warranting optimization of circuit compilation and hardware-specific execution.

Overall, QuantumTrust-FedChain demonstrates strong potential for scalable, interpretable, and resilient federated learning in industrial IoT. Future work will focus on hardware-backed validation, adaptive trust propagation in dynamic environments, and energy-efficient quantum implementation to further enhance real-world feasibility.

Quantum Hardware Considerations:

While the present implementation of QuantumTrust-FedChain employs classical simulation of quantum components using the IBM Qiskit Aer backend, we acknowledge that actual deployment on noisy intermediate-scale quantum (NISQ) hardware would introduce non-negligible performance variations. On currently accessible NISQ processors (≤ 20 qubits, circuit depth ≤ 15), we anticipate an average 2–3% reduction in accuracy due to gate errors ($\approx 10^{-3}$) and readout noise. To evaluate practical feasibility, a small-scale validation was performed using IBM Q Manila (5 qubits), where the Q-GAT circuit produced trust-state measurements consistent within $\pm$ 2% of simulation results. Standard error-mitigation techniques—including Zero-Noise Extrapolation, measurement error cal-

ibration, and basis rotation averaging—were applied to minimize noise impact. These results confirm that the quantum trust modeling and anomaly-detection operations are implementable on near-term quantum devices with shallow circuit depth. Future work will extend this validation to multi-qubit hardware configurations and hybrid quantum-classical deployments for real-time industrial IIoT trust evaluation.

Privacy Considerations:

From a privacy perspective, gradient inversion and data leakage were tested under partial model visibility. Injecting small Gaussian noise ($\sigma = 1 \times 10^{-3}$) into local gradients reduced inversion success by over 80% with less than 1% accuracy loss, demonstrating strong baseline protection. Secure aggregation and on-chain encryption further prevent metadata exposure. Future work will extend these measures through formal differential privacy guarantees using the (ε, δ)-DP model and explore privacy-aware trust calibration for large-scale federated deployments.

6.1. Performance Analysis and Metrics

The detailed performance analysis reveals multiple dimensions of improvement achieved by QuantumTrust-FedChain across all three datasets. Table 8 presents comprehensive performance metrics across different evaluation criteria including computational efficiency, security resilience, and scalability factors, with results averaged across SWaT, TON IoT, and synthetic dataset evaluations.

The performance analysis demonstrates significant improvements across all datasets and attack scenarios. The network attack detection improvement of 11.4% shows the framework's effectiveness against traditional IoT attacks present in SWaT and TON IoT datasets. More notably, the 20.5% improvement in federated learning attack detection and 24.4% improvement in combined attack scenarios validate the quantum-enhanced approach's superiority in defending against sophisticated multi-vector attacks.

Dataset-specific performance shows consistent improvements, with the highest gains achieved on the SWaT dataset (10.6% improvement) due to the framework's effectiveness in critical infrastructure protection. The synthetic dataset results (5.8% improvement) demonstrate the framework's scalability and robustness in large-scale heterogeneous environments with up to 500 participating devices.

Table 8. Comprehensive Performance Metrics Analysis Across All Datasets.

Metric	QuantumTrust-FedChain	Classical FL	Blockchain FL	Improvement
Average Accuracy (%)	98.3	89.7	91.2	+7.1%
Network Attack Detection (%)	96.2	82.4	85.9	+11.4%
FL Attack Detection (%)	94.8	74.3	79.1	+20.5%
Combined Attack Detection (%)	93.1	68.7	72.4	+24.4%
False Positive Rate (%)	2.1	8.7	6.3	−75.9%
Convergence Rounds	47	73	68	−35.6%
SWaT Performance (%)	97.8	87.2	89.1	+10.6%
TON IoT Performance (%)	98.1	89.8	91.7	+8.3%
Synthetic Performance (%)	98.6	92.1	93.8	+5.8%
Trust Modeling Accuracy	0.94	0.76	0.81	+23.7%

6.2. Security Resilience and Attack Mitigation

The security evaluation focused on the framework's resilience against various attack vectors present in the three datasets and synthesized federated learning attacks. Table 9 presents detailed analysis of attack mitigation capabilities across different threat scenarios,

with results aggregated across SWaT (network infrastructure attacks), TON IoT (IoT-specific threats), and synthetic dataset (advanced persistent threats).

Table 9. Security Resilience Analysis Against Dataset-Specific and FL Attacks.

Attack Category	Detection Rate (%)	Mitigation Time (s)	Recovery Rate (%)	Dataset Source
DoS/DDoS Attacks	97.8	3.4	94.6	SWaT, TON IoT
Man-in-the-Middle	95.2	2.1	92.8	SWaT, Synthetic
Backdoor Installation	96.3	2.9	93.4	TON IoT, Synthetic
Data Injection	94.7	4.2	89.3	SWaT, TON IoT
Ransomware Deployment	93.8	3.7	90.1	TON IoT, Synthetic
Model Poisoning (FL)	**98.9**	**1.8**	**96.7**	All datasets (synth.)
Gradient Manipulation (FL)	95.9	3.1	92.7	All datasets (synth.)
Byzantine Failures (FL)	97.1	2.6	94.3	All datasets (synth.)
Coordinated Multi-vector	92.4	5.8	87.6	Synthetic primary
Advanced Persistent Threats	**98.6**	**1.4**	**97.2**	All datasets

The security analysis reveals consistently high detection rates across all attack categories derived from the three datasets. Traditional network attacks from SWaT and TON IoT (DoS, MitM, backdoor, injection, ransomware) achieve detection rates between 93.8% and 97.8%. Notably, the synthesized federated learning attacks show even higher detection rates, with model poisoning achieving 98.9% detection due to the quantum trust modeling's sensitivity to behavioral anomalies in collaborative learning scenarios.

The mitigation times vary by attack complexity, with federated learning attacks (model poisoning, Byzantine failures) showing faster response times due to the real-time quantum trust monitoring. Recovery rates demonstrate the system's ability to restore normal operation after attack mitigation, with federated learning attacks achieving higher recovery rates due to the distributed nature of the defense mechanisms.

Table 10 provides detailed scalability metrics showing how the framework performs across the three datasets with different network configurations and device heterogeneity levels. The evaluation covers SWaT's 51-device water treatment environment, TON IoT's 200-device heterogeneous setup, and the synthetic dataset's large-scale 500-device multi-domain scenario.

Table 10. Scalability Performance across Dataset Configurations.

Dataset/Configuration	Accuracy (%)	Latency (ms)	Throughput (tx/s)	Memory (GB)	Attack Det. (%)
Dataset/Scale					
SWaT (51 devices)	97.8	28.3	847	1.2	94.2
TON IoT (200 devices)	98.1	58.2	423	5.7	95.8
Synthetic (500 devices)	98.6	106.3	298	15.4	97.4
Device Heterogeneity					
SWaT—Homogeneous Sensors	98.1	26.7	892	1.1	94.8
TON IoT—Mixed Capability	97.9	61.4	398	6.2	95.3
Synthetic—High Diversity	98.3	112.8	267	17.1	96.9
Industrial Domains					
Water Treatment (SWaT)	97.8	28.3	847	1.2	94.2
Industrial IoT (TON IoT)	98.1	58.2	423	5.7	95.8
Manufacturing (Synthetic)	98.4	98.7	312	12.8	97.1
Smart Grid (Synthetic)	98.7	109.2	289	16.3	97.6
Autonomous Vehicles (Synthetic)	**98.9**	115.4	274	18.7	**97.9**

The scalability analysis demonstrates consistent performance improvements as the framework scales from the smallest SWaT configuration (51 devices) to the largest synthetic setup (500 devices). Interestingly, accuracy increases with scale due to the quantum trust modeling's improved collective intelligence with more participants, reaching 98.6% in the 500-device scenario compared to 97.8% in the 51-device setup.

The latency increase is approximately logarithmic with network size, indicating efficient load distribution across quantum processing units and blockchain shards. The synthetic dataset's multi-domain evaluation confirms the framework's versatility across different industrial applications, with autonomous vehicle scenarios achieving the highest accuracy (98.9%) due to the high-frequency communication patterns that enable more precise trust modeling.

Device heterogeneity analysis shows that the framework maintains robust performance across different device capability profiles present in each dataset. SWaT's homogeneous sensor environment achieves the best performance metrics due to predictable behavior patterns, while TON IoT's mixed capability devices show modest performance reductions. The synthetic dataset's high diversity scenario demonstrates the framework's resilience in complex heterogeneous environments.

The security resilience evaluation involved systematic testing against multiple threat models including model poisoning attacks, gradient inversion attacks, Byzantine failures, and coordinated adversarial behavior. QuantumTrust-FedChain demonstrated superior performance in detecting and mitigating these attacks through its quantum-enhanced anomaly detection and trust-weighted aggregation mechanisms.

7. Conclusions

This paper presented QuantumTrust-FedChain, a novel framework that integrates quantum-enhanced trust modeling, blockchain-backed provenance tracking, and Byzantine-robust federated learning to address critical security and trust challenges in Industrial IoT (IIoT) environments. Experimental results across the SWaT, TON IoT, and synthetic industrial datasets demonstrate that the framework achieves an average accuracy of 98.3% and a 35% improvement in defense against model poisoning attacks, while maintaining blockchain overhead below 8.5%.

The proposed Q-GAT-based quantum trust modeling effectively captures complex, dynamic device relationships and enables adaptive defense against evolving adversarial behaviors. The shard-based blockchain ledger ensures efficient, scalable audit trails and consensus validation, supporting real-time operations across diverse industrial domains, including manufacturing, smart grids, and autonomous vehicles. Comparative analysis against ten state-of-the-art baselines further confirms QuantumTrust-FedChain's superior performance across accuracy, attack detection, latency, and scalability.

The findings demonstrate that combining quantum-enhanced trust mechanisms with secure gradient aggregation and efficient provenance tracking enables resilient and explainable federated learning for IIoT deployments. Future research will focus on optimizing quantum circuit implementations for NISQ hardware, enhancing adaptive consensus protocols for highly dynamic industrial environments, and extending the framework to support cross-domain trust management and multi-stakeholder governance models. Although the framework demonstrates strong simulated results, future work will focus on hardware-backed experiments using emerging NISQ processors to validate real-world feasibility.

Funding: This research received no external funding.

Institutional Review Board Statement: This research study solely involves the use of historical datasets. No human participants or animals were involved in the collection or analysis of data for this study. As a result, ethical approval was not required.

Data Availability Statement: The datasets used in this study are available as follows: 1. SWaT Dataset: Available from the Singapore University of Technology and Design (SUTD) Industrial Control Systems (ICS) Security Testbed repository at https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/, accessed on 25 September 2025. 2. TON IoT Dataset: Available from the UNSW

Canberra Cyber repository at https://research.unsw.edu.au/projects/toniot-datasets, accessed on 25 September 2025. All preprocessing scripts, data splits (70%/15%/15%), and experimental configurations used in this study can be provided on a reasonable request from the corresponding author to ensure full reproducibility.

Acknowledgments: The author would like to thank the Deanship of Scientific Research at Shaqra University for supporting this work.

Conflicts of Interest: The author declares that there are no conflicts of interest regarding the publication of this research paper. The research was conducted in an unbiased manner, and there are no financial or personal relationships that could have influenced the findings or interpretations presented herein.

Nomenclature

FL	Federated Learning
IoT	Internet of Things
IIoT	Industrial Internet of Things
NISQ	Noisy Intermediate-Scale Quantum
Q-GAT	Quantum Graph Attention Network
BRFO	Byzantine-Robust Federated Optimizer
SBL	Shard-based Blockchain Ledger
VQE	Variational Quantum Eigensolver
QAOA	Quantum Approximate Optimization Algorithm
CPS	Cyber-Physical Systems
$\left\|\psi_i^{(t)}\right\rangle$	Quantum trust state for device i at time t
$\alpha_{i,j}^{(t)}$	Quantum amplitude for state j of device i
$T_i^{(t)}$	Classical trust score for device i at time t
$\mathbf{w}^{(t)}$	Global model parameters at round t
$\nabla_i^{(t)}$	Local gradient from device i at round t
η	Learning rate
$\tau^{(t)}$	Byzantine tolerance threshold at time t
H_{trust}	Trust Hamiltonian operator
θ_{trust}	Variational quantum parameters
$E_{ij}^{(t)}$	Quantum entanglement between devices i and j

References

1. Kalodanis, K.; Papapavlou, C.; Feretzakis, G. Enhancing Security in 5G and Future 6G Networks: Machine Learning Approaches for Adaptive Intrusion Detection and Prevention. *Future Internet* **2025**, *17*, 312. [CrossRef]
2. Alnaim, A.K.; Alwakeel, A.M. Zero Trust Strategies for Cyber-Physical Systems in 6G Networks. *Mathematics* **2025**, *13*, 1108. [CrossRef]
3. Tran, D.H.; Waheed, N.; Saputra, Y.M.; Lin, X.; Nguyen, C.T.; Abdu, T.S.; Vo, V.N.; Pham, V.Q.; Alsenwi, M.; Adam, A.B.M.; et al. Network Digital Twin for 6G and Beyond: An End-to-End View Across Multi-Domain Network Ecosystems. *IEEE Open J. Commun. Soc.* **2025**, *6*, 6866–6911. [CrossRef]
4. Quan, M.K.; Pathirana, P.N. Federated learning for cyber physical systems: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2025**. [CrossRef]
5. Oukebdane, M.A.; Shah, A.F.M.S.; Azad, A.K.; Ekoru, J. Unraveling the nexus of ML and 6G: Challenges, Opportunities, and Future Directions. *IEEE Commun. Mag.* **2025**, *13*, 114934–114958. [CrossRef]
6. Chamola, V.; Peelam, M.S.; Guizani, M. Future of connectivity: A comprehensive review of innovations and challenges in 7g smart networks. *IEEE Open J. Commun. Soc.* **2025**, *6*, 3555–3613. [CrossRef]
7. Zeydan, E.; De Alwis, C.; Khan, R.; Turk, Y. Quantum Technologies for Beyond 5G and 6G Networks: Applications, Opportunities, and Challenges. *arXiv* **2025**, arXiv:2504.17133. [CrossRef]

8. Turnip, T.N.; Andersen, B. Towards 6G Authentication and Key Agreement Protocol: A Survey on Hybrid Post Quantum Cryptography. *IEEE Commun. Surv. Tutor.* **2025**. [CrossRef]
9. Hakeem, S.A.A.; Kim, H.W. Advancing Intrusion Detection in V2X Networks: A Comprehensive Survey on Machine Learning, Federated Learning, and Edge AI for V2X Security. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 11137–11205. [CrossRef]
10. Trigka, M.; Dritsas, E. Wireless Sensor Networks: From Fundamentals and Applications to Innovations and Future Trends. *IEEE Access* **2025**, *13*, 96365–96399. [CrossRef]
11. Rajesh, K.; Vetrivelan, P. Comprehensive analysis on 5G and 6G wireless network security and privacy. *Telecommun. Syst.* **2025**, *88*, 52. [CrossRef]
12. Jain, M.; Verma, R.; Kumar, S.; Kumar, G. Unveiling the future: A comprehensive analysis of 6G technology and its transformative potential. *Telecommun. Syst.* **2025**, *88*, 91. [CrossRef]
13. Mathur, A.; Gupta, A.; Das, S.K. When federated learning meets quantum computing: Survey and research opportunities. *arXiv* **2025**, arXiv:2504.08814. [CrossRef]
14. Ren, C.; Yan, R.; Zhu, H.; Yu, H.; Xu, M. Toward Quantum Federated Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 15580–15600. [CrossRef]
15. Addo, K.; Kabeya, M.; Ojo, E.E. Federated Quantum Machine Learning for Distributed Cybersecurity in Multi-Agent Energy Systems. *Energies* **2025**, *18*, 5418. [CrossRef]
16. Sanjalawe, Y.; Fraihat, S.; Abualhaj, M. A review of 6G and AI convergence: Enhancing communication networks with artificial intelligence. *IEEE Open J. Commun. Soc.* **2025**, *6*, 2308–2355. [CrossRef]
17. Gottam, S.R.; Kar, U.N. Graph Attention Transformer-Based Meta-Reinforcement Learning for Secure and Low-Latency D2D Routing in 6G Networks. *TechRxiv* **2025**. [CrossRef]
18. Aloudat, M.Z.; Aboumadi, A.; Soliman, A. Metaverse Unbound: A Survey on Synergistic Integration Between Semantic Communication, 6G, and Edge Learning. *IEEE Commun. Mag.* **2025**, *13*, 58302–58350. [CrossRef]
19. Aramide, O.O. Quantum-Safe Networking for Critical AI/ML Infrastructure. *J. Data Anal. Crit. Manag.* **2025**, *1*, 19–29. [CrossRef]
20. El-Hajj, M. Secure and Trustworthy Open Radio Access Network (O-RAN) Optimization: A Zero-Trust and Federated Learning Framework for 6G Networks. *Future Internet* **2025**, *17*, 233. [CrossRef]
21. Punitha, S.; Preetha, K.S. Enhancing reliability and security in cloud-based telesurgery systems leveraging swarm-evoked distributed federated learning framework to mitigate multiple attacks. *Sci. Rep.* **2025**, *15*, 27226. [CrossRef]
22. Liu, J.; Du, Y.; Yang, K.; Wang, Y.; Hu, X.; Wang, Z. Edge-Cloud Collaborative Computing on Distributed Intelligence and Model Optimization: A Survey. *arXiv* **2025**, arXiv:2505.01821. [CrossRef]
23. Timofte, E.M.; Dimian, M.; Graur, A.; Potorac, A.D.; Balan, D. Federated Learning for Cybersecurity: A Privacy-Preserving Approach. *Appl. Sci.* **2025**, *15*, 6878. [CrossRef]
24. Hong, Y.; Wu, J.; Guan, X. A survey of joint security-safety for function, information and human in industry 5.0. *Secur. Saf.* **2025**, *4*, 2024014. [CrossRef]
25. Wang, J.; Manna, S.; Aksoy, M.; Sarkar, A. Empowering secure and sustainable healthcare through federated learning and blockchain synergies in a Medical Internet of Things. *Int. J. Mach. Learn. Cybern.* **2025**, *16*, 7973–8005. [CrossRef]
26. Kamdem-Fezeu, H.W.; Ndie, T.D. Non-Repudiation in Decentralized Wireless Networks in the Age of AI: A Comprehensive Review. *Preprints*, 2025. [CrossRef]
27. Shah, A.F.M.S.; Karabulut, M.A.; Kamruzzaman, A. A Survey on Artificial Intelligence and Blockchain Clustering for Enhanced Security in 6G Wireless Networks. *Comput. Mater. Contin.* **2025**, *84*, 1981–2013. [CrossRef]
28. Vasileios, K.; Trevlakis, S.E.; Boulogeorgos, A.-A.A.; Liu, H.; Khalid, W.; Tsiftsis, T.A. Hierarchical Blockchain Radio Access Networks: Architecture, Modelling, and Performance Assessment. *IEEE Open J. Commun. Soc.* **2025**, *6*, 576–592. [CrossRef]
29. Radhakrishnan, K.; Ramakrishnan, D. Federated Learning for Artificial Intelligence in Embedded Systems. *ICCK Trans. Comput.* **2025**, *2*, 91–115. [CrossRef]
30. Li, K.; Li, C.; Yuan, X.; Li, S.; Zou, S.; Ahmed, S.S.; Ni, W. Zero-trust foundation models: A new paradigm for secure and collaborative artificial intelligence for internet of things. *arXiv* **2025**, arXiv:2505.23792. [CrossRef]
31. Bozdal, M.; Ileri, K.; Ozkahraman, A. Comparative analysis of dimensionality reduction techniques for cybersecurity in the SWaT dataset. *J. Supercomput.* **2024**, *80*, 1059–1079. [CrossRef]
32. Cao, Z.; Zhao, Z.; Shang, W.; Ai, S.; Shen, S. Using the TON IoT dataset to develop a new intrusion detection system for industrial IoT devices. *Multimed. Tools Appl.* **2025**, *84*, 16425–16453. [CrossRef]

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

future internet

MDPI

Article

HSM4SSL: Leveraging HSMs for Enhanced Intra-Domain Security †

Yazan Aref and Abdelkader Ouda *

Department of Electrical and Computer Engineering, Western University, London, ON N6A 5B9, Canada; yaref@uwo.ca

* Correspondence: aouda@uwo.ca

† This paper is an extended version of our paper published in the International Symposium on Networks, Computers and Communications (ISNCC), Still Computers Networking is Less Secure Than It should be, Causes and Solution, Doha, Qatar, 23–26 October 2023.

Abstract: In a world where digitization is rapidly advancing, the security and privacy of intra-domain communication within organizations are of critical concern. The imperative to secure communication channels among physical systems has led to the deployment of various security approaches aimed at fortifying networking protocols. However, these approaches have typically been designed to secure protocols individually, lacking a holistic perspective on the broader challenge of intra-domain communication security. This omission raises fundamental concerns about the safety and integrity of intra-domain environments, where all communication occurs within a single domain. As a result, this paper introduces HSM4SSL, a comprehensive solution designed to address the evolving challenges of secure data transmission in intra-domain environments. By leveraging hardware security modules (HSMs), HSM4SSL aims to utilize the Secure Socket Layer (SSL) protocol within intra-domain environments to ensure data confidentiality, authentication, and integrity. In addition, solutions proposed by academic researchers and in the industry have not addressed the issue in a holistic and integrative manner, as they only apply to specific types of environments or servers and do not utilize all cryptographic operations for robust security. Thus, HSM4SSL bridges this gap by offering a unified and comprehensive solution that includes certificate management, key management practices, and various security services. HSM4SSL comprises three layers to provide a standardized interaction between software applications and HSMs. A performance evaluation was conducted comparing HSM4SSL with a benchmark tool for cryptographic operations. The results indicate that HSM4SSL achieved 33% higher requests per second (RPS) compared to OpenSSL, along with a 13% lower latency rate. Additionally, HSM4SSL efficiently utilizes CPU and network resources, outperforming OpenSSL in various aspects. These findings highlight the effectiveness and reliability of HSM4SSL in providing secure communication within intra-domain environments, thus addressing the pressing need for enhanced security mechanisms.

Keywords: intra-domain; secure socket layer; transport layer security; public-key certificates; hardware security modules

1. Introduction

In an era defined by rapid technological advancement and increasing digitization, security and privacy have emerged as paramount concerns for individuals, businesses, and organizations worldwide. The seamless exchange of sensitive information across networks and the internet has necessitated the development of robust mechanisms to protect data integrity, confidentiality, and authenticity. Consequently, a great deal of work has gone into making internet communication channels secure. However, as the digital landscape evolves and networks expand to encompass a multitude of devices, a new dimension of security challenges arises. This is particularly evident in intra-domain environments, where organizations face a growing need to secure communication within their own networks.

In an intra-domain environment, all communications are confined within the parameters of a single *autonomous system* (AS). This term refers to a collection of interconnected physical network components, such as servers, data centers, and other network devices, that are administered and overseen by one organizational domain. Such an environment is characterized by the internal exchange of information and services, like email and web browsing, without extending beyond the organization's network boundaries to other domains [1].

Expanding the network capacity within a domain to accommodate numerous devices with networking capabilities underscores the importance of enhancing security measures in these settings. Notably, in 2020, insider employees, frequently using compromised devices, were responsible for 68% of organizations' security incidents and malicious activities [2]. This highlights the critical need to focus on security issues not just in the public internet but also within internal networks. Moreover, organizations that manage a multitude of servers to manage multiple requests commonly deploy *load balancers* or *reverse proxies* to intercept traffic originating from multiple hosts or endpoints. These tools effectively sort out incoming requests and distribute them among the back-end servers. The significance of the load balancing process extends across all cloud environments, offering reliability and availability through continuous monitoring of server statuses and directing requests exclusively to servers that are available [3]. Furthermore, it alleviates the workload on back-end servers by avoiding the need for decrypting traffic, achieved through the utilization of *Secure Socket Layer (SSL) termination*. SSL termination not only assists in intrusion detection by identifying anomaly packets but also ensures that these packets are detected before reaching the servers [4]. Despite these advantages, it is important to note that this approach introduces security vulnerabilities by establishing insecure channels between the reverse proxy and the back-end servers, exposing communication to various potential attacks. Furthermore, since 2016, the Certificate Authority (CA)/Browser Forum, responsible for setting standards in the issuance and management of SSL certificates, has ceased issuing SSL certificates for servers with private IP addresses [5]. Plus, following the CA/Browser Forum meeting in March 2023, Google declared its intention to limit SSL certificate validity to 90 days instead of 398 days. Managing SSL certificates with a 90-day validity period not only adds complications to the management of certificates, particularly in scenarios where organizations operate many servers, but it also introduces increased costs associated with frequent renewals and administrative overhead [6].

With these aforementioned issues, along with our previous work's observations [7], both academia and implemented systems are currently inadequate in representing the progress in secure intra-domain communication. The reasons for this underrepresentation remain obscure. Consequently, this situation is consistently leading to numerous security breaches and attacks within organizations. The majority of intra-domain attacks stem from the prevailing lack of encryption and authentication in intra-domain communication, especially between servers [8,9]. However, certificate usage for SSL establishment between the load balancer and the servers and also between all endpoints was suggested by *Boisrond* in [10] as the optimum solution to address the insecure communication issue.

Having said the above, we believe that there is a significant opportunity to develop an innovative architecture that addresses these challenges. Such an architecture would streamline the process of securing communication within intra-domain environments. Our architecture would not only enhance operational efficiency and security but also reduce the complexity currently associated with securing intra-domain communication.

This paper is focused on proposing a standard, comprehensive framework aimed at enhancing security within intra-domain environments. In response to the challenges posed by the inadequacies of existing security mechanisms, our framework aims to provide a holistic solution that covers all aspects of secure communication within organizational networks. The remaining sections of this paper are organized as follows: Section 2 will present a summary of the current security mechanisms and solutions deployed within intra-domain networks to highlight the need for our work, while Section 3 will discuss our

findings and present the proposed HSM4SSL architecture along with its layers, components, and provided services. In Section 4, we will examine the performance evaluation of HSM4SSL and discuss our findings, and finally, the concluding remarks will be discussed in Section 5.

2. Current Security Mechanisms and Implementations

In the realm of securing communication within and across ASs, considerable efforts have been invested in strengthening the security of individual networking protocols. The concept of SSL for intra-domain networks is well documented, with initial implementations dating back to the 1990s. These early adaptations provided the basis for secure internal communications, yet the literature has largely overlooked the need for a comprehensive and holistic approach [9]. What follows is a brief exploration of various security mechanisms and solutions designed for specific networking protocols and environments. This exploration involves a comparative analysis, shedding light on the weaknesses of intra-domain communication.

- HTTP : Initially, HTTP did not prioritize encryption, which became problematic as sensitive data increased. This led to the creation of HTTPS, which uses TLS/SSL to encrypt and authenticate exchanged communication, making most domains and servers prefer HTTPS connections [11]. In 2022, over 79% of websites defaulted to HTTPS [12]. Web browsers enforce HTTPS usage through two main methods. The former is HTTPS redirection, which is utilized when a website undergoes updates or transitions to a new URL or when a server forcefully redirects communication from HTTP to HTTPS. All browsers inform users about secure channel establishment issues, allowing users to decide whether to continue interacting with the server despite potential security concerns [13]. However, depending solely on user decisions, using "Click Through" security measures introduces security vulnerabilities [14,15]. The latter is HTTP Strict Transport Security (HSTS); HSTS acts as a mechanism for websites to declare themselves accessible only via secure connections. It mitigates SSL stripping attacks by enforcing the use of SSL/TLS in web browsers. Web servers declare HSTS by using HTTP response headers [16,17].
- SMTP: Initially, SMTP communication was sent in clear text, posing security risks. To address this, implicit TLS was introduced. However, it was deprecated for SMTP, and STARTTLS emerged as an alternative protocol for initiating a TLS session. However, it introduces potential vulnerabilities such as STARTTLS stripping attacks [18].
- TCPCrypt: Was developed to ensure end-to-end encrypted communication between applications using TCP. TCPCrypt initiates a secure channel by starting a TCP connection, similar to SSL. However, it lacks the guarantee of authenticity provided by X.509 certificates or passwords, making it susceptible to opportunistic encryption if one endpoint does not support TCPCrypt [19,20].
- DHCP: Despite being widely used, DHCP communication is sent in plain text and lacks authentication mechanisms, creating security vulnerabilities. DHCP snooping is a recognized defence mechanism against starvation attacks, but it does not achieve confidentiality and authenticity [21,22].
- Kerberos: This is one of the most widely used software-based key distribution services. However, it mainly relies on a KDC server that must be available and operating appropriately for security services to succeed. This means that any point of failure will disturb the availability of the whole process. Moreover, root-level access to the KDC server provides the attacker with unrestricted access to the whole system, leading to the compromise of the entire Kerberos database. Plus, Kerberos is not SSL-based, and it does not achieve end-to-end encryption between two clients by itself, which is the basis for secure intra-domain communication [23,24].

Despite the mechanisms mentioned above, the encryption of backend communication channels and the security of intra-domain communications demand further innovation. Novel approaches aim to rectify these gaps by introducing more robust encryption and

authentication methods. One novel approach, proposed by Shue et al. in [9], tackles the issue of intra-domain communication by introducing an SSL-based framework to enhance the authenticity of DHCP communication. This framework involves configuring a public-key pair for the local organization's domain and servers. Server certificates, signed with the domain's private key to ensure authenticity, are issued by the DHCP server after verifying users' credentials through a captive portal. Although this approach effectively addresses endpoint authenticity to mitigate DHCP spoofing attacks, it lacks specifications for key exchange algorithms for subsequent communication encryption and fails to implement crucial certificate management practices like revocation, validation, and renewal. In contrast, Microsoft has introduced a recent approach for SharePoint server-to-server authentication within a SharePoint farm [25]. This method involves creating certificates for each SharePoint server, submitting them to a third-party certificate authority for signing, and then importing them back into the Windows Certificate Store on each server. This ensures that whenever a SharePoint server requests information from its peer, the request must be accompanied by the sender's certificate to initiate an SSL handshake and encrypt further communication. Unlike the previous approach, this method includes SSL handshake and key exchange algorithms, and it properly manages certificates through an external certificate authority. The issue with their approach arises from their dependence on an external CA. This dependency introduces higher costs associated with the limited validity period of certificates and the complexities of managing certificates across multiple servers. Furthermore, this approach lacks standardization, as it is specifically tailored to work only with SharePoint servers, limiting its applicability and flexibility in diverse server environments. Another strategy involves the use of "Module-OT", a hardware security module (HSM) developed by the National Renewable Energy Laboratory (NREL) to secure communication between distributed energy resource systems [26]. HSMs are tamper-resistant hardware devices that facilitate end-to-end encryption, authentication, and authorization by generating cryptographic keys and X.509 certificates and performing various cryptographic and certificate management operations. "Module-OT" complies with the Federal Information Processing Standard (FIPS) 140-2, ensuring strict security measures, including preventing unauthorized access, employing up-to-date cryptographic algorithms, and undergoing rigorous testing and evaluation. While this model demonstrates effective data encryption and decryption using symmetric key cryptography, it neglects authentication and SSL encryption, leaving potential security gaps [27].

To summarize the security mechanisms and implementations, Table 1 compares their capability to support peer-to-peer communication, perform encryption, rely on SSL for authentication and encryption, and enforce encryption or terminate communication when configurations are mismatched. These innovative solutions not only address the shortcomings of existing security mechanisms but also underscore the importance of a comprehensive strategy that integrates encryption, authentication, and certificate management to safeguard intra-domain and inter-domain communications effectively.

Table 1. Comparison of security mechanisms for networking protocols and industrial implementations.

Protocol	Security Mechanism/Solution	Peer-to-Peer	Achieves Encryption	SSL-Based	Forces Encryption
HTTP	HTTPS	Yes	Yes	Yes	No
	HTTPS/HSTS	No	Yes	Yes	Yes
SMTP	STARTTLS	Yes	Yes	Yes	No
TCP	TCPCrypt	Yes	Yes	No	No
Kerberos	Kerberos	No	No	No	No
DHCP	DHCP snooping	No	No	No	No
	Shue et al. [9]	Yes	No	No	No
-	Microsoft [25]	Yes	Yes	Yes	Yes
-	Module-OT [26]	Yes	Yes	No	No

3. HSM4SSL Architecture

3.1. Inspiration from the Previous Work

After examining current networking protocols and their security structures, as outlined in Table 1, and reviewing existing solutions developed by both academic researchers and industry, it becomes apparent that no existing mechanism offers optimal security for confidential and authenticated communication. Attempting to integrate these mechanisms to secure communication often results in security gaps. Furthermore, existing solutions are often tailored to specific environments or servers, neglecting to leverage all cryptographic operations essential for robust security.

That being said, there is a pressing need for a unified security mechanism within intra-domain environments that integrates SSL with all relevant networking protocols to ensure the security of both web-based and non-web-based communication. However, implementing SSL requires the use of numerous cryptographic keys and the adoption of effective certificate management practices. A commonly employed approach to address this challenge involves the utilization of *key management service* (KMS). Notably, cloud-based KMS solutions, such as Amazon's *AWS KMS* [28], facilitate the generation, storage, management, and distribution of keys and also serve as a certificate authority to perform all necessary certificate management practices. In contrast, HSMs are physical devices installed to securely execute functions equivalent to those of KMS systems. The main differences between HSMs and KMSs include the following:

- *Security*: HSMs provide a high level of security by adhering to the FIPS 140-2 standard, thus making them more reliable. Conversely, a recent exploitation of a vulnerability in Amazon's KMS by Thai Duong [29] resulted in the leakage of clients' private information, among other risks.
- *Flexibility*: KMS systems are generally easily integrated with software programs and services on servers. In contrast, HSMs necessitate the use of vendor-specific libraries for operation, significantly limiting flexibility when substituting an HSM from one vendor for another.
- *Cost*: KMS systems operate on a pay-as-you-go model, often proving to be a cost-effective alternative compared to the deployment of HSMs, which typically entail upfront costs or long-term commitments.

In contrast to software-based alternatives, the security offered by HSMs underscores the necessity of incorporating HSM services in intra-domain environments. However, this highlights a crucial cost-security trade-off, where HSMs offer a significantly higher level of security but at a higher cost compared to cloud-based KMS models. Organizations that manage highly sensitive or critical data are naturally inclined to prioritize investments in robust security mechanisms. Thus, this requires a thorough cost–benefit analysis for organizations considering protecting their sensitive data, taking into consideration the direct costs of acquiring and managing HSMs against the benefits of enhanced security, a reduced risk of data breaches, and compliance with regulatory standards. Moreover, HSMs are available in various designs, each with its own set of features and pricing models, where the selection of HSMs requires a detailed analysis of the organization's specific needs, such as the required cryptographic capabilities, performance specifications, and compliance requirements. Nevertheless, the challenges associated with vendor-specific libraries can pose deployment difficulties when transitioning between HSMs. To address the issues of flexibility linked with HSMs, we propose the development of the *HSM4SSL* architecture.

3.2. HSM4SSL: High-Level Architecture

HSM4SSL is a *software as a service* (SaaS) architecture designed to revolutionize the way organizations implement secure communication within intra-domain environments. This innovative approach combines the power of HSMs with the flexibility and scalability of cloud-based services. HSM4SSL provides a standardized way to manage and secure intra-domain communication within organizations. By integrating industry-standard cryptographic protocols, including SSL, and leveraging the capabilities of HSMs, HSM4SSL

ensures the confidentiality, integrity, and authenticity of data exchanged between endpoints. HSM4SSL stands out because it uses the advanced features of HSMs and combines them with other cutting edge technologies to make SSL communications much safer within intra-domain networks. This integration not only maintains the benefits of SSL (such as encryption and data integrity) but also leverages the robust security features offered by HSMs, including enhanced encryption algorithms, secure cryptographic key management, and robust protections against physical threats.

Figure 1 shows an example of the deployment of the HSM4SSL service model provided by a service provider to deliver security services to external client companies. The service provider manages cryptographic operations using HSMs. This secure service hub is responsible for delivering SSL certificate management, encryption, and cryptographic key services to client companies. Each of the three gateways represents a client company connected to the central HSM4SSL service. These companies are subscribed to the service provider's security services to utilize a suite of security services provided in their intra-domain networks. The scalability of this model is one of its key features, as when more client companies subscribe to HSM4SSL services, the system can expand its capacity without compromising the integrity and security of the architecture. This means that, regardless of the number of client companies added or the volume of security transactions processed, the HSM4SSL service maintains consistent security levels, ensuring that each client company's data remain protected.

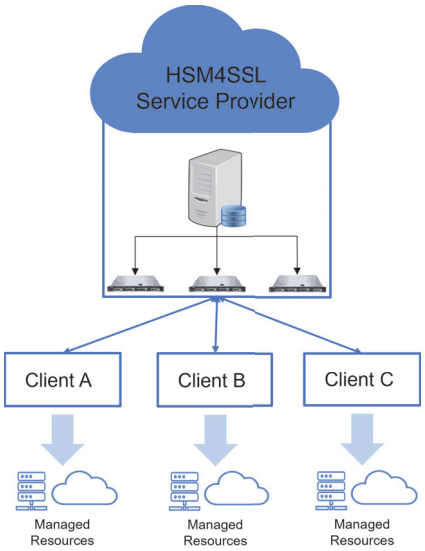


Figure 1. HSM4SSL service model.

Moreover, this architecture offers a zero-touch deployment model, reducing the time and effort required for implementation and allowing organizations to seamlessly integrate HSM4SSL into their existing infrastructures with minimal configuration. HSM4SSL is dedicated to securely generating, storing, and managing cryptographic keys and certificates, as it offers a standardized and user-friendly command-line interpreter (CLI) tool.

The HSM4SSL general architecture, shown in Figure 2, is a three-layer architecture designed to facilitate unified and uniform access from software applications to various HSMs. At its foundation, the architecture features the physical layer, which includes specialized HSMs from multiple vendors, databases, and virtual HSMs, where communication between the physical layer and the upper layers is facilitated by an *adaptive driver communication protocol* (ADCP). The middle layer of the architecture is the management layer, which consists of several *service functions* and the *hardware abstraction layer* (HAL). The service functions encompass multiple managers, each dedicated to a specific cryptographic operation or the management of the HSM4SSL. Finally, the top layer is the application layer, where the SSL protocol and software applications reside. This layer leverages the

underlying software and hardware components through an *application programming interface* (API) to use the various HSM4SSL *security services*. The security services provide a wide range of cryptographic operations and HSM-related functions. Some of the key security services offered by HSM4SSL include the following:

1. Secure storage initialization.
2. Key pair generation.
3. Digital certificate generation.
4. Data encryption/decryption.
5. Access control and authorization.

Moreover, HSM4SSL is a modular architecture designed to provide uniform integration across various network environments and configurations. This adaptability is essential for addressing legacy and modern systems' different security requirements and designs. This is achieved by the usage of uniform CLI commands that provide a consistent interface to eliminate overhead when using HSM4SSL's services in pre-existing infrastructures. Plus, the ADCP parameters can be dynamically adjusted to align with the exact requirements of both advanced and legacy systems, and the HAL can also be effortlessly modified to include HSMs' drivers when more recent HSMs are introduced in the physical layer to fulfill the organization's security requirements.

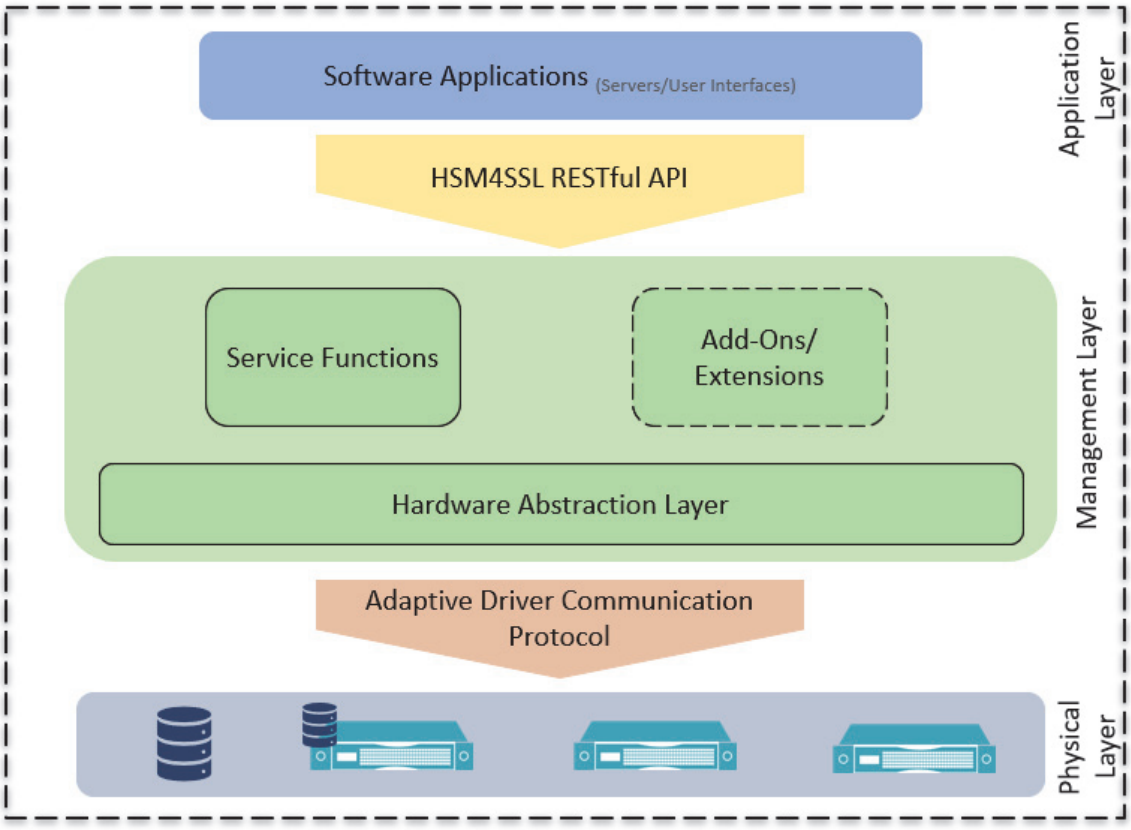


Figure 2. HSM4SSL high-level architecture.

3.3. HSM4SSL: Low-Level Architecture

The low-level architecture of HSM4SSL, shown in Figure 3, offers an in-depth exploration of the internal components and their intricate interactions within the system.

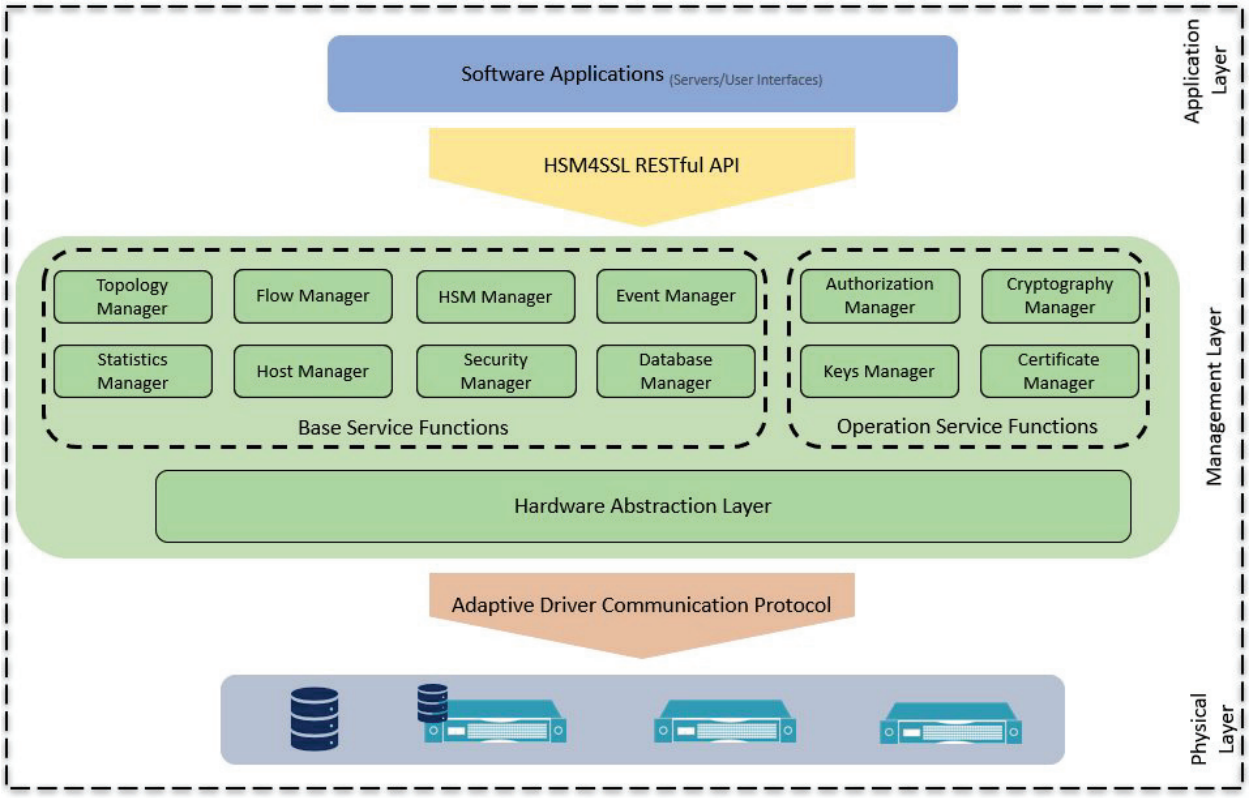


Figure 3. HSM4SSL low-level architecture.

3.3.1. HSM4SSL Application Layer

The application layer of HSM4SSL includes software applications running on intra-domain servers. These servers utilize the security services utilized by HSM4SSL by interacting with the management layer via an API to provide unified and standard access to the available HSMs.

HSM4SSL API: To establish a unified access point for software applications to the architecture, we developed RESTful APIs that adhere to the REST architecture. These APIs utilize HTTP requests to communicate with resources identified by URLs, enabling the exchange of information through standard HTTP methods like GET, POST, PUT, and DELETE. The API design process was complemented by conducting a comparative analysis of the currently most-utilized API development tools, which are FastAPI, Flask, Spring, and Django. Figure 4 presents a performance comparison of the aforementioned tools based on the number of RPS that each framework can handle as the number of concurrent users increases from 0 to 40. From the results, Django demonstrates the highest RPS rate across the board. However, FastAPI preserves a consistent RPS rate, suggesting robust performance under a concurrent user load. Further, the latency comparison shown in Figure 5 depicts the latency measured in milliseconds as the number of concurrent users increases from 0 to 40. As the user load rises, all frameworks exhibit an increase in latency. FastAPI shows the lowest increase, suggesting it handles the additional load with minimal latency impact.

After evaluating the performance of various web frameworks, we chose FastAPI for its impressive balance of handling a high number of requests per second and its low latency. FastAPI's performance metrics indicate that it can handle a significant number of requests with minimal delay, which is essential for ensuring a smooth user experience. Prioritizing a lower latency is crucial for our application, as it directly impacts the responsiveness and efficiency of the service.

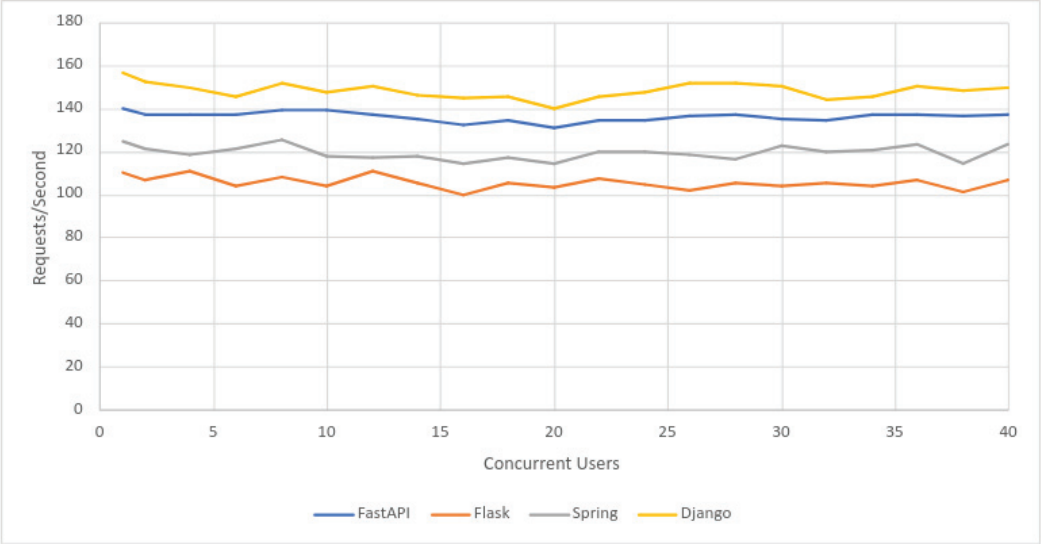


Figure 4. RPS comparison results.

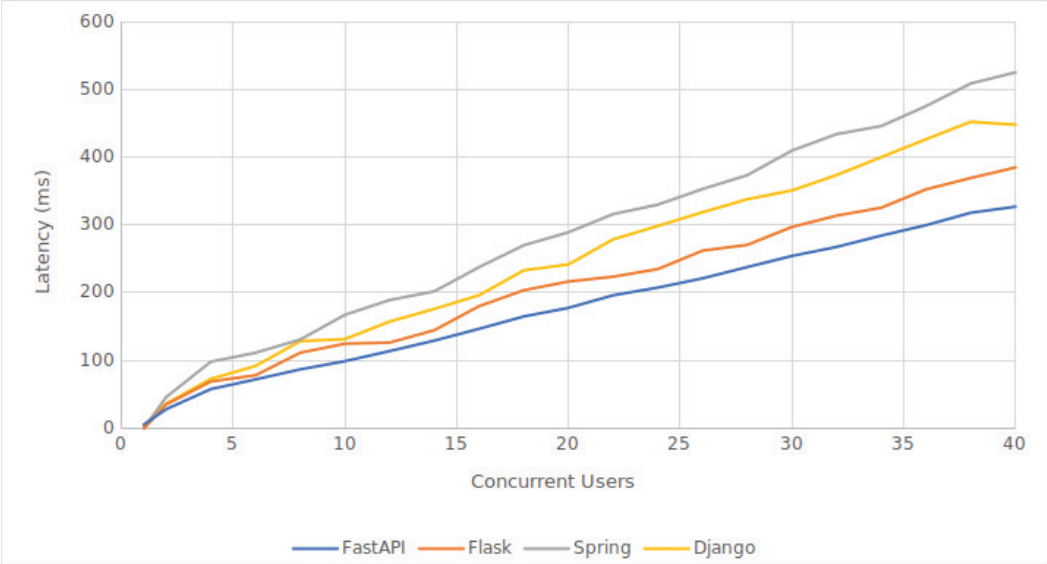


Figure 5. Latency comparison results.

3.3.2. HSM4SSL Management Layer

Exposing the higher-level API to communicate with the underlying HSMs happens in the management layer, where HSM4SSL's centralized control and monitoring mechanisms come into play. This is primarily made up of multiple service managers that can be classified as follows:

1. Base Service Functions:
   The base service functions are employed to provide the primary services required to properly operate the HSM4SSL. The base service functions include the following:
   - Topology Manager: Stores information about the HSMs and organizes HSMs with their physical and logical connections in a hierarchical structure. The topology manager monitors newly added and removed HSMs to construct this structure dynamically.
   - Statistics Manager: Collects statistical information from the underlying HSMs, such as request counts and error rates. It stores and maintains this information to comprehensively generate reports based on the collected statistics.

- Flow Manager: Creates policy sets that guide the request routing process. These policy sets define the rules and criteria for determining the appropriate destination for each incoming request. By configuring these policies, the flow manager guides the HAL, enabling it to determine the specific HSM to which a request should be sent. These policies can consider various factors, such as security requirements, load balancing, the availability of HSM resources, and other relevant considerations.
- Host Manager: Manages and stores connected endpoint hosts' information. It maintains a repository that stores crucial details about each host, including their MAC addresses and IP addresses.
- HSM Manager: Manages and stores the underlying HSMs' information. It maintains a centralized repository of HSM-related data, including their statuses and configurations. This information includes details about the available HSMs, their connectivity, operational status, and any relevant configuration settings.
- Security Manager: Analyzes the HSMs' states and incoming data to identify anomalies. Maservice managerlearning-based anomaly detection systems can be employed in the security manager to continuously scan incoming requests and detect threats.
- Event Manager: Collaborates closely with the security manager. Its primary role is to monitor critical events within the HSMs. By continuously monitoring various aspects of the system, such as performance and security indicators, in any critical event, such as a security breach or system failure, the event manager generates reports and alerts. Moreover, in case of an HSM failure error generated by the security manager, the event manager communicates with the flow manager with a specific error code requiring it to update the policy set. This allows the HAL to route incoming requests to available HSMs until the error is handled.
- Database Manager: Manages and backs up data storage and generates statistics about saved data, such as certificates and audit logs.

2. Operation Service Functions: On the other hand, operation service functions contain managers oriented to perform specific cryptographic and HSM-related tasks, such as the following:
    - Authorization Manager: Performs host authentication and authorization and determines which hosts can access HSM resources. It is responsible for validating the identity and credentials of connecting hosts and determining their level of access to HSM resources. By enforcing access control policies and rules, the authorization manager ensures that only authorized hosts can utilize the HSM resources and perform cryptographic operations.
    - Cryptography Manager: Performs cryptographic operations, such as encryption and decryption, and secure storage initialization within the HSMs to store cryptographic objects.
    - Keys Manager: Handles key generation and retrieves cryptographic keys for hosts for encryption and decryption processes.
    - Certificate Manager: Manages the validation and revocation of digital certificates used for secure communication and authentication purposes. It maintains a repository of trusted root certificates and intermediate certificates, enabling the validation of certificates within the system. The certificate manager also handles the revocation of certificates in case of compromised or expired certificates.

3.3.3. Hardware Abstraction Layer

The HAL, shown in Figure 6, is the heart of the HSM4SSL. It enables software applications to communicate with different HSMs by abstracting the low-level details of the underlying HSMs. It contains a *service manager* and a *driver manager*. The service manager acts as an intermediary between the controller managers and the underlying hardware, while the driver manager handles the execution of hardware-specific commands.

The HAL communicates with the flow manager to obtain the policy set. The policy set provides instructions on various aspects, such as security measures, access control, error handling, and resource utilization, based on analyzing the security requirements, capabilities, and current utilization of each HSM. Additionally, the policy set may include guidelines for access control, specifying which entities or users have permission to perform certain operations on the HSMs.

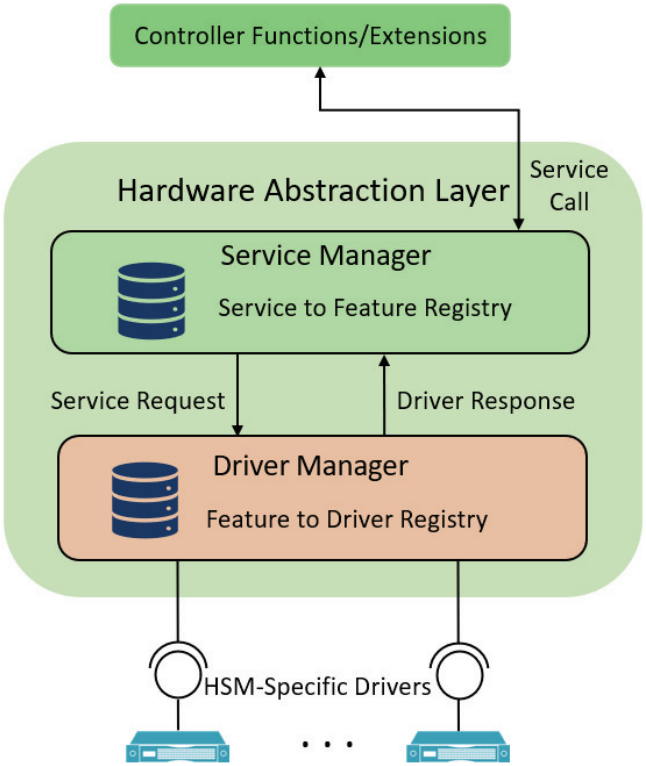


Figure 6. The HSM4SSL hardware abstraction layer (HAL).

When a request is received from the controller managers, the service manager translates the request into a feature request. This translation process entails applying the policy set to ensure the inclusion of the appropriate destination HSM parameter in the message. By adhering to these policies, the service manager guarantees that the feature request contains the necessary information to route it to the correct HSM. The driver manager maintains a registry that maps each feature request to the appropriate HSM's command and driver. This registry serves as a lookup table, allowing the driver manager to identify the specific hardware command and corresponding driver that can handle the requested feature.

3.3.4. Adaptive Driver Communication Protocol (ADCP)

Communication between different components is facilitated using the ADCP. The ADCP is a JSON-based messaging protocol that forms the backbone of the communication infrastructure within HSM4SSL. The standardized format, shown in Figure 7, provides a structured approach to exchanging information and instructions. The message is organized into two primary objects: the header and the body.

```
{
    "header":{
        "message_direction": "request/response",
        "message_id": "msg_id",
        "message_type": "requested_operation",
        "source_mac_address": "src_mac",
        "source_ip_address": "src_ip",
        "timestamp": "2023-11-14T16:00:36.937948",
        "dest_hsm_module": "hsm_destination",
},
    "body":{
        "session_label": "session_identifier",
        "credentials":{
            "admin_pin": "admin_access_code",
            "user_pin": "user_access_code"
        },
        "key_info":{
            "type": "generated_key_type",
            "length": "generated_key_length",
            "label": "generated_key_label",
            "id": "generated_key_id"
        },
        "csr_details"{
            "csr": "csr_data",
            "csr_file": "csr_path"
        },
        "cert_details": {
             "hash_algorithm": "signing_algorithm",
             "days": "certificate_validity"
         }
    }
}
```

Figure 7. The adaptive driver communication protocol (ADCP) standard JSON-based message.

The header object includes keys such as *message_direction*, which denotes if this message is a request for a service or a response to the client. This key can also hold a value to indicate that a message is for internal purposes, such as messages between managers for information exchange. Another key is the *"message_id"*, which assigns a unique serial number to each message, starting from "1" and incrementing with each new message. This enables the tracking and referencing of messages within the system. The *"message_type"* key indicates the specific HSM operation to be performed. The *"source_mac_address"* key represents the MAC address of the device or entity sending the message, enabling internal architecture usage and the association of cryptographic elements with their corresponding hosts. Similarly, the *"source_ip_address''* key denotes the IP address of the source device, providing additional internal architecture usage information. Further, the *"timestamp"* key in the header serves to record the time at which the message is generated, providing a temporal reference for tracking and synchronization purposes within the architecture. Additionally, the header contains the *"dest_hsm_module"* key, which specifies the destination HSM where the command should be executed. It identifies the path to the HSM library file responsible for handling the requested operation. By incorporating these standardized header objects, the message format ensures consistency, interoperability, and efficient processing across different HSM-related operations and systems.

The body object of the message format, on the other hand, encapsulates the specific keys and values related to the intended operation or task. Within the body section, additional keys and values that are specific to the operation defined in the header are included. For instance, the *"session_label"* key represents the label or name assigned to the secure session being initialized or an identification of where to store generated cryptographic elements. This allows for the proper identification and management of different sessions within the HSM. The *"credentials"* sub-object contains the *"admin_pin"* and *"user_pin"* keys. These credentials are utilized for administrative and user-level operations, granting access to the token's cryptographic functions and enabling authentication and authorization. Furthermore, the *"key_info"* sub-object encompasses keys related to the generation of cryptographic key pairs. The *"type"* key specifies the type of cryptographic algorithm to be used, such as RSA, AES, or EC. The *"length"* key denotes the desired length of the generated encryption key. Additionally, the *"label"* key assigns a unique name or label to the generated public-key pair, facilitating identification and management. Finally, the *"id"* key provides an ID number for the generated public-key pair, enabling easy referencing and retrieval when needed. Moreover, for X.509 certificate generation requests, the *"csr_details"* sub-object serves as a repository for storing critical CSR details. These details encompass two key attributes: *"csr"*, which encapsulates the CSR data received from the client for signing, and *"csr_file"*, which specifies the file path where the CSR is to be saved within the HSM4SSL server. This approach of saving the CSR file before proceeding with the signing process significantly streamlines the certificate generation process. Along with the *"csr_details"*, the second sub-object, *"cert_details"*, is also involved in the certificate generation security service. *"cert_details"* includes two important keys. First is the *hash_algorithm*, which denotes the hashing algorithm to be used for signing the certificate. If not specified by the user, the *hash_algorithm* is *"sha256"* by default. The second is *"days"*, which specifies the certificate validity in days, which is hard-coded *"365"* days and can be changed by the HSM4SSL administrators.

3.3.5. ADCP Assignment Policy Set

The effective operation of the HAL depends on a well-structured JSON-based assignment policy set. The policy set design is a critical aspect of the HSM4SSL architecture, as it determines the proper assignment of requests to the available HSMs based on predefined policies. The policy set is designed to consider various factors, including the capabilities, algorithms, hash functions, and utilization of each HSM, which ensures efficient resource utilization and optimal performance. For example, the policy set shown in Figure 8 includes the *"hsm"* key, which identifies the specific HSM to which the policy applies. For each HSM, the *"capabilities"* key specifies the capabilities of the HSM, indicating the operations or functions it can perform. For example, the policy set may specify that HSM1 is capable of public-key pair generation, certificate generation, and encryption, while HSM2 can only perform key pair generation and encryption. Plus, the *"algorithms"* key lists the supported cryptographic algorithms for the specified HSM. It includes algorithms such as RSA, ECC, AES, and RC4, which can be used for various cryptographic operations. The *"hash_funcs"* key defines the supported hash functions for the HSM. Hash functions like MD5, SHA1, SHA256, and SHA512 are commonly used for data integrity and verification purposes. Finally, the *"utilization"* key reflects the current utilization level of the HSM. It can be categorized into different levels, such as low, medium, or high, based on factors like the workload, performance, and resource availability of the HSM, enabling efficient decision-making in request routing.

By designing a comprehensive and well-defined assignment policy set, the HSM4SSL architecture can effectively optimize resource allocation and ensure compatibility with cryptographic operations.

```
policies: [
{
    "hsm": "HSM1",
    "capabilities": ["initsession",
    "keypairgen", "certificate_gen"],
    "algorithms": ["rsa","ecc"],
    "hash_funcs": ["md5","sha1","sha256"],
    "utilization":["med"],
},
{
    "hsm": "HSM2",
    "capabilities": ["initsession",
    "keypairgen", "encryption"],
    "algorithms": ["rsa","aes","rc4"],
    "hash_funcs": ["md5","sha256","sha512"],
    "utilization": ["high"],
}
]
```

Figure 8. HSM4SSL JSON-based assignment policy set.

3.3.6. HSM4SSL Physical Layer

The physical layer includes the hardware components and infrastructure necessary for the secure operation of the system, such as the following:

- HSMs: The physical layer includes tamper-resistant HSMs that provide cryptographic services. HSM4SSL relies on the security of those HSMs as they adhere to strict security standards such as FIPS 140-2, which are engineered to protect sensitive data and manage cryptographic keys with unmatched security, thereby strengthening the SSL operations, which are crucial for secure intra-domain communication. Plus, to accommodate various needs and requirements and to allow for greater flexibility, HSM4SSL supports an array of HSMs from multiple vendors. This multi-vendor approach ensures that organizations are not locked into a single provider, thereby reducing the risk of vendor lock-in and allowing for a customized security fit. The integration process is streamlined, ensuring that adding a new HSM to the environment is as simple as possible. Moreover, in light of the evolving landscape of cybersecurity threats due to the arrival of quantum computers, HSM4SSL must evolve at a comparable pace. Of note is that the National Institute of Standards and Technology (NIST) has selected four cryptographic algorithms covering key exchange and digital signatures that are pending standardization. However, the development of post-quantum safe HSMs is considered a hot topic by leading HSM manufacturers, such as Thales and Crypto4A [30,31], as building a post-quantum safe HSM requires completely different cryptographic frameworks compared to those employed in contemporary HSMs. Moreover, as HSM4SSL is designed for quick adaptability, we are certain that as these post-quantum safe algorithms and HSMs are standardized and developed, HSM4SSL will be able to incorporate these HSMs into the architecture.
- Databases: For supplementary storage capacity, the physical layer may include databases that store cryptographic elements, such as keys, certificates, and audit logs. These databases can be located either on the same server as the HSMs or on a separate server.
- Virtual HSM: In addition to physical HSMs, the physical layer supports the virtual HSM, also known as a HSM simulator. It is important to note that the HSM4SSL architecture has been developed and thoroughly tested in the SoftHSM simulator environment. SoftHSM provides a software-based emulation of an HSM and offers a controlled environment for cryptographic operations and secure storage [32].

The physical layer is designed with scalability in mind. As new cryptographic challenges arise and industrial needs change, HSM4SSL's physical layer can easily incorporate newer HSMs that meet these evolving standards. Plus, the HAL can be dynamically updated to include the newly added HSM's driver to facilitate communication. This process allows HSM4SSL to leverage the latest HSM technology without requiring any modifications to the system's core architecture and ensure long-term viability and effectiveness.

3.3.7. HSM4SSL CLI Tool

The developed *"ssleverywhere"* command-line interpreter (CLI) tool serves as a versatile interface for interacting with the HSM4SSL architecture from the terminal. It offers various commands to initiate secure sessions, generate certificates, and perform essential cryptographic operations. The design of this tool prioritizes user-friendliness, enabling users to effortlessly harness the capabilities of the HSM4SSL architecture.

Once installed and configured, the *"ssleverywhere"* tool generates several folders within the system, namely *"private"*, *"csr"*, and *"certs"*. Within these folders, specific functionalities are organized. The *"private"* folder serves as the repository for server key pairs. The *"csr"* folder is designated for CSRs to be sent to the HSM4SSL, be signed, and create X.509 certificates. Finally, the *"certs"* folder stores the issued certificates obtained after successful CSR submissions. The *"ssleverywhere"* tool accepts multiple parameters, each serving a specific purpose and contributing to the customization of commands. For example, the *'-gen_cert'* command is responsible for creating an X.509 certificate and saving it in the *"certs"* folder. The process starts by providing the following parameters:

- *-name*: Specifies the name associated with the certificate; this name can either be manually entered or can be modified by administrators to remain consistent for repeated usage of the command.
- *-default*: An option that can be used with the *"-gen_cert"* command to indicate default certificate generation settings. Default parameters include key type set to *RSA*, key length set to *2048*, encryption set to *AES*, and hash set to *SHA256*.
- *-manual*: An option that can be used with the *"-gen_cert"* command to enable manual configuration of certificate generation settings.
- *-key_type*: Specifies the type of private key when manually configuring certificate generation settings.
- *-key_length*: Specifies the desired length of the private key when manually configuring certificate generation settings.
- *-encryption*: Specifies the encryption method to be used for private keys when manually configuring certificate generation settings.
- *-hash*: Specifies the hashing algorithm for cryptographic operations when manually configuring certificate generation settings.

Another command is *"-keypair_gen"*. This command is used to request HSM4SSL to generate a cryptographic key pair and return it to the user. It accepts three parameters:

- *-label*: Specifies a label or name for the generated key pair, so the public key can be saved in the HSM4SSL database.
- *-key_type*: Specifies the type of cryptographic key to be generated.
- *-length*: Specifies the desired length of the key.

The usage example of the *ssleverywhere* tool involves two machines on the same network establishing secure communication. Machine A, the server, generates a certificate named "SE" using the *ssleverywhere* command:

```
$ ssleverywhere -gen_cert -name SE -default
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  5978  100  4965  100  1013   108k  22678 --:--:-- --:--:-- --:--:--  132k
```

Machine B, the client, follows a specific configuration to generate its certificate:

```
$ ssleverywhere -gen_cert -name CL -manual -key_type rsa -key_length 4096
-encryption des3 -hash sha512
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  7059  100  5331  100  1728   28048   9091 --:--:-- --:--:-- --:--:--  37951
```

Machine B generates a certificate named "CL" with specific configurations: RSA key type, 4096-bit key length, DES3 encryption, and SHA-512 hashing algorithm. After successful generation, both machines have their certificates, enabling secure communication.

Both machines possess certificates from the HSM4SSL architecture, as per their specific requirements. To utilize these certificates, Machine A acts as the server, employing a Python socket programming code to listen for connections on a predetermined port. It loads its SSL certificate and the root CA certificate into the SSL/TLS server context, awaiting client connections. Machine B initiates a connection to Machine A's server using its own SSL certificate and the root CA certificate in its SSL/TLS client context. The SSL handshake process occurs upon connection establishment, facilitating mutual authentication. Once completed, a secure channel is established, allowing encrypted and integrity-protected application data exchange between the client and server. With both programs running and configurations in place, the SSL handshake process, as captured by Wireshark and illustrated in Figure 9, unfolds when Machine B connects to Machine A. During this handshake, the messages enclosed within the yellow rectangle depict the SSL handshake process, which involves mutual authentication. After the handshake is completed, both the client and server form a secure channel and can exchange application data securely, which is encrypted and integrity-protected.

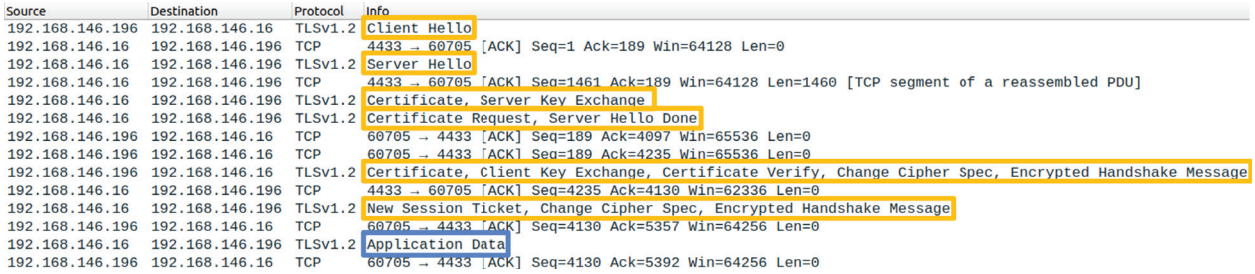


Figure 9. Captured handshake process.

4. HSM4SSL Performance

To perform a valid performance evaluation of the HSM4SSL architecture where three HSM simulators are deployed, it was compared with *OpenSSL* version 3.0.11, a benchmarking tool for performing cryptographic operations. *OpenSSL* is a comprehensive cryptographic software and command-line tool that is well known for its open-source implementation of the TLS protocol. Users and developers can use *OpenSSL* to perform a wide range of cryptographic functions, such as generating key pairs, CSRs, and more [33]. Note that the sole usage of *OpenSSL* in sensitive intra-domain environments being to act as a root of trust for certificate signing and generation introduces security risks, as there have been 177 vulnerabilities reported in OpenSSL through the period of 2002 to 2019 [34].

The experimental setup involved testing on two separate PCs within the same network environment to ensure a consistent and controlled testing framework. The primary machine, acting as the server and used for hosting the HSM4SSL architecture, had the specifications detailed in Table 2. The secondary machine, responsible for generating certificate requests and performing benchmarks, was equipped to emulate realistic operational conditions. The simulated environment used SoftHSM version 2.6.1, an open-source software implementation of a cryptographic store accessible through a PKCS#11 interface. This setup allowed for the testing of the HSM4SSL architecture's integration with HSM capabilities and without the need for physical hardware. SoftHSM was configured with the default settings, such as its support for ECC, AES, and RSA for cryptographic algorithms,

and non-paged memory was maintained in its enabled state to prioritize security over performance, ensuring that sensitive data were kept out of disk-based swap spaces, thus minimizing the risk of data leakage. Plus, P11-kit integration was enabled, facilitating the interaction of SoftHSM with PKCS#11 modules installed on the system to specify how to allow the user to communicate with the cryptographic tokens. Moreover, SoftHSM uses the Botan cryptographic library by default for its cryptographic operations.

The network connecting these devices was configured with a bandwidth of 127 MB/s and a latency of 5 ms between the two devices, mimicking intra-domain network conditions that might impact performance metrics.

Table 2. Hardware specifications.

Specification	Details
Operating System	Ubuntu 22.04
Processor	AMD Ryzen 7
Memory	16 GB
Hard Drive	1 TB SSD

As for the software tools, we used the *loadtest* command line and Node.js module version 8.0.3, which simulates high volumes of traffic to web applications to assess their performance, identify bottlenecks, and understand how they handle various levels of load [35]. It operates in various modes, such as by specifying the requested RPS, the total number of requests to be processed, the total time to send as many requests as the API can handle, or the number of concurrent users. In addition, it provides detailed metrics and reporting, including RPS, mean latency, and error rates. The *loadtest* tool was used to simulate real-world high traffic levels and interaction with the HSM4SSL architecture to understand how it scales under stress. Additional tools, such as the *top* (table of processes), which provides real-time insight into active processes and presents a system information view that includes resource utilization data, and *nload*, which monitors real-time network traffic, were employed to monitor performance metrics beyond what loadtest offers.

4.1. Testing Scenarios and Metrics

To ensure precise evaluation, a separate PC was used to carry out all test scenarios using *loadtest*. These scenarios mainly involved varying two parameters, which are the number of concurrent users, which refers to the number of virtual users that are actively making requests to the target application or API at the same time, and the number of requests, which indicates the total number of HTTP requests that will be generated and sent to the API. Moreover, the key evaluation metrics used are the effective RPS, which measures the rate at which the system can handle incoming certificate generation requests; latency, which refers to the time delay or the elapsed time between the initiation of a request and the moment when a response is received; CPU usage, to measure the amount of resources consumed by the CPU during the test; and network bandwidth, which measures the amount of data transmitted over the network during the load test.

4.2. Results and Discussion

Figure 10 shows that both HSM4SSL and OpenSSL have a roughly steady and consistent RPS rate as the number of concurrent users increased from 1 to 100, suggesting that both systems demonstrated an ability to handle an increasing workload without a significant drop in RPS, emphasizing their scalability under the test conditions. However, HSM4SSL consistently exhibits a higher RPS compared to OpenSSL at various user load levels. This indicates the remarkable efficiency and scalability of HSM4SSL in processing a greater number of requests in parallel. Plus, Figure 11 reveals that, as the number of concurrent users increases in the same range, both HSM4SSL and OpenSSL exhibit a gradual increase in latency, which is expected as system loads intensify. However, on

average, the latency for each request served by OpenSSL tends to be 13.08% higher than that of HSM4SSL.

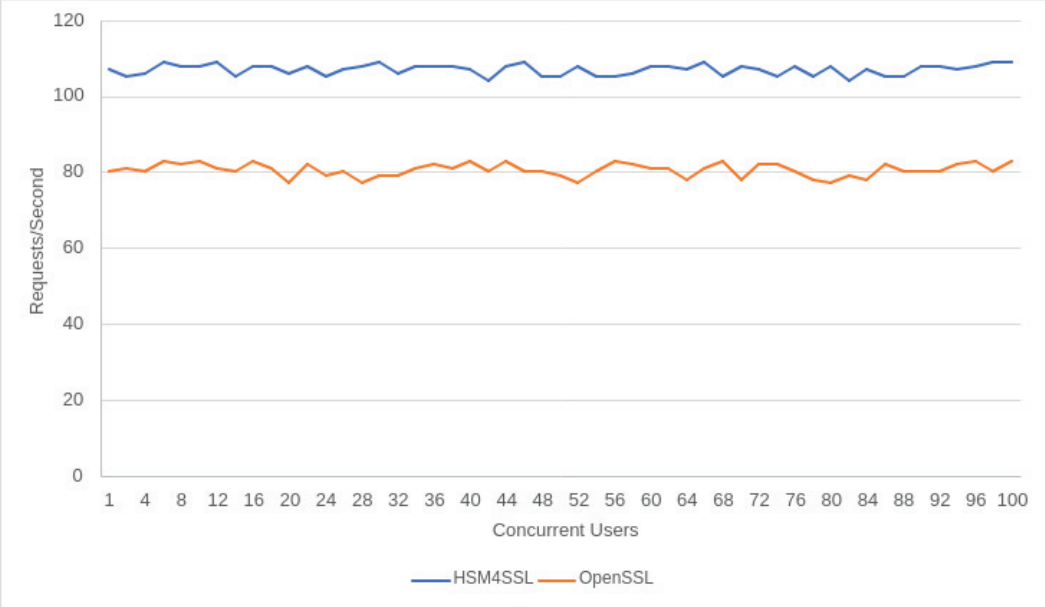


Figure 10. RPS comparison results.

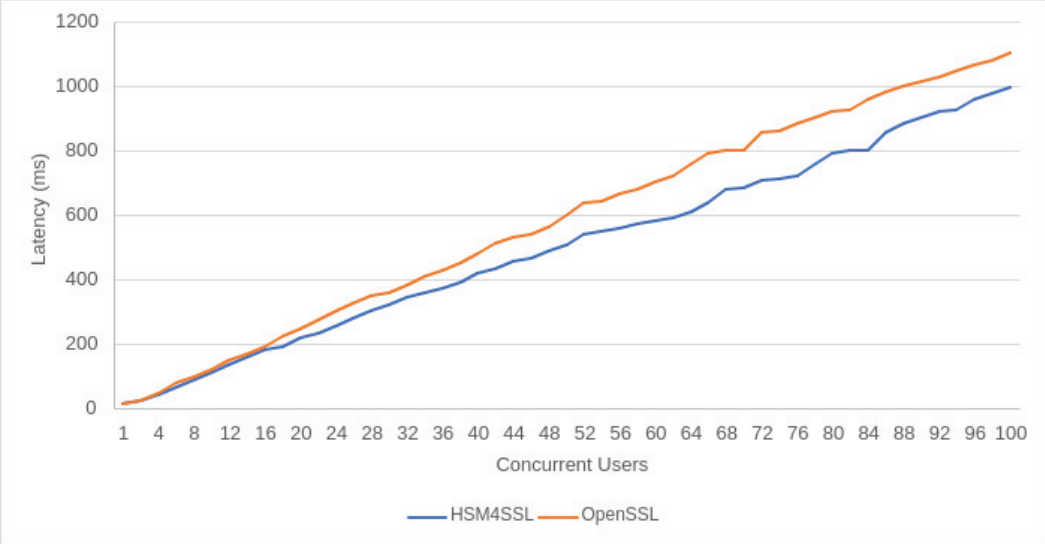


Figure 11. Latency comparison results.

Another evaluation criterion is the evaluation of resource and network utilization, where CPU usage was determined by closely monitoring essential components, namely Python, Google Chrome, and the MySQL database, for the HSM4SSL architecture. An interesting observation from the results shown in Figure 12 is the clear and consistent logarithmic growth exhibited by HSM4SSL and OpenSSL. However, HSM4SSL's growth in CPU usage is more pronounced compared to OpenSSL since HSM4SSL includes more complex backend processes, such as generating the ADCP message, forwarding the message to multiple managers, generating the policy set, and accessing databases to save or retrieve information. Throughout the tests, HSM4SSL showcases an ability to efficiently utilize CPU resources, as evidenced by the substantial growth in CPU usage. This growth, although higher, eventually stabilizes at a maximum of 15.6%. This is because the application or

processes included in HSM4SSL may be configured with resource allocation limits that restrict its ability to consume more than a certain percentage of the CPU's total capacity.

The network utilization concentrates on understanding the influence of certificate generation on network resources, specifically focusing on HSM4SSL. Notably, OpenSSL operates and is utilized locally within each server, making the monitoring of network utilization primarily relevant to HSM4SSL. Figure 13 shows a consistent upward trajectory in network bandwidth consumption by HSM4SSL. However, it is noteworthy that the network bandwidth utilization eventually stabilizes at approximately 690–700 KB/s. This cap indicates that HSM4SSL, while efficiently scaling to meet growing demands, has an upper limit in terms of network resource utilization. This stability underscores the architecture's ability to effectively allocate and manage network resources without excessive saturation.



Figure 12. CPU usage comparison results.

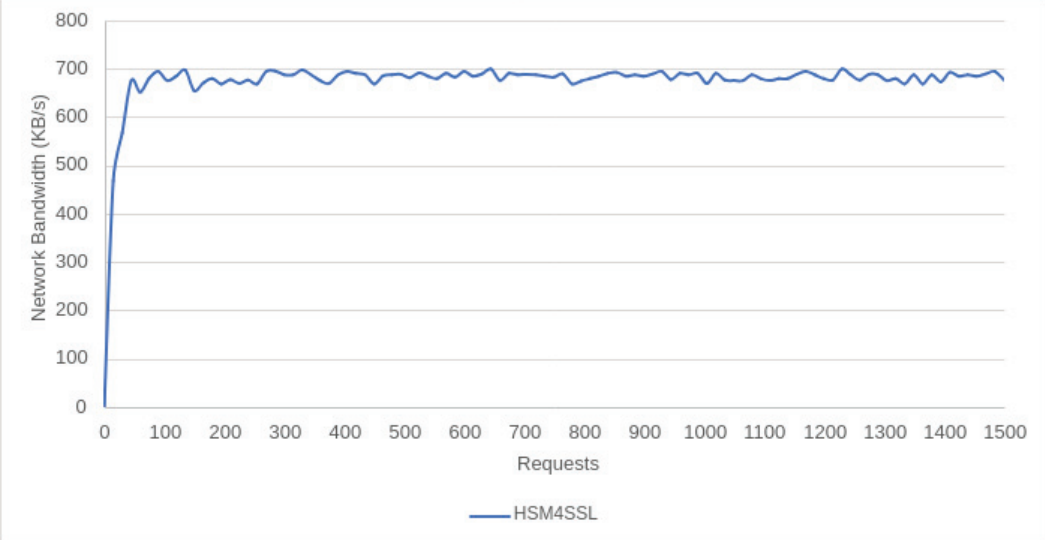


Figure 13. Network bandwidth comparison results.

Based on the evaluation outlined, multiple methods can be involved to enhance the performance of HSM4SSL. The evaluation of HSM4SSL shows a weakness in CPU usage. To reduce CPU utilization caused by MySQL processes, one strategy is to implement database-caching mechanisms. Caching frequently accessed data can reduce the computational load on the CPU by minimizing the need to repeatedly retrieve data from the database. Plus, by utilizing more parallelization techniques, HSM4SSL can reduce the latency and processing

time by enabling the simultaneous execution of multiple certificate generation requests. Moreover, by performing request throttling to limit the number of requests from a single user, HSM4SSL can prevent potential spikes in resource usage and maintain a smoother and more predictable operation. This approach can help ensure fair resource allocation, reduce the risk of resource contention, and mitigate the impact of excessive concurrent requests on system performance.

5. Conclusions

This paper addresses the current security issues for communication in intra-domain environments, as most peer-to-peer communication is unencrypted and unauthenticated, and proposes the HSM4SSL architecture, which appeared as a novel solution to address the evolving challenges of secure communication within intra-domain environments. This research has discussed the architecture's design, components, and performance evaluation, providing insights into its capabilities and contributions.

HSM4SSL is a unified and standardized architecture that aims to strengthen the security of intra-domain environments. Its integration of hardware security modules has contributed to its robustness by providing secure storage and key management. This implementation ensures that cryptographic keys and sensitive data are protected against unauthorized access and tampering, reinforcing the architecture's commitment to security. Plus, comparing HSM4SSL with the widely used OpenSSL demonstrates the former's superior ability to manage requests more efficiently, with higher requests per second and lower latency while ensuring the effective utilization of CPUs and network resources. This performance advantage underscores HSM4SSL's potential to serve as a reliable foundation for secure communication within intra-domain environments. However, it is important to mention that the provided comparative analysis with OpenSSL represents preliminary results, which were aimed at providing a foundational overview of HSM4SSL's performance against a well-established tool within the security realm. Looking forward, we acknowledge the importance of extending our comparative analysis to include a broader range of security tools and technologies, where we plan to perform a comprehensive evaluation that not only revisits our preliminary findings with OpenSSL but also encompasses additional comparisons with other cryptographic tools. Plus, it is crucial to focus on a comprehensive security analysis of HSM4SSL against relevant architectures, including a detailed evaluation of the API in the face of well-known security attacks, along with developing threat models that enable us to simulate various attack victors and assess the system's response under different threat scenarios.

However, HSM4SSL introduces challenges such as the dependency on physical HSMs, which introduces cost complexities, especially when multiple HSMs are deployed. Plus, the architecture, while optimized for multiple HSM integration, may encounter latency and overhead, especially when integrating a single HSM within the premises. Furthermore, despite efforts to ensure system robustness, certain components could become failure points if not managed properly, such as the central management of HSMs, the API, and the centralized policy management system. Additionally, the necessity to regularly update both the HSMs and software components to align with evolving security standards underscores our commitment to continual improvement and adaptation.

In summary, HSM4SSL represents notable progress in tackling security issues in intra-domain situations. However, there are various areas for future study and development that might improve its capabilities and influence. Further research might be conducted to improve the compatibility of HSM4SSL with a wider array of HSMs, therefore enabling enterprises to select from a varied selection of HSM alternatives while upholding the security requirements and principles of the architecture. This presents an opportunity to investigate sophisticated load-balancing methodologies, caching methods, and parallel processing strategies in order to minimize latency and enhance throughput. Furthermore, future endeavors may encompass initiatives to advocate HSM4SSL as a widely accepted option for ensuring secure communication within a certain domain. The attainment of widespread

acceptance may necessitate collaboration with pertinent standards bodies and industry groups. For the time being, HSM4SSL codes and API are under development to meet the standards of security, functionality, and scalability, where the aim is to provide comprehensive documentation, including detailed guides on endpoints, usage examples, and configuration instructions, to support future exploration and validation of "HSM4SSL"'s capabilities.

Author Contributions: Conceptualization, Y.A. and A.O.; methodology, Y.A. and A.O.; software, Y.A.; validation, Y.A. and A.O.; formal analysis, Y.A. and A.O.; investigation, Y.A. and A.O.; resources, Y.A. and A.O.; data curation, Y.A. and A.O.; writing—original draft, Y.A.; writing—review and editing, Y.A. and A.O.; project administration, A.O.; funding acquisition, A.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data is not accessible to the public due to its involvement in an ongoing study.

Acknowledgments: During the preparation of this work the authors used Grammarly tool in order to check the English grammar of the manuscript. After using this tool, the author(s) reviewed and edited the content as needed and take full responsibility for the content of the publication.

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

References

1. Zhang, M. *On the State of the Inter-Domain and Intra-Domain Routing Security*; University of Oregon: Eugene, OR, USA, 2016; p. 23.
2. Schulze, H. 2020 Insider Threat Report. Available online: https://www.cybersecurity-insiders.com/wp-content/uploads/2019/11/2020-Insider-Threat-Report-Gurucul.pdf (accessed on 22 April 2024).
3. Bourke, T. *Server Load Balancing*, 1st ed.; O'Reilly: Beijing, China; Sebastopol, CA, USA, 2001.
4. Membrey, P.; Hows, D.; Plugge, E. SSL Load Balancing. In *Practical Load Balancing*; Apress: Berkeley, CA, USA, 2012. [CrossRef]
5. SectigoStore. SSL Certificate for IP Address—An Expert Guide on SSL for IP Address. Available online: https://sectigostore.com/page/ssl-certificate-for-ip-address/ (accessed on 22 April 2024).
6. Sectigostorepages. Google Announces Intentions to Limit TLS Certificates to 90 Days: Why Automated CLM Is Crucial. Available online: https://www.sectigo.com/resource-library/google-announces-intentions-to-limit-tls-certificates-to-90-days-why-automated-clm-is-crucial (accessed on 22 April 2024).
7. Aref, Y.; Ouda, A. Still Computers Networking is Less Secure Than It should be, Causes and Solution. In Proceedings of the 2023 International Symposium on Networks, Computers and Communications (ISNCC), Doha, Qatar, 23–26 October 2023; pp. 1–8. [CrossRef]
8. Herzberg, A.; Hollick, M.; Perrig, A. Secure Routing for Future Communication Networks (Dagstuhl Seminar 15102). *Dagstuhl Rep.* **2015**, *5*, 28–40. [CrossRef]
9. Shue, C.A.; Kalafut, A.J.; Gupta, M. A Unified Approach to Intra-domain Security. In Proceedings of the 2009 International Conference on Computational Science and Engineering, Vancouver, BC, Canada, 29–31 August 2009. [CrossRef]
10. Boisrond, P.D. To Terminate or Not to Terminate Secure Sockets Layer (SSL) Traffic at the Load Balancer. *arXiv* **2020**, arXiv:2011.09621.
11. Rescorla, E. RFC2818; HTTP over TLS. Available online: https://www.rfc-editor.org/info/rfc2818 (accessed on 22 April 2024).
12. W3Techs. Usage Statistics of Default Protocol Https for Websites. Available online: https://w3techs.com/technologies/details/ce-httpsdefault (accessed on 22 April 2024).
13. Chordiya, A.R.; Majumder, S.; Javaid, A.Y. Man-in-the-Middle (MITM) Attack Based Hijacking of HTTP Traffic Using Open Source Tools. In Proceedings of the 2018 IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 3–5 May 2018. [CrossRef]
14. Fung, A.P.H.; Cheung, K.W. SSLock: Sustaining the trust on entities brought by SSL. In Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security—ASIACCS '10, Beijing, China, 13–16 April 2010. [CrossRef]
15. Chang, L.; Hsiao, H.C.; Jeng, W.; Kim, T.H.J.; Lin, W.H. Security Implications of Redirection Trail in Popular Websites Worldwide. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017. [CrossRef]
16. Hodges, J.; Jackson, C.; Barth, A. HTTP Strict Transport Security (HSTS). RFC 6797 (Proposed Standard). Available online: https://www.rfc-editor.org/info/rfc6797 (accessed on 22 April 2024).
17. Dolnak, I.; Litvik, J. Introduction to HTTP security headers and implementation of HTTP strict transport security (HSTS) header for HTTPS enforcing. In Proceedings of the 2017 15th International Conference on Emerging eLearning Technologies and Applications (ICETA), Stary Smokovec, Slovakia, 26–27 October 2017. [CrossRef]

18. Poddebniak, D.; Ising, F.; Böck, H.; Schinzel, S. Why TLS is better without STARTTLS: A Security Analysis of STARTTLS in the Email Context. In Proceedings of the 30th USENIX Security Symposium (USENIX Security 21), Virtual Event, 11–13 August 2011.
19. Bittau, A.; Hamburg, M.; Handley, M.; Boneh, D. The case for ubiquitous transport-level encryption. In Proceedings of the 19th USENIX Security Symposium (USENIX Security 10), Washington, DC, USA, 9 August 2010.
20. Nikolidakis, S.A.; Giotsas, V.; Georgakakis, E.; Vergados, D.D. Towards Utilizing Tcpcrypt in Mobile Healthcare Applications. In Proceedings of the Wireless Mobile Communication and Healthcare, Paris, France, 21–23 November 2012; Springer: Berlin/Heidelberg, Germany, 2012.
21. Bhushan, B.; Sahoo, G.; Rai, A.K. Man-in-the-middle attack in wireless and computer networking — A review. In Proceedings of the 2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA) (Fall), Dehradun, Indian, 15–16 September 2017. [CrossRef]
22. Odom, W. *Ccna 200-301 Official Cert Guide*, 1st ed.; Pearson Education, Inc.: Hoboken, NJ, USA, 2019; Volume 2.
23. Motero, C.; Higuera, J.R.; Bermejo, J.; Montalvo, J.A.; Gomez, N. On Attacking Kerberos Authentication Protocol in Windows Active Directory Services: A Practical Survey. *IEEE Access* **2021**, *9*, 109289–109319. [CrossRef]
24. Deland-Han. Kerberos Authentication Troubleshooting Guidance—Windows Server. Available online: https://learn.microsoft.com/en-us/troubleshoot/windows-server/windows-security/kerberos-authentication-troubleshooting-guidance (accessed on 22 April 2024).
25. SerdarSoysal. Authentication Overview for SharePoint Server—SharePoint Server. Available online: https://learn.microsoft.com/en-us/sharepoint/security-for-sharepoint-server/authentication-overview (accessed on 22 April 2024).
26. Han, J.; Kim, S.; Kim, T.; Han, D. Toward Scaling Hardware Security Module for Emerging Cloud Services. In Proceedings of the 4th Workshop on System Software for Trusted Execution, Huntsville, ON, Canada, 27–30 October 2019. [CrossRef]
27. Hupp, W.; Hasandka, A.; de Carvalho, R.S.; Saleem, D. Module-OT: A Hardware Security Module for Operational Technology. In Proceedings of the 2020 IEEE Texas Power and Energy Conference (TPEC), College Station, TX, USA, 6–7 February 2020. [CrossRef]
28. Vereecke, A. Koninklijke Militaire School: Amazon-KMS. Available online: https://aws.amazon.com/kms/ (accessed on 22 April 2024).
29. NIST. CVE-2020-8897 Detail. Available online: https://nvd.nist.gov/vuln/detail/CVE-2020-8897 (accessed on 22 April 2024).
30. Group, T. Post-Quantum Cryptography Solutions. Available online: https://cpl.thalesgroup.com/encryption/post-quantum-crypto-agility (accessed on 22 April 2024).
31. Crypto4A. Available online: https://crypto4a.com/news/quantum-safe-secure-manufacturing/ (accessed on 22 April 2024).
32. OpenDNSSEC. SoftHSM. Available online: https://www.opendnssec.org/softhsm/ (accessed on 22 April 2024).
33. Tomita, C.; Takita, M.; Fukushima, K.; Nakano, Y.; Shiraishi, Y.; Morii, M. Extracting the Secrets of OpenSSL with RAMBleed. *Sensors* **2022**, *22*, 3586. [CrossRef] [PubMed]
34. Walden, J. The Impact of a Major Security Event on an Open Source Project: The Case of OpenSSL. Available online: http://arxiv.org/abs/2005.14242 (accessed on 22 April 2024).
35. NPM. "loadtest" Linux Tool. Available online: https://www.npmjs.com/package/loadtest (accessed on 22 April 2024).

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

future internet

MDPI

Article

A Blockchain-Based Strategy for Certifying Timestamps in a Distributed Healthcare Emergency Response Systems

Daniele Marletta, Alessandro Midolo and Emiliano Tramontana *

Dipartimento di Matematica e Informatica, University of Catania, 95125 Catania, Italy;
daniele.marletta@phd.unict.it (D.M.); alessandro.midolo@phd.unict.it (A.M.)
* Correspondence: tramontana@dmi.unict.it

Abstract: A high level of data integrity is a strong requirement in systems where the life of people depends on accurate and timely responses. In healthcare emergency response systems, a centralized authority that handles data related to occurring events is prone to challenges, such as, e.g., disputes over event timestamps and data authenticity. To address both the potential lack of trust among collaborating parties and the inability of an authority to clearly certify events by itself, this paper proposes a blockchain-based framework designed to provide proof of integrity and authenticity of data in healthcare emergency response systems. The proposed solution integrates blockchain technology to certify the accuracy of events throughout their incident lifecycle. Critical events are timestamped and hashed using SHA-256; then, such hashes are stored immutably on an EVM-compatible blockchain via smart contracts. The system combines blockchain technology with cloud storage to ensure scalability, security, and transparency. Blockchain technology provides the advantage of eliminating a trusted server, providing timestamping and reducing costs by forgoing such a service. The experimental results, using publicly available incident data, demonstrated the feasibility and effectiveness of this approach. The system provides a cost-effective, scalable solution for managing incident data while keeping a proof of their integrity. The proposed blockchain-based framework offers a reliable, transparent mechanism for certifying incident-related data. This fosters trust among healthcare emergency response system actors.

Keywords: event certification; immutability; data integrity; smart contracts; proof of existence; timestamping

1. Introduction

In healthcare emergency response systems, the set of activities to be performed must be promptly recorded for potential further investigations. In such cases, external authorities can challenge healthcare service providers regarding the timing of critical events, such as the arrival time of an ambulance at the designated location, the starting time of a required analysis, etc. Generally, the activities to be performed follow a well-established practice and, thus, they are not at the discretion of the intervening personnel; however, the timing of such activities can be subject to scrutiny. Disagreements about the proper timing of some events can escalate into disputes, and the healthcare service provider should demonstrate that the appropriate sequence of actions was implemented at the correct time.

A general practice is that the healthcare provider records events and their timestamps, as they occur, on a dedicated server. However, the integrity of such timestamps may be

questioned when the authority responsible for recording the data is involved among the disputing parties. e.g., the challenging authority could argue that the data were tampered with in certain circumstances, making it appear that the ambulance arrived on time. In such cases, the defendant would not be able to demonstrate otherwise.

Although a centralized database would be under the control of some authority, when no authority can be considered trustworthy, blockchain technology is apt to ensure data integrity and access control, thanks to its mechanisms based on cryptographic signatures and hashes that confirm that the recorded data remain unmodifiable [1].

Several blockchain applications have been proposed for healthcare systems, including the management of electronic health records (EHR) and patient data [2–4], emergency medical services [5,6], patient-centric data sharing [7,8], integration with IoT devices [9], clinical research [10,11], healthcare security and audit processes [12–15], and medical data sharing [15–17]. Certainly, patient confidentiality and secure data exchanges should be ensured [18]. Unlike methods that depend on trusted third parties or centralized timestamping servers, our framework leverages a blockchain model to eliminate such dependencies. In our solution, blockchain-stored hashes either provide a certification of data kept unaltered in the database or make alteration evident and easily detected. Previous blockchain-based timestamping approaches have not solved issues due to transaction delays, high costs, and privacy. Our dual-layered design, which combines real-time cloud storage with blockchain recording, resolves these issues by ensuring both efficiency and robust security even under high data loads. Importantly, our approach uniquely ensures non-repudiation of recorded events, a critical capability in healthcare incident response that, to the best of our knowledge, has not been addressed by previous solutions.

We propose a blockchain-based framework designed to certify the integrity and authenticity of data and their timestamps for events occurring in healthcare emergency response systems. The proposed system includes a health emergency aggregator (HEA), a software component that collects incident data, records such timestamped data on a cloud server, and hashes them using SHA-256, as soon as they occur. Then, data certification is ensured by storing hashes on an EVM-compatible blockchain via smart contracts. This dual-layered approach, which combines blockchain and cloud storage, is known as off-chain data storage, a well-known technique that provides scalability, security, and transparency, while reducing the costs related to the blockchain [19–21]. Later, the authenticity of recorded events can be checked by comparing the hashes computed on the data stored on a cloud server with those stored on the blockchain [22].

To assess the practicality of our solution, we performed a comprehensive evaluation of our prototype software system using a substantial data set extracted from the incident dispatch dataset provided by the NYC OpenData initiative [23]. Our findings demonstrate the efficacy of our approach, making it a cost-effective strategy for aggregating the numerous data points and timestamps that would otherwise necessitate more recordings per day.

The main contribution of this paper is the development of a decentralized framework that leverages blockchain technology to certify the integrity and authenticity of incident data. By employing a client-server and blockchain-based software architecture, our framework guarantees that all recorded events become certified; thus, it holds a proof of the recorded timing. This solution facilitates secure and transparent data handling and enhances trust among actors. Additionally, we performed a thorough cost assessment to demonstrate the framework's economic viability, highlighting its cost-effectiveness in real-world scenarios. This combination of features provides emergency system authorities and patients with a trusted third-party mechanism to certify incident-related data efficiently and effectively.

The remainder of the paper is structured as follows. Section 2 reviews the most relevant approaches in the literature and compares them with our proposed method. Section 3 describes the architecture and methodology for preserving data integrity using blockchain. Section 4 presents experimental results from real-world simulations. Section 5 discusses the results and limitations of our approach. Finally, Section 6 draws our conclusion.

2. Related Works

Several approaches have been proposed to ensure the integrity and authenticity of long-term time-stamping (LTTS) systems, which leverage cryptographic techniques, trusted third parties, and decentralized architectures [24]. While some focus on formal security models, others explore blockchain-based solutions to eliminate central authorities.

Meng et al. [25] introduced a formal security model for LTTS schemes based on Message Authentication Codes (MACs), archives, and transient keys, addressing the lack of cryptographic analysis in alternative approaches. Despite enhancing robustness against cryptographic advancements, their solution depends on trusted third parties (e.g., archives and TSAs), incurs storage overhead, and requires frequent cryptographic updates. Similarly, Vigil et al. [26] explored decentralized timestamping for long-term digital archives but still relied on timestamping servers, making their approach vulnerable to centralization risks. Unlike these methods, our approach leverages blockchain to eliminate the need for trusted third parties while ensuring verifiable timestamps and built-in dispute resolution.

Blockchain-based timestamping has gained traction as an alternative to centralized solutions. Meng and Chen [27] proposed a scheme that records hash digests on a private blockchain, ensuring long-term verifiability without relying on external authorities. However, a private blockchain cannot provide guarantees as a public one, and the transaction delays introduced by the consensus mechanism and the risk of loss of keys affecting verification pose significant challenges. Wilson [28] extended this concept by implementing a permissioned blockchain for secure document timestamping, integrating off-chain storage and a quarantine mechanism. Although document management was enhanced, centralized control over the blockchain raises concerns regarding governance. Similarly, Zhang et al. [29] leveraged Ethereum blockchain to leverage transactions for timestamping, benefitting from its immutability but introducing privacy risks and high operational costs. To address these limitations, our approach adopts a blockchain model that stores only data hashes, thus eliminating privacy risks and significantly reducing operational costs.

Scalability and computational efficiency remain key concerns in long-term trusted timestamping (LTTS). Bin et al. [30] introduced IoETTS, a decentralized scheme for the Internet of Energy (IoE), leveraging public blockchain to improve data integrity. However, the computational and energy overhead associated with blockchain operations, particularly those relying on Proof of Work consensus, conflicts with the efficiency goals of IoE. Similarly, Zhang et al. [31] proposed a scalable LTTS scheme using commitment signatures and bilinear pairing accumulators to reduce transaction costs, but their approach lacks analysis of real-world deployment.

In addition to system implementations, theoretical analyses of LTTS security and efficiency have emerged. Meyer [32] formalized timestamping mechanisms using transitive prefix authentication graphs, providing a structured comparison of efficiency and security trade-offs. However, maintaining authenticated prefix structures introduces storage and computational overhead, limiting large-scale applicability. Meng et al. [33] examined client-side security risks, identifying vulnerabilities in key management and cryptographic resilience, but did not offer concrete mitigation strategies. Existing solutions often involve trade-offs between security, decentralization, scalability, and efficiency. In contrast, our ap-

proach leverages a public blockchain (Ethereum) to ensure decentralization and scalability, while maintaining the integrity of the stored hash and reducing operational costs.

Blockchain technology has been used in several contexts where it is fundamental that data remain unchanged, tamper-proof, and provable [34–36]. In the healthcare domain, the need for advanced technologies is rapidly increasing, with blockchain playing a key role in the transformation of the sector [37–39].

Odeh et al. [3] provided an overview of blockchain-based applications in healthcare, focusing on its ability to improve electronic health records (EHR), enhance data security, and support patient monitoring and drug traceability. They emphasized blockchain ability to ensure that only authorized parties can access sensitive medical data. Le et al. [4] proposed a permissioned blockchain system that uses the Hyperledger Fabric platform to allow quick and secure access to health records during emergencies. The system enables patients to define access rules for their data, which are enforced through smart contracts.

Suthaputchakun et al. [5] explored blockchain's potential to improve communication and security in emergency rescue operations, particularly in ambulance-to-everything communications. Their system provides secure data exchange between ambulances, hospitals, and other emergency services, preventing tampering and ensuring authenticity. Ksibi et al. [6] designed a blockchain-based system that integrates the Internet of Vehicles and the Internet of Medical Things to enhance emergency medical responses in smart cities, allowing the real-time transmission of accident victim data to nearby healthcare services.

Yue et al. [7] introduced a public blockchain architecture that empowers patients to control their healthcare data while ensuring privacy and enabling secure sharing among different healthcare providers. This decentralized approach improves data access and prevents unauthorized access and tampering. Griggs et al. [9] proposed a system that integrates a private blockchain with IoT medical devices, using smart contracts to automate notifications for medical interventions and ensure HIPAA-compliant data handling.

Lin et al. [13] presented a framework designed to address security concerns by ensuring secure mutual authentication and fine-grained access control. This framework leverages blockchain's capabilities to provide confidentiality, auditability, and scalability, making it suitable for dynamic environments such as smart factories. Fernandez-Aleman et al. [12] introduced a blockchain-based system for managing audit logs among healthcare organizations, enabling interoperability and ensuring the integrity of log data. Nugent et al. [11] leveraged blockchain's tamper-resistant features to secure clinical trial data, using Ethereum smart contracts to automate updates and ensure data transparency using a permissioned blockchain.

Benchoufi et al. [10] discussed how blockchain can address challenges in clinical research, such as data reproducibility, sharing, privacy, and patient enrollment. They emphasize blockchain's decentralized, transparent, and tamper-resistant nature, which is crucial for maintaining data integrity in clinical trials. Zheng et al. [8] propose a hybrid blockchain-cloud system for secure sharing of personal health data, using blockchain for transaction management and validation, while cloud storage handles large datasets.

Shi et al. [14] reviewed how blockchain technology was used to enhance the security and privacy of electronic health record (EHR) systems. They highlighted critical limitations of traditional EHR systems, such as data breaches, unauthorized access, and lack of transparency, and proposed that blockchain decentralization, immutability, and access control capabilities address these challenges. Practical use cases include secure patient data sharing, tamper-proof audit trails, and privacy enforcement through smart contracts. However, the study acknowledges significant barriers to adoption, including scalability, high computational costs, integration complexity, and regulatory constraints.

Xia et al. [16] proposed MeDShare, a blockchain-based system designed to facilitate medical data sharing in a trust-less environment. MeDShare ensures data provenance, auditing, and control for shared medical data in cloud repositories among big data entities. The system monitors entities accessing data for malicious use and records all data transitions and sharing activities in a tamper-proof manner. It employs smart contracts and an access control mechanism to effectively track data behavior and revoke access to offending entities upon detecting permission violations.

Similarly, Xia et al. [15] addressed the challenges of securely sharing electronic medical records (EMRs) in cloud environments, particularly concerning patient privacy and data breaches. The authors proposed a permissioned blockchain-based data-sharing framework that leverages the immutability and autonomy properties of blockchain technology to improved access control for sensitive data stored in the cloud. Their system utilizes a permissioned blockchain, granting access exclusively to invited and verified users, thus ensuring accountability by maintaining a log of all user actions.

Azaria et al. [17] introduced a private blockchain-based system designed to streamline access management and permissions for medical data. It addresses the fragmentation and inaccessibility of medical records by enabling patients to retain control over their data while providing secure, auditable access to healthcare providers. However, MedRec faces several challenges, including scalability limitations, high transaction costs, privacy risks from metadata exposure, and the complexity of patient and provider adoption. Additionally, regulatory compliance, energy consumption due to Ethereum's Proof of Work, and integration with existing EHR systems pose significant barriers to real-world deployment.

Patientory [40] is a blockchain-based peer-to-peer network for securely storing and sharing electronic medical records (EMRs). It integrates with existing EHR systems, allowing patients and healthcare providers to access medical data while ensuring security and compliance through smart contracts and the token economy. However, the approach faces several challenges, including scalability limitations, high transaction costs, privacy risks from off-chain storage, and regulatory compliance issues, particularly with data immutability.

Previous studies, reported above, have explored a wide range of blockchain-based applications in healthcare, primarily focusing on authentication, data privacy, access control, and data integrity. However, privacy risks and scalability limitations remain significant challenges, restricting the real-world applicability of these approaches. In contrast, our approach innovatively leverages blockchain technology to ensure the authenticity and integrity of event timestamps by storing only their hashes. This design choice significantly reduces blockchain costs while enhancing system maintainability, addressing a key limitation of existing research, where escalating costs hinder scalability [21]. Furthermore, our framework introduces event filtering mechanisms, allowing the selection and storage of only relevant events to optimize resource usage. It also incorporates data aggregation techniques to efficiently group multiple event records, further minimizing blockchain transaction costs. To validate its practicality, we conducted extensive experiments on real-world datasets, analyzing both transaction costs and the time required to securely store hashes on the blockchain. These strategies ensure that our solution is effective, cost-efficient, and feasible for the targeted medical domain.

3. System Architecture of the Decentralized Framework

3.1. Overview of the Main Components

Current limitations in healthcare emergency response systems highlight a critical need for effective solutions to certify the authenticity and integrity of events throughout the

incident lifecycle. To address this shortcoming, we propose an innovative architecture that integrates blockchain technology to have a secure and transparent mechanism for event certification in emergency response systems. This architecture ensures that critical data related to each incident are certified, fostering trust and accountability in the system. Figure 1 presents a high-level overview of the proposed architecture, outlining the key actors involved and the technologies employed.

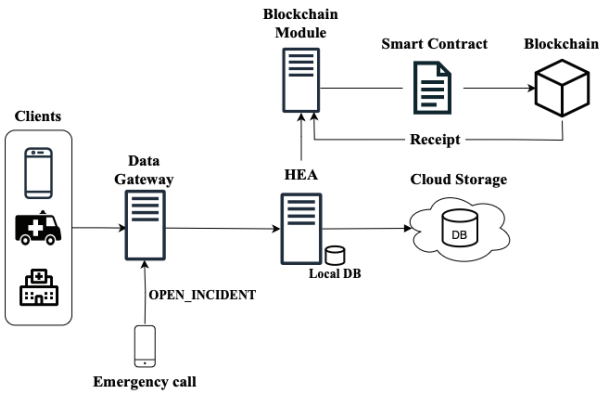


Figure 1. Overview of the proposed architecture, featuring the HEA, data gateway, blockchain integration for secure incident data management, and cloud storage for persistent data storage.

At the heart of the architecture is the health emergency aggregator (HEA), the main component responsible for managing incidents and triggering the activities necessary to safeguard data integrity. The lifecycle of an incident begins with the emergency call, which yields to the creation of a new incident record. This call goes to the data gateway, a component that acts as a facade and handles all the data from the clients and filters and forwards all the incident data to the HEA, which will store them in a local database. The HEA collects all the data related to an incident until it is resolved and can be archived in the cloud for long-term storage. The HEA functions as a reliable storage system, ensuring that all incident-related data are securely held without undergoing further processing or analysis. This capability provides an accurate historical record of emergency events.

To further enhance the security, a mechanism that can demonstrate the authenticity of incident data has been put into place. Critical events identified by the HEA are submitted to the blockchain module, where they are permanently recorded. This blockchain integration allows checking whether stored data were kept unaltered and renders any tampering activities worthless, since it would be easily exposed in case of disputes. The benefit of this approach is twofold: The data are protected from potential tampering, and a transparent audit trail is kept for future reference. Timestamped events are periodically sent to the blockchain while they occur, whereas the analysis of data is performed when needed, much after the incident has been handled, e.g., when further analyses are started to gather statistics on the healthcare service, or to check whether proper activities were performed in due time.

The workflow followed by the HEA is presented in Algorithms 1 and 2, which outline a systematic data handling and aggregation. Algorithm 1 shows that payloads received from a data gateway and categorized into one of four types—open, close, ordinary, and critical—trigger specific operations to manage incidents and their associated data. For open payloads, a new incident record is created, having a unique identifier, and then stored. For closed payloads, data related to a specific incident are retrieved and aggregated; then, the results of their management are appended to a list of ordinary data. When

processing ordinary and critical payloads, received data are stored; for critical cases, data are immediately aggregated and the results added to a dedicated critical data list.

Algorithm 1 Healthcare Emergency Aggregator (HEA) workflow when receiving data

```
ordinary_data ← []
critical_data ← []
while isListening do
    data_payload ← receiveFromDataGateway()
    switch (data_payload.type)
        case open:
            incident_id ← create_incident(data_payload)
            sendToDataGateway(incident_id)
            store(incident_id, data_payload)
        break
        case close:
            incident_id ← data_payload.incident_id
            incident_data ← getIncidentData(incident_id)
            agg_data ← aggregateData(incident_data, data_payload)
            ordinary_data.append(agg_data)
        break
        case ordinary:
            incident_id ← data_payload.incident_id
            store(indicent_id, data_payload)
        break
        case critical:
            incident_id ← data_payload.incident_id
            store(indicent_id, data_payload)
            agg_data ← aggregateData(incident_data, data_payload)
            critical_data.append(agg_data)
        break
end while
```

Algorithm 2 shows a timeout mechanism allowing us to periodically aggregate and secure both ordinary and critical data. Hashes of aggregated data are sent to a blockchain to ensure integrity and traceability, with transaction receipts stored for verification purposes. This workflow ensures efficient data management, secure storage, and streamlined incident handling in dynamic and high-stakes healthcare environments.

Algorithm 2 Healthcare Emergency Aggregator (HEA) workflow when timeouts occur

```
while isUp do
    switch (timeout)
        case critical:
            agg_data ← aggregateData(critical_data)
            trans_receipt ← sendToBlockchain(agg_data)
            store(critical_data, trans_receipt)
            critical_data ← []
        break
        case ordinary:
            agg_data ← aggregateData(ordinary_data)
            trans_receipt ← sendToBlockchain(agg_data)
            store(ordinary_data, trans_receipt)
            ordinary_data ← []
        break
end while
```

3.2. Healthcare Emergency Aggregator

The healthcare emergency aggregator (HEA) component serves as a central hub for managing all incidents and their associated data, ensuring seamless coordination and integration within the emergency response system. For each incident, the aggregator receives comprehensive data from the data gateway, which initiates the incident record. This record holds the time of the occurrence, the situational context, and detailed information about the stakeholders assigned to the incident. Such stakeholders are ambulances, medical

professionals, such as doctors and paramedics, healthcare facilities, such as hospitals, and other emergency response units.

The data gateway acts as a component that abstracts system complexity by providing a unified interface. Clients interact exclusively with the gateway, which consolidates incident data and forwards them to HEA for processing and storage. Centralizing incident data through HEA facilitates efficient resource allocation and real-time decision making. This approach is critical for prompt emergency responses and maintaining a comprehensive repository for post-incident analysis and system optimization. Emergency assistance requests typically begin when a patient contacts the designated emergency telephone number. Upon receiving a call, the operator collects relevant information about the patient and their situation. These data is used to create a standardized incident record in the system, which HEA manages. Each incident is assigned a unique identifier to ensure seamless tracking and management throughout its lifecycle.

Figure 2 presents a sequence diagram to show the interactions among the proposed components throughout the lifecycle of an incident. By showing the interactions among components of our proposed solution, we provide details of the communication handling (requests and answers), data filtering (separating ordinary and critical events), data transformation (aggregation and computing hashes) and permanent storage of data (hashes on the blockchain and data on the cloud server).

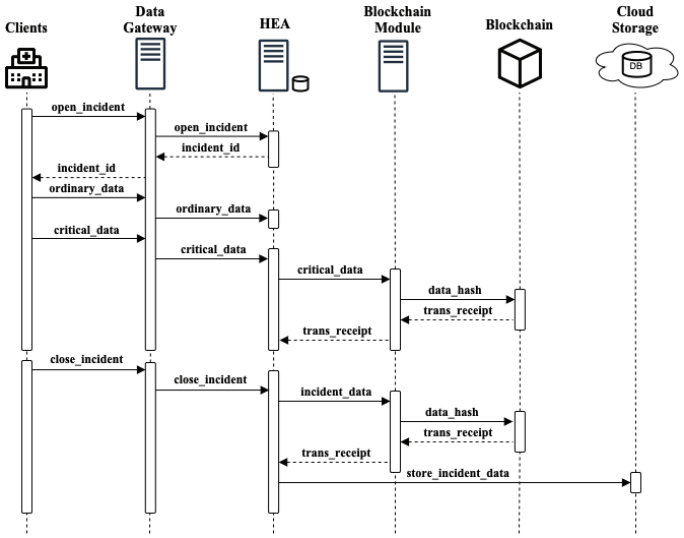


Figure 2. The lifecycle of an incident highlights interactions among components and the HEA's role logging critical events into the blockchain and storing incident data in the cloud servers.

The lifecycle begins with the formal creation of the incident (open_incident), initiated by a starting event that triggers the sequence of actions required for resolution. HEA assigns a unique identifier (incident_id) to each incident and communicates this ID to clients, enabling consistent reference during subsequent data transmissions related to the incident. As the incident unfolds, updates on status changes, resource allocation, and priority adjustments are recorded, along with communication traces arising from coordination within emergency response teams. The lifecycle ends with a final event (close_incident) marking the resolution or closure of the incident. This process provides a comprehensive chronological record of the incident's progression, offering valuable insights for accountability and continuous improvement.

The data collected during the lifecycle are categorized into two priority levels: ordinary and critical. Ordinary data encompass ancillary information such as the geographical

dispatch area, postal code, and descriptive details of the situation. As these pieces of data provide contextual understanding, they do not directly influence real-time decision making.

Critical data, in contrast, consist of timestamped events that document pivotal moments in the incident's lifecycle, such as dispatch times, arrival times at the scene, and specific interventions performed. These data are essential for real-time coordination and post-incident analysis. To ensure operational efficiency, legal compliance, and process improvement, the integrity and accuracy of critical data must be maintained rigorously.

The HEA employs a multi-layered approach to incident data management that ensures both system responsiveness and data integrity. A local database serves as a temporary cache for incident data, enabling rapid access and updates during an incident's active phase. When an incident is closed, the HEA consolidates all related information, including timestamps, resource allocations, stakeholder actions, and blockchain transaction receipts, into a coherent record. These data are then transferred to remote cloud storage, which serves as the permanent repository for historical data analysis and reporting, offering scalability, redundancy, and accessibility.

To further enhance data security and trust, HEA uses blockchain technology to save hashes of incident data. Such hashes, being immutable, would prove whether the exhibited data were the original data, or, vice versa, if hashes do not match, that the data were altered. After receiving critical data, the information is stored in the local database and then passed to the blockchain module for subsequent aggregation and submission. Likewise, when an incident is closed, all associated data are aggregated and their corresponding hashes are submitted to the blockchain. In the event that a blockchain transaction does not receive confirmation, the affected data batch and its hash remain securely stored in the local database, and the HEA subsequently reissues the hash-writing request to the blockchain module. This mechanism leverages the inherent resilience of blockchain technology, ensuring that, despite any transient failures, the hash writing is ultimately and reliably completed. Once a blockchain transaction is completed, a receipt is generated and attached to the incident record to facilitate later verification.

This dual-layered storage strategy, combining cloud storage with blockchain solutions, balances scalability with security. It reinforces the reliability and integrity of the emergency response system, fostering stakeholder confidence and ensuring compliance with legal and regulatory requirements.

3.3. Blockchain Module

Blockchain has emerged as an innovative technology in secure and decentralized data management, offering properties that are particularly relevant to sensitive domains such as the healthcare sector. At its core, blockchain is a distributed ledger system that records transactions in a series of cryptographically linked blocks. These blocks are designed to be immutable, ensuring that data, once added, cannot be modified or deleted, thus providing a high level of security and trust. Ethereum and other Ethereum Virtual Machine (EVM)-compatible blockchains have further advanced this technology by enabling programmable smart contracts. These contracts operate as self-executing code triggered by predefined conditions, eliminating the need for intermediaries while ensuring transparency.

In ambulance dispatch and patient data management, blockchain provides a secure and reliable framework to validate the critical events, while safeguarding sensitive information. The proposed system uses an Ethereum Virtual Machine (EVM)-compatible blockchain to certify the records, kept in a cloud storage, of incident data.

For enhanced security, incident data are hashed using the SHA-256 algorithm, generating a unique digital fingerprint that guarantees its integrity. This hash is then immutably recorded on the blockchain, establishing a transparent and unalterable record of all system transactions. This mechanism secures data and facilitates independent verification, as the blockchain serves as a reliable audit trail that can be referenced to confirm that the recorded data have not been tampered with.

The proof of the existence of an event generated by the client (mobile application) is represented by the concatenated hash of three components: a description of the event (e.g., a patient requesting an ambulance), its timestamp, and an optional document associated with the event (e.g., a picture of wound, the ID cards of a patient). The resulting hash is stored on the blockchain to guarantee the proof of existence of the event while preserving its integrity and validity. Hash writings and retrievals are performed by the DataHashStorage smart contract deployed on an EVM-compatible blockchain. The contract provides two core functions: (i) storeDataHash(bytes32 _dataHash), and (ii) getBlockTimestampFromHash(bytes32 _dataHash).

The storeDataHash() function takes as input the hash sent by the blockchain module and records it on the blockchain using an internal mapping. This mapping associates each hash with the timestamp of the block in which it was recorded. The block timestamp, represented as a uint256 type in Solidity, is measured in seconds since the Unix epoch. The HEA retrieves the block timestamp from the transaction receipt and associates it with the corresponding data batch in its local database.

The getBlockTimestampFromHash() function confirms the presence of a specific hash in the blockchain storage. It takes the hash as input and uses it to query the internal mapping, retrieving the corresponding block timestamp if the hash is found. This timestamp serves as proof of the hash's existence on the blockchain, confirming the associated data's registration. From the block timestamp, the existence of the block and the hash writing transaction can be verified. Conversely, if the function returns the default value of zero, it indicates that the provided hash is not present in the blockchain storage, showing it has not been recorded.

Disputes regarding data accuracy may arise when external parties (e.g., patients) report timestamps for critical events (e.g., the arrival time of an ambulance) that differ from those recorded by the system. In these cases, blockchain-stored data hashes are employed to verify the integrity and accuracy of the timestamps. The process involves retrieving the complete incident record from remote cloud storage and computing two hashes: (i) the system-recorded data hash, and (ii) the hash of the data with the disputed timestamp substituted. The smart contract function getBlockTimestampFromHash() is then called to verify the existence of these hashes on the distributed ledger. A matching hash confirms the data's integrity and the correct timestamp, whereas a mismatch indicates invalid data, thereby resolving the dispute. Table 1 illustrates an example using SHA-256 hashes computed from incident data.

Table 1. Example of SHA-256 hashes computed from incident data for integrity verification.

Timestamp	Hash
10 July 2023 11:51:08 a.m.	*b*51*f*737*f*466338*acd*27*ded*1*f*217*b*2279*de*416545*c*7*e*1*e*5*e*5*b*446*ec*4*be*3*a*10*a*56
10 July 2023 11:52:31 a.m.	*b*2*e*98*ad*6*f*6*eb*8508*dd*6*a*14*cfa*704*bad*7*f*05*f*6*fb*2*f*0*e*39*c*3*a*9*c*72*a*933*b*3*de*2*b*89

In Table 1, the first row shows the hash computed from the system-recorded incident data (including its timestamp), while the second row displays the hash computed with

the external, disputed timestamp. The presence of only the first hash on the blockchain confirms the integrity of the stored data and the accuracy of its timestamp.

It is important to note that although the HEA manages incident data, our design incorporating a public blockchain provides independent external verification. In situations such as legal disputes or regulatory audits, external stakeholders require assurance that recorded event timestamps and data integrity have not been subject to unilateral modification. By utilizing a public blockchain rather than an Ethereum sidechain or a private blockchain under HEA's control, the audit trail remains decentralized, trustless, and immune to centralized tampering. The immutability of the blockchain guarantees that once a hash is recorded, it remains unalterable, establishing a robust record that is time-stamped.

Only storing the data's hash on-chain offers several advantages. The hash serves as an exact digital fingerprint of the original data, preserving its integrity and confidentiality. When data are retrieved from secure off-chain (cloud-based) storage, the system recomputes the hash and compares it with the corresponding blockchain entry, ensuring that the data remain unaltered. This process provides a cost-effective and scalable record of incident data. Moreover, because the hash is generated using a one-way cryptographic function, no sensitive information is revealed, thus ensuring compliance with privacy requirements. The blockchain's immutability further guarantees that once a hash is recorded, it cannot be modified, creating a reliable audit trail that verifies the data's existence at a specific time. Finally, only storing the hash, rather than the complete dataset, yields significant cost savings by avoiding the substantial expenses associated with on-chain storage of large data volumes.

To efficiently manage the storage of critical and ordinary data, the system employs two strategies: time-based and threshold-based hash storage. The time-based strategy ensures that blockchain updates occur at regular intervals, aggregating critical data every 10 min while appending ordinary data less frequently, such as every 60 min. This strategy provides predictability and consistency, ensuring that data are recorded regardless of fluctuations in event frequency. In contrast, the threshold-based strategy dynamically adapts the frequency of blockchain writings based on real-time data flow. A new hash is computed only when the number of accumulated critical records surpasses a predefined threshold, allowing this method to adaptively respond to variations in event occurrence. Unlike the time-based approach, which maintains a fixed schedule, the threshold-based method optimizes resource usage by writing data only when necessary, reducing suboptimal transaction requests.

The choice between these strategies depends on the specific system requirements: The time-based strategy is suitable when regular and predictable updates are required, while the threshold-based approach is more effective in scenarios where event frequency is highly variable and periodic writing may lead to inefficiencies. For enhanced flexibility and efficiency, the system can also adopt a hybrid strategy that dynamically combines time-based and threshold-based approaches. This combined strategy allows the system to remain cost-effective when varying the frequency of event occurrence, providing configurable parameters to specify the lower and upper bounds for event aggregation before writing. The experimental evaluation of these three strategies is provided in the following section.

4. Experiments and Results

To perform our experiments, we employed the Incident Dispatch Data publicly available on the NYC OpenData website [23]. Each entry in the dataset contains detailed information about a specific incident, spanning from the time it was opened in the dispatch

system to the moment it was marked as closed. At the time of writing, the dataset includes entries up until 3 October 2024.

To model critical data in our proposed approach, we extracted all dataset fields containing timestamps, which were generated and recorded by the dispatch system whenever significant events occurred. A complete list of these fields (7 out of 31 columns in the dataset), along with their descriptions, is provided in Table 2.

Table 2. Name and description of key fields in the incident dispatch dataset used to identify critical data within the proposed approach.

Field Name	Field Description
Incident_Datetime	Incident creation in the dispatch system
First_Assignment_Datetime	First unit is assigned to the incident
First_Activation_Datetime	First unit is enroute to the incident location
First_On_Scene_Datetime	First unit arrives at the incident location
First_To_Hosp_Datetime	First unit is enroute to the hospital
First_Hosp_Arrival_Datetime	First unit arrives at the hospital
Incident_Close_Datetime	Incident closes in the dispatch system

The dataset contains 24 additional fields that provide supplementary details about each incident, such as the borough name, dispatch location zip code, and police precinct code. These fields, categorized as ordinary data in our system, are collected by HEA until the corresponding incident is officially closed. Due to space constraints, Table 3 presents a representative subset of these fields, including example values to illustrate the data format. The complete list of fields and their descriptions is available in Appendix A and on the dataset's homepage [23]. The salient characteristics of such a dataset are in Appendix B.

Table 3. Representative subset of fields categorized as ordinary data in the incident dispatch dataset, with descriptions and example values. The full list of fields and descriptions is available on the dataset homepage [23].

Field Name	Description	Sample Entry 1	Sample Entry 2
CAD_ID	Unique incident identifier	50011297	50132272
Call_Type	Type of emergency call	INJURY	OTHER
Severity_Code	Call priority level	5	6
Disposition_Code	Final incident outcome	CANCEL	90
Borough	Incident borough	BRONX	QUEENS
Zipcode	Incident zip code	10467	11369
Police_Precinct	Incident police precinct	47	115
Standby_Indicator	Indicates if units were on standby	N	Y

HEA operated continuously on a central server, receiving and storing incoming incident data in its local database. The data gateway was deployed on a cloud node, ensuring robust client interaction and seamless forwarding of incident data to HEA. The blockchain module, operating on a separate node, processed the incident data by generating a hash and recording it on the blockchain according to a predefined storage strategy. Once the transaction was successfully written on the blockchain, a transaction receipt was generated and returned to HEA for logging it onto the local database. Subsequently, both the receipt and the corresponding incident data, along with the computed hash, were transmitted to the cloud database. In our experiments, we evaluated three storage strategies: (i) time-based hash storage, (ii) threshold-based hash storage, and (iii) combined hash storage strategy.

4.1. Time-Based Hash Storage

In the time-based strategy, the blockchain module selected data at fixed time intervals to form a transaction batch. We set a time interval of 10 min to select critical data, and a time interval of 60 min to select the complete record of closed incidents, which comprises all the ordinary data. Subsequently, the transaction batch was processed using the SHA-256 hash function, and the resulting data hash was submitted to the DataHashStorage smart contract for permanent storage on the blockchain. The contract was deployed on the Ethereum Sepolia testnet, and is accessible at the following address: $0x2a5A789beBeE29c8fc824d76a82D3aE787b6d1F5$. The contract code, along with the complete transaction history, is accessible on the corresponding Sepolia Etherscan page (https://sepolia.etherscan.io/address/0x2a5a789bebee29c8fc824d76a82d3ae787b6d1f5, (accessed on the 7 February 2025)).

The DataHashStorage contract implements the function storeDataHash(bytes32 _dataHash), which accepts the SHA-256 hash as input and stores it in an internal mapping. This mapping persistently associates each hash with the timestamp of the block in which it was included, consuming some portion of the storage space in the Sepolia network.

We ran extensive experiments to evaluate the transaction fees and the time required to write hashes to the blockchain. In total, we executed 2050 transactions on the Sepolia network, spanning different days of the week and different times of the day. The success rate for executing transactions was 100%. Figure 3 presents a histogram showing the distribution of block confirmation times.

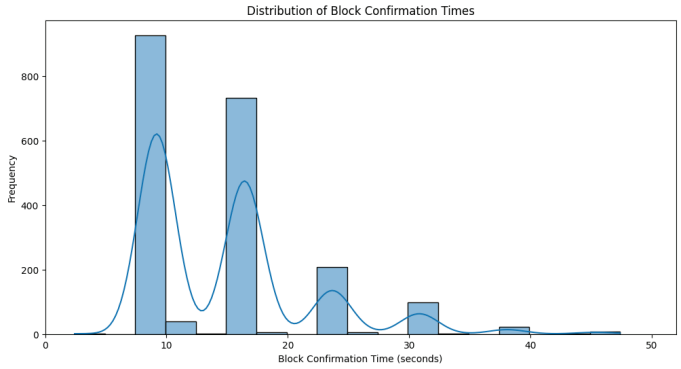


Figure 3. Histogram illustrating the distribution of block confirmation times, highlighting variability in transaction processing speeds within the blockchain network.

Data distribution analysis revealed that block confirmation times tend to cluster. To identify the time intervals corresponding to each cluster, we used the kernel density estimate (KDE) inferred from the data. Specifically, we extracted the relative minima of the multimodal KDE plot (see Figure 3) and observed the corresponding x-axis values as the boundaries for each time cluster. Table 4 presents reference metrics and count values for each cluster.

Table 4. Summary of cluster analysis metrics for block confirmation times.

Time Intervals (Seconds)	Mean	Median	Count	(%)
00.00–12.82	9.18	9.16	967	47.17%
12.82–20.50	16.43	16.39	739	36.05%
20.50–27.51	23.69	23.63	213	10.39%
27.51–35.19	30.85	30.78	99	4.83%
35.19–41.97	38.09	38.08	22	1.07%
41.97–50.00	45.45	45.49	10	0.49%
Overall	14.83	15.88	2050	100%

To assess the costs associated with hash writing, we analyzed the transaction receipts and calculated the fees in Gwei (1 Gwei $= 1 \times 10^{-9}$ ETH). The total fee was determined by multiplying the gas price at the time of block creation by the gas units consumed. In our experiments, we set a gas limit to 44,500 units and a maximum priority fee of 0.001 Gwei per transaction. To convert fees from Gwei to USD, we used CoinGecko, a publicly accessible cryptocurrency data aggregator that provides real-time exchange rates, including ETH-USD.

To evaluate the effect of the time of day on block confirmation times, we grouped transactions into three equally sized timespans, each spanning eight hours. Table 5 reports the metrics for each transaction group. Figure 4 shows the results of this analysis as a plot of the distribution of transaction fees in Gwei and USD. Figure 5 presents boxplots of transaction execution times, while Figure 6 displays boxplots of transaction fees.

Table 5. Comparison of key transaction metrics across different timespans.

Timespan	Mean Time	Median Time	Mean USD	Median USD	Count
00:00–08:00	16.59	16.23	0.21	0.17	677
08:00–16:00	13.29	9.56	0.41	0.29	707
16:00–00:00	14.69	16.01	0.22	0.17	666
Overall	14.83	15.88	0.29	0.19	2050

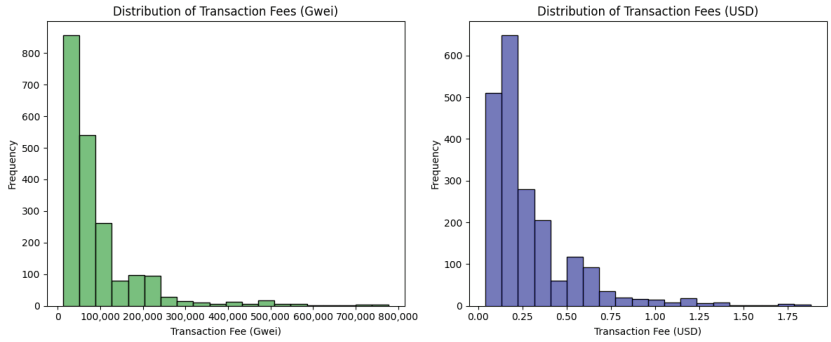


Figure 4. Comparative distribution of transaction fees, presented in both Gwei (**left**) and USD (**right**), illustrating cost variability in blockchain-based data recording.

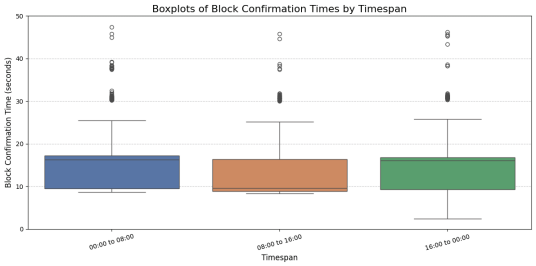


Figure 5. Boxplots representing the distribution of block confirmation times across timespans.

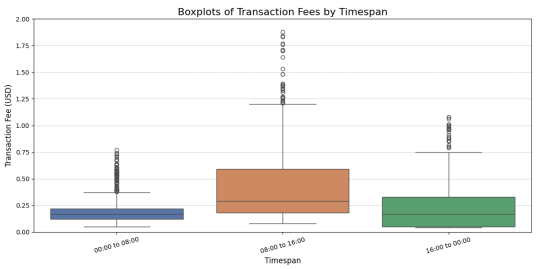


Figure 6. Boxplots illustrating transaction fee variations across different timespans.

4.2. Threshold-Based Hash Storage

In the threshold-based strategy, the blockchain module selected critical data from its local database to compute a hash only when a predefined threshold for the number of critical data entries was reached. The handling of ordinary data remained unchanged: Every 60 min, complete records of closed incidents were included in the transaction batch.

To determine the frequency of critical events in a real-world scenario, we performed a comprehensive analysis of the Incident Dispatch Data dataset. The dataset was cleaned and filtered as follows: Girstly, records covering the span of one year, from 1 January 2023 to 31 December 2023, were extracted; subsequently, duplicates and incidents marked as cancelled were removed; and finally, entries lacking an incident conclusion timestamp were discarded. The resulting clean dataset consisted in 1,457,802 unique incident entries.

The analysis of the clean dataset revealed consistent daily incident counts throughout the year, with an average of 3994 incidents per day and a median of 4010. For an incident, a maximum of seven critical events were counted (see Table 2), though not all entries include every critical timestamp. e.g., when an incident concludes with patient treatment at the location, timestamps for First_To_Hosp_Datetime and First_Hosp_Arrival_Datetime were not recorded and the incident was subsequently marked as closed.

For our experiments, we selected 3 October 2023, the day with the highest recorded number of critical data entries. On this day, 4499 unique incidents were recorded, corresponding to a total of 28,119 critical timestamps. We evaluated the system's performance for writing data hashes while using four threshold values for critical data: 100, 150, 175, and 200. Figure 7 illustrates the hourly costs of writing transactions for each threshold, with the mean hourly cost of the time-based strategy included as a reference.

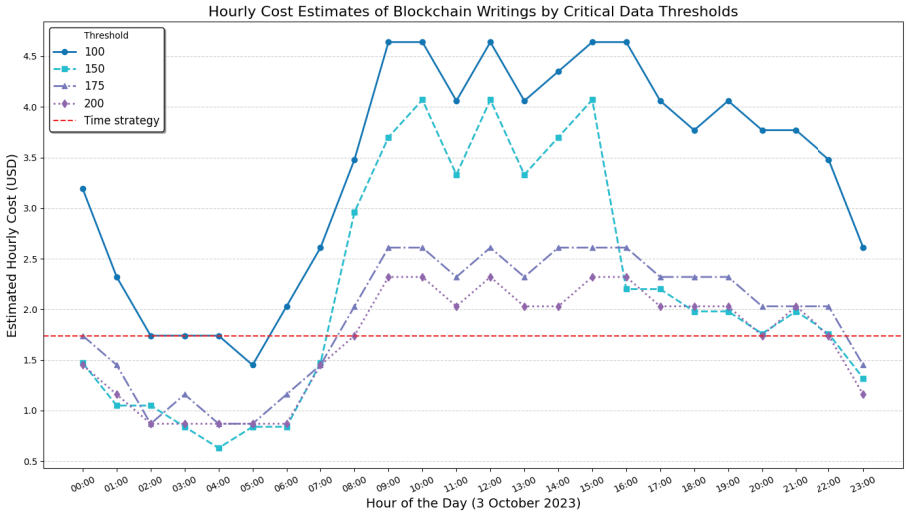


Figure 7. Analysis of cost variations based on different data threshold values on 3 October 2023, the day with the highest recorded volume of critical incident data.

Table 6 gives the total number of hash writing operations and the corresponding estimated costs. The estimates were based on a mean transaction fee of USD 0.29 (see Table 4). For comparison, the total values for the time-based strategy are also included at the bottom of the table.

Table 6. Blockchain hash writing operations and their associated costs for different thresholds, on 3 October 2023, the day with the highest recorded volume of critical incident data.

Threshold	Total Writings	Mean Cost (USD)	Mean Writings	Max Writings
100	281	81.49	11.7	16
150	187	54.23	7.8	11
175	160	46.4	6.67	9
200	140	40.60	5.83	8
Time strategy	144	41.76	6	6

4.3. Combined Hash Storage Strategy

To provide a more accurate cost assessment and demonstrate the system's flexibility, we extended our analysis over one month and developed a hash writing strategy that combines both time-based and threshold-based approaches. This strategy defines a minimum and maximum threshold, which allows for an adaptive storage mechanism. At fixed 10-min intervals, the system evaluates the volume of accumulated critical data. If the volume falls below the minimum threshold, hash computation and writing are skipped, reducing unnecessary transactions and optimizing cost efficiency. Conversely, if the maximum threshold is reached before the next scheduled interval, the system immediately performs a hash computation and writes the critical data to the blockchain. This ensures timely data aggregation while maintaining an efficient balance between cost and performance.

We conducted a detailed cost analysis of the combined strategy using various threshold pairs. The analysis was based on data from July 2023, the month with the highest recorded volume of critical data. In July 2023, 128,296 unique incidents were recorded, corresponding to a total of 799,008 critical timestamps. Figure 8 illustrates the estimated daily costs for four threshold configurations (125–175, 115–190, 135–195, and 145–205), compared to the time-based strategy.

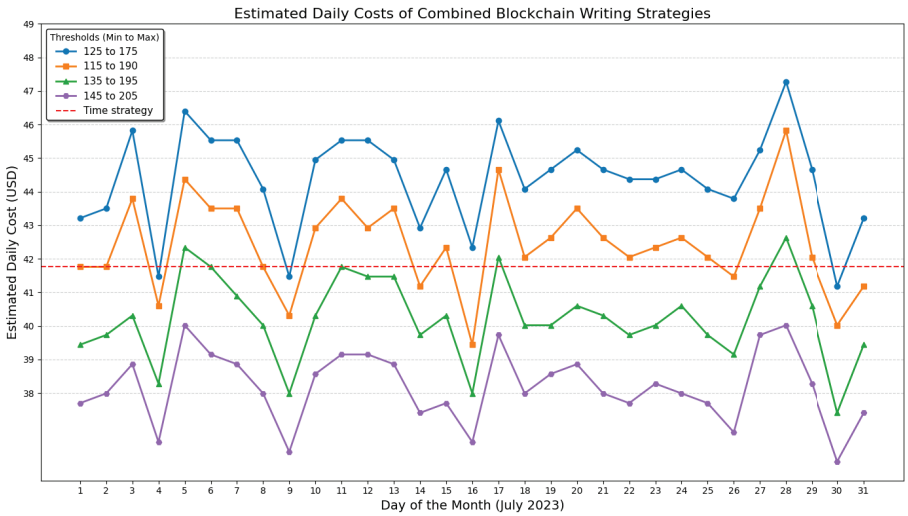


Figure 8. Cost analysis of combined blockchain data writing strategies over July 2023, the month with the highest critical data volume.

Table 7 summarizes the total number of hash-writing operations and the estimated costs associated with each threshold pair. These estimates are based on an average transaction fee of USD 0.29 (see Table 4). For comparison, the corresponding values for the time-based strategy are also provided.

Table 7. Hash writing operations and estimated costs for different pairs of threshold values in July 2023, the month with the highest recorded volume of critical data.

Thresholds	Mean Daily Cost (USD)	Mean Daily Writings	Max Daily Writings
125-175	44.37	153	163
115-190	42.45	146.4	158
135-195	40.23	138.7	147
145-205	38.19	131.7	138
Time strategy	41.76	144	144

5. Discussion

A total of 2050 hash writing requests were confirmed on the Ethereum Sepolia blockchain, spanning various days of a week and several time slots of a day. The time-based strategy executed 144 daily transactions, whereas the threshold-based strategy ranged from 140 to 281 daily writings, depending on threshold values (see Table 6). Consequently, the experiments covered over two weeks of operation for the time-based strategy and one to two weeks for the threshold-based strategy. This transaction volume, designed to assess the performance, timing, and cost efficiency of blockchain-based hash writing under realistic conditions, provides a sufficient basis for meaningful economic feasibility estimates.

The combined strategy was evaluated using comprehensive data spanning an entire month, ensuring a more extensive analysis of its adaptability and cost efficiency under varying operational conditions. This broader timeframe allowed us to assess long-term performance trends, capturing fluctuations in data volume and evaluating threshold configurations accordingly. Our experimental results indicate that nearly half (47.17%) of the total transactions were written on the blockchain in less than 13 s (see Table 4), with a mean confirmation time of 9.18 s and a median of 9.16 s. Moreover, the vast majority of transactions (93.61%) were confirmed within 30 s, with overall mean and median confirmation times of 14.83 s and 15.88 s, respectively.

The confirmation time boxplots in Figure 5 show that transactions were confirmed more rapidly between 08:00 and 16:00, with a mean time of 13.29 s and a median of 9.56 s (see Table 5). In contrast, slower confirmation times were observed during the 00:00–08:00 and 16:00–00:00 timespans, with median values of 16.23 s and 16.01 s, respectively. This represents an increase of approximately 67–70% compared to the 08:00–16:00 timespan. Furthermore, the integration of blockchain technology significantly augments the system by providing a verifiable timeline of events. By permanently recording event hashes on the blockchain, any change to events stored in the database would be detected; hence, a dispute on recorded timestamps would be resolved. A match of the hash computed on recorded timestamps with the blockchain hash would demonstrate that timestamps were unaltered, while a difference would show that they were altered. This is vital for accountability and legal verification in emergency response scenarios. Notably, in our experiments, all blockchain transactions were confirmed, underscoring the reliability of established platforms such as Ethereum for secure data storage. In cases where a transaction is not confirmed, the affected data remain securely stored in the local database and the hash writing request is reissued, leveraging the inherent resilience of blockchain technology to ensure reliable completion. Collectively, these features enhance data integrity and fault tolerance and provide an indisputable record of incident timelines.

In every strategy we tested, the data aggregation performed by the HEA prevents congestion by avoiding overly frequent writing requests on the blockchain. In the transaction fee distributions (Figure 4), we observed that 83.61% transactions incurred costs below USD 0.50. The difference between the Gwei and USD distributions is due to the actual fee in Gwei, which depends on the congestion of the blockchain network at the time of the transaction,

and the ETH/USD exchange rate, which is subject to continuous fluctuations. To handle price fluctuations, some strategies can be employed. e.g., a polling mechanism can be activated some minutes before the timeout to monitor the current gas cost and request a transaction whenever the cost falls below a set threshold. Depending on the system requirements, either the transaction cost or the timeout to send the transaction can be prioritized.

The analysis of the transaction fee boxplots in Figure 6 revealed that the 08:00–16:00 timespan was the most expensive, with a significant number of outliers: the median fee was USD 0.29 and the mean fee was USD 0.41 (see Table 5). Conversely, the 00:00–08:00 and 16:00–00:00 timespans exhibited lower and less variable fees, with median values of USD 0.17 and mean values of USD 0.21 and USD 0.22, respectively.

We estimated the total daily transaction cost using the mean transaction fee of USD 0.29 (see Table 5). In the time-based strategy, 144 daily transactions amounted to an average cost of USD 41.76. In the threshold-based strategy, daily costs ranged from USD 40.60 for a threshold of 200 to USD 81.49 for a threshold of 100 (see Table 6). In the combined strategy, estimated daily costs varied from USD 38.19 for the threshold pair of 145–205 to USD 44.37 for the threshold pair of 125–175 (see Table 7).

The effectiveness of the three strategies can be assessed through their impact on writing frequency and cost efficiency. The time-based strategy enforced a fixed update interval of one hash every 10 min, offering consistency but potentially incurring unnecessary transaction costs during low-activity periods. In contrast, the threshold-based strategy responded dynamically to data flow, increasing the frequency of writing operations when critical data accumulation exceeded predefined thresholds. Even in the worst case scenario (threshold = 100), the system maintained an average of 12 writings per hour, demonstrating a balance between responsiveness and cost (see Table 6).

The combined strategy leveraged both approaches to optimize cost-effectiveness while ensuring timely data recording. Notably, its daily writing frequency remained closely aligned with that of the time-based approach (see Table 7). The 125–175 threshold pair configuration resulted in 153 average daily writings, peaking at 163, reflecting only moderate increases of 6% and 13%, respectively, relative to the time-based model. Meanwhile, the 145–205 threshold pair maintained a lower transaction volume than the time-based approach, with a maximum of 138 daily writings, compared to 144 in the time-based strategy. This configuration also yielded a mean daily cost of USD 38.19, representing a daily operational cost reduction of USD 3.57 while retaining the combined strategy's capacity to dynamically adapt to fluctuating data loads.

Our approach significantly mitigates the high costs typically associated with on-chain storage of large data volumes. In our system, only a 32-byte hash is written to the blockchain; hence, recording a fixed-length hash incurs only a small fraction of the cost required for storing complete data records. The experimental results confirm that transaction fees for hash writing operations remain low even under variable gas prices. These findings highlight the inherent trade-offs between predictability, cost-efficiency, and adaptability. The time-based strategy ensures a consistent recording schedule but may result in redundant blockchain interactions. Conversely, the threshold-based approach offers greater adaptability but introduces variability in update timing. The combined strategy, particularly with the 145–205 configuration, demonstrates an optimal balance, achieving cost savings while maintaining responsiveness to fluctuations in critical data volume.

Compared to third-party and commercial timestamp certification systems, our approach leverages the inherent properties of a public blockchain to ensure that the critical function of proving data integrity remains within the system rather than being outsourced to external entities that may impose arbitrary conditions for service access and utilization. By recording data hashes on a public blockchain, the proof of integrity can be independently

verified by authorized entities in the event of disputes. This internal verification process enhances system trust and differentiates our approach from conventional timestamping methods, which rely on external certification providers.

We utilized a publicly available dataset of the New York metropolitan area, encompassing approximately eight million residents, and conducted experiments to derive average transaction cost values. These values, derived from analyzing multiple threshold configurations over various time intervals, offer a practical baseline for expense estimation. This baseline, supported by empirical data, allows for the projection of costs over different durations, thereby enabling more informed decisions regarding investment and operational expenses. The results demonstrate that our approach maintains feasible costs even amidst blockchain fee variability. By aggregating data and employing a threshold-triggered strategy, the system avoids unnecessary network congestion and minimizes operational expenses. Consequently, these cost estimates underscore the scalability and applicability of our system in densely populated, data-intensive environments.

6. Conclusions

This paper presented a blockchain-based framework designed to allow healthcare service providers to hold (and, when necessary, exhibit) a proof of the integrity and authenticity of timestamped events, in case of disputes. The designed system ensures accurate and blockchain-certified timestamps for critical events, addressing a major limitation of centralized architectures that depend on a single authority. This capability was achieved through blockchain technology, which provides a transparent and verifiable record of events. The proposed framework permanently records event hashes on the blockchain, making the timeline of incidents (stored in the cloud) verifiable. Therefore, the tampering of cloud-stored data becomes worthless, as it would be easily detected in case of disputes.

Our experimental evaluation, based on a real-world incident dataset, confirmed that the proposed framework is feasible and effective for the management of secure incidents. The integration of regular, time-based recording with an adaptive threshold strategy enables the system to record and secure critical event data reliably, ensuring that the incident timeline is maintained accurately. This balanced approach ensures that essential information is logged without excessive redundancy, thus reducing resource utilization and maintaining system efficiency.

The dynamic nature of the combined strategy allows the framework to adapt seamlessly to varying operational conditions, ensuring that the level of data capture aligns closely with the actual flow of incident information. Such a continuous adaptation ensures that the data are kept accurate and verifiable, which is essential in scenarios where swift response and strict accountability are paramount. In addition, the insights derived from our study highlight the broader applicability of blockchain technology in incident management systems. By combining server-side timestamping with blockchain-backed verification, the system guarantees the auditable history of emergency events, fostering trust among stakeholders such as emergency responders, healthcare providers, and patients.

Our future work will aim at proposing a solution enabling the exchange of anonymized data, gathered in emergency response systems, by using a blockchain-based storage that acts as an indexing system and a proof for data stored in a cloud-based system.

Author Contributions: Conceptualization, E.T.; methodology, D.M., A.M. and E.T.; software, D.M.; validation, D.M., A.M. and E.T.; writing—original draft preparation, D.M. and A.M.; writing—review and editing, E.T.; visualization, D.M. and A.M.; supervision, E.T.; project administration, E.T.; funding acquisition, E.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original data presented in the study are openly available in NYC OpenData at https://data.cityofnewyork.us/Public-Safety/EMS-Incident-Dispatch-Data/76xm-jjuj.

Acknowledgments: The authors acknowledge the support of the University of Catania PIACERI Project "TEAMS".

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

Appendix A. EMS Incident Dispatch Dataset Description

This appendix provides a comprehensive overview of the EMS Incident Dispatch Dataset utilized in this study. It details each field present in the dataset, including its data type, a representative sample entry, and a concise description explaining its meaning and purpose within the context of emergency medical service dispatch operations. The dataset and all related information are publicly accessible on the NYC OpenData website [23].

Table A1. Full list of fields in the incident dispatch dataset, including descriptions and example values. The dataset is publicly available on the NYC OpenData website [23].

Field Name	Data Type	Sample Entry	Field Description
CAD_Incident_Id	Text	50010001	An incident identifier comprising the Julian date and a 4-character sequence number starting at 1 each day
Incident_Datetime	Timestamp	1 Jan 2005 12:00:24 a.m.	The date and time the incident was created in the dispatch system
Initial_Call_Type	Text	INJURY	The call type assigned at the time of incident creation
Initial_Severity_Level_Code	Text	5	The segment (priority) assigned at the time of incident creation
Final_Call_Type	Text	INJURY	The call type at the time the incident closes
Final_Severity_Level_Code	Text	5	The segment (priority) assigned at the time the incident closes
First_Assignment_Datetime	Timestamp	1 Jan 2005 12:04:11 a.m.	The date and time the first unit is assigned
Valid_Dispatch_Rspns_Time_Indc	Text	Y	Indicates that the components comprising the calculation of the Dispatch_Response_Seconds_QY are valid
Dispatch_Response_Seconds_QY	Number	227	The time elapsed in seconds between the incident_datetime and the first_assignment_datetime
First_Activation_Datetime	Timestamp	1 Jan 2005 12:06:15 a.m.	The date and time the first unit gives the signal that it is enroute to the location of the incident
First_On_Scene_Datetime	Timestamp	1 Jan 2005 12:16:06 a.m.	The date and time the first unit signals that it has arrived at the location of the incident
Valid_Incident_Rspns_Time_Indc	Text	Y	Indicates that the components comprising the calculation of the Incident_Response_Seconds_QY are valid
Incident_Response_Seconds_QY	Number	942	The time elapsed in seconds between the incident_datetime and the first_on_scene_datetime
Incident_Travel_Tm_Seconds_QY	Number	715	The time elapsed in seconds between the first_assignment_datetime and the first_on_scene_datetime
First_To_Hosp_Datetime	Timestamp	1 Jan 2005 12:26:53 a.m.	The date and time the first unit gives the signal that it is en route to the hospital
First_Hosp_Arrival_Datetime	Timestamp	1 Jan 2005 12:39:48 a.m.	The date and time the first unit signals that it has arrived at the hospital
Incident_Close_Datetime	Timestamp	1 Jan 2005 1:04:36 a.m.	The date and time the incident closes in the dispatch system
Held_Indicator	Text	N	Indicates that, for some reason, a unit could not be assigned immediately
Incident_Disposition_Code	Text	82	A code indicating the final outcome of the incident. See incident dispositions
Borough	Text	BROOKLYN	The borough of the incident location
Incident_Dispatch_Area	Text	K6	The dispatch area of the incident
Zipcode	Text	11217	The zip code of the incident
Police_Precinct	Number	76	The police precinct of the incident
City_Council_District	Number	33	The city council district
Community_District	Number	306	The community district

Table A1. *Cont.*

Field Name	Data Type	Sample Entry	Field Description
Community_School_District	Number	15	The community school district
Congressional_District	Number	7	The congressional district
Reopen_Indicator	Text	N	Indicates that, at some point, the incident was closed but then reopened
Special_Event_Indicator	Text	N	Indicates that the incident was a special event such as the NYC Marathon
Standby_Indicator	Text	N	Indicates that the units were assigned to stand by in case they were needed
Transfer_Indicator	Text	N	Indicates that the incident was created for the transportation of a patient from one facility to another

Appendix B. EMS Incident Dispatch Dataset Characteristics

The used dataset consists of 31 fields (see Table A1 in Appendix A). For the number of instances, the day with highest critical data volume was 3 October 2023, having 4499 unique incidents; the month with the highest critical data volume was July 2023, having 128,296 unique incidents. For the size of the dataset, the day with the highest critical data volume (i.e., 3 October 2023) had approximately 1.1 MB, as a CSV-format file comprising the whole raw data; the month with the highest critical data volume (i.e., July 2023) had approximately 32 MB, as a CSV-format file. The hash size for each blockchain transaction is 32 bytes. The transaction request frequency varies according to the proposed writing strategy (time-based or threshold-based). Hash writing operation statistics are detailed in Tables 6 and 7.

References

1. Estevam, G.; Palma, L.M.; Silva, L.R.; Martina, J.E.; Vigil, M. Accurate and decentralized timestamping using smart contracts on the Ethereum blockchain. *Inf. Process. Manag.* **2021**, *58*, 102471. [CrossRef]
2. McGhin, T.; Choo, K.K.R.; Liu, C.Z.; He, D. Blockchain in healthcare applications: Research challenges and opportunities. *J. Netw. Comput. Appl.* **2019**, *135*, 62–75. [CrossRef]
3. Odeh, A.; Keshta, I.; Al-Haija, Q.A. Analysis of Blockchain in the Healthcare Sector: Application and Issues. *Symmetry* **2022**, *14*, 1760. [CrossRef]
4. Le, H.T.; Thanh, L.N.T.; Vo, H.K.; Luong, H.H.; Tuan, K.N.H.; Anh, T.D.; Nguyen, T.A.; Vuong, K.H.N.; Son, H.X. Patient-Chain: Patient-centered Healthcare System a Blockchain-based Technology in Dealing with Emergencies. In Proceedings of the Parallel and Distributed Computing, Applications and Technologies, Guangzhou, China, 17–19 December 2021; Springer: Cham, Switzerland, 2022; pp. 576–583.
5. Suthaputchakun, C.; Cao, Y. Blockchain-based Secure Ambulance-to-Everything Communications in Emergency Rescue Operations. In Proceedings of the International Conference on Ubiquitous and Future Networks (ICUFN), Barcelona, Spain, 5–8 July 2022; pp. 174–179. [CrossRef]
6. Ksibi, A.; Mhamdi, H.; Ayadi, M.; Almuqren, L.; Alqahtani, M.S.; Ansari, M.D.; Sharma, A.; Hedi, S. Secure and Fast Emergency Road Healthcare Service Based on Blockchain Technology for Smart Cities. *Sustainability* **2023**, *15*, 5748. [CrossRef]
7. Yue, X.; Wang, H.; Jin, D.; Li, M.; Jiang, W. Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control. *J. Med. Syst.* **2016**, *40*, 218. [CrossRef]
8. Zheng, X.; Mukkamala, R.R.; Vatrapu, R.; Ordieres-Mere, J. Blockchain-based Personal Health Data Sharing System Using Cloud Storage. In Proceedings of the IEEE International Conference on e-Health Networking, Applications and Services (Healthcom), Ostrava, Czech Republic, 17–20 September 2018; pp. 1–6. [CrossRef]
9. Griggs, K.N.; Ossipova, O.; Kohlios, C.P.; Baccarini, A.N.; Howson, E.A.; Hayajneh, T. Healthcare blockchain system using smart contracts for secure automated remote patient monitoring. *J. Med. Syst.* **2018**, *42*, 130. [CrossRef]
10. Benchoufi, M.; Ravaud, P. Blockchain technology for improving clinical research quality. *Trials* **2017**, *18*, 335. [CrossRef]
11. Nugent, T.; Upton, D.; Cimpoesu, M. Improving data transparency in clinical trials using blockchain smart contracts. *F1000Research* **2016**, *5*, 2541. [CrossRef]
12. Fernández-Alemán, J.L.; Señor, I.C.; Ángel Oliver Lozoya, P.; Toval, A. Security and privacy in electronic health records: A systematic literature review. *J. Biomed. Inform.* **2013**, *46*, 541–562. [CrossRef]

13. Lin, C.; He, D.; Huang, X.; Choo, K.K.R.; Vasilakos, A.V. BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0. *J. Netw. Comput. Appl.* **2018**, *116*, 42–52. [CrossRef]
14. Shi, S.; He, D.; Li, L.; Kumar, N.; Khan, M.K.; Choo, K.K.R. Applications of blockchain in ensuring the security and privacy of electronic health record systems: A survey. *Comput. Secur.* **2020**, *97*, 101966. [CrossRef]
15. Xia, Q.; Sifah, E.B.; Smahi, A.; Amofa, S.; Zhang, X. BBDS: Blockchain-Based Data Sharing for Electronic Medical Records in Cloud Environments. *Information* **2017**, *8*, 44. [CrossRef]
16. Xia, Q.; Sifah, E.B.; Asamoah, K.O.; Gao, J.; Du, X.; Guizani, M. MeDShare: Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain. *IEEE Access* **2017**, *5*, 14757–14767. [CrossRef]
17. Azaria, A.; Ekblaw, A.; Vieira, T.; Lippman, A. MedRec: Using Blockchain for Medical Data Access and Permission Management. In Proceedings of the International Conference on Open and Big Data (OBD), Vienna, Austria, 22–24 August 2016; pp. 25–30. [CrossRef]
18. Haleem, A.; Javaid, M.; Singh, R.P.; Suman, R.; Rab, S. Blockchain technology applications in healthcare: An overview. *Int. J. Intell. Netw.* **2021**, *2*, 130–139. [CrossRef]
19. Hepp, T.; Sharinghousen, M.; Ehret, P.; Schoenhals, A.; Gipp, B. On-chain vs. off-chain storage for supply- and blockchain integration. *It-Inf. Technol.* **2018**, *60*, 283–291. [CrossRef]
20. Xu, C.; Zhang, C.; Xu, J.; Pei, J. SlimChain: Scaling blockchain transactions through off-chain storage and parallel processing. *Proc. VLDB Endow.* **2021**, *14*, 2314–2326. [CrossRef]
21. Jayabalan, J.; Jeyanthi, N. Scalable blockchain model using off-chain IPFS storage for healthcare data security and privacy. *J. Parallel Distrib. Comput.* **2022**, *164*, 152–167. [CrossRef]
22. Iansiti, M.; Lakhani, K.R. The truth about blockchain. *Harv. Bus. Rev.* **2017**, *95*, 118–127.
23. EMS Incident Dispatch Data | NYC Open Data. Available online: https://data.cityofnewyork.us/Public-Safety/EMS-Incident-Dispatch-Data/76xm-jjuj (accessed on 7 November 2024).
24. Bayer, D.; Haber, S.; Stornetta, W.S. Improving the Efficiency and Reliability of Digital Time-Stamping. In *Proceedings of the Sequences II*; Capocelli, R., De Santis, A., Vaccaro, U., Eds.; Springer: New York, NY, USA, 1993; pp. 329–334.
25. Meng, L.; Chen, L. Long-Term Time-Stamping Schemes Based on MACs, Archives, and Transient Keys. In Proceedings of the 2024 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics, Copenhagen, Denmark, 19–22 August 2024; pp. 535–542. [CrossRef]
26. Vigil, M.; Weinert, C.; Demirel, D.; Buchmann, J. An efficient time-stamping solution for long-term digital archiving. In Proceedings of the 2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC), Austin, TX, USA, 5–7 December 2014; pp. 1–8. [CrossRef]
27. Meng, L.; Chen, L. A Blockchain-Based Long-Term Time-Stamping Scheme. In *Proceedings of the Computer Security—ESORICS 2022*; Atluri, V., Di Pietro, R., Jensen, C.D., Meng, W., Eds.; Springer: Cham, Switzerland, 2022; pp. 3–24.
28. Wilson, K.S. Blockchain with Daisy Chained Records, Document Corral, Quarantine, Message Timestamping, and Self-Addressing. US Patent 11,444,776, 13 September 2022.
29. Zhang, Y.; Xu, C.; Cheng, N.; Li, H.; Yang, H.; Shen, X. Chronos: An Accurate Blockchain-Based Time-Stamping Scheme for Cloud Storage. *IEEE Trans. Serv. Comput.* **2020**, *13*, 216–229. [CrossRef]
30. Qian, B.; Luo, Y.; Ou, J.; Xiao, Y.; Hu, H. IoETTS: A Decentralized Blockchain-based Trusted Time-stamping Scheme for Internet of Energy. *J. Internet Technol.* **2023**, *24*, 519–529.
31. Zhang, S.; Zhou, Y.; Wang, Z.; Huang, Y.; Zhang, C.; Wei, L. An Efficient Blockchain-Based Time-Stamping Scheme Using Commitment Signatures. In Proceedings of the IEEE Global Communications Conference, Rio de Janeiro, Brazil, 4–8 December 2022; pp. 2591–2596. [CrossRef]
32. Meyer, A. SoK: Authenticated Prefix Relations–A Unified Perspective On Relative Time-Stamping and Append-Only Logs. *arXiv* **2023**, arXiv:2308.13836. [CrossRef]
33. Meng, L.; Chen, L. Analysis of Client-Side Security for Long-Term Time-Stamping Services. In Proceedings of the Applied Cryptography and Network Security, Kamakura, Japan, 21–24 June 2021; Sako, K., Tippenhauer, N.O., Eds.; Springer: Cham, Switzerland, 2021; pp. 28–49.
34. Olawale, O.P.; Ebadinezhad, S. Cybersecurity Anomaly Detection: AI and Ethereum Blockchain for a Secure and Tamperproof IoHT Data Management. *IEEE Access* **2024**, *12*, 131605–131620. [CrossRef]
35. Sarker, A.; Wuthier, S.; Kim, J.; Kim, J.; Chang, S.Y. Blockchain Handshaking with Software Assurance: Version++ Protocol for Bitcoin Cryptocurrency. *Electronics* **2024**, *13*, 3857. [CrossRef]
36. Zhang, B.; Pan, H.; Li, K.; Xing, Y.; Wang, J.; Fan, D.; Zhang, W. A Blockchain and Zero Knowledge Proof Based Data Security Transaction Method in Distributed Computing. *Electronics* **2024**, *13*, 4260. [CrossRef]

37. Hölbl, M.; Kompara, M.; Kamišalić, A.; Nemec Zlatolas, L. A Systematic Review of the Use of Blockchain in Healthcare. *Symmetry* **2018**, *10*, 470. [CrossRef]
38. Tandon, A.; Dhir, A.; Islam, A.N.; Mäntymäki, M. Blockchain in healthcare: A systematic literature review, synthesizing framework and future research agenda. *Comput. Ind.* **2020**, *122*, 103290. [CrossRef]
39. Mandarino, V.; Pappalardo, G.; Tramontana, E. A Blockchain-Based Electronic Health Record (EHR) System for Edge Computing Enhancing Security and Cost Efficiency. *Computers* **2024**, *13*, 132. [CrossRef]
40. McFarlane, C.; Beer, M.; Brown, J.; Prendergast, N. *Patientory: A Healthcare Peer-to-Peer EMR Storage Network v1*; Entrust Inc.: Addison, TX, USA, 2017; Volume 3, p. 19.

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

future internet

MDPI

Article

A Blockchain-Based Framework for OSINT Evidence Collection and Identification

Han-Wen Huang [1], Chih-Hung Shih [1], Chen-Yu Li [2,*] and Hao-Yung Teng [1]

[1] Department of Criminal Investigation, Central Police University, Taoyuan 333322, Taiwan; ci1124012@mail.cpu.edu.tw (H.-W.H.); una171@mail.cpu.edu.tw (C.-H.S.); ci1124013@mail.cpu.edu.tw (H.-Y.T.)
[2] Department of Information Management, Central Police University, Taoyuan 333322, Taiwan
* Correspondence: cyli@mail.cpu.edu.tw

Abstract: The rapid advancement of social media and the exponential increase in online information have made open-source intelligence an essential component of modern criminal investigations. However, existing digital forensics standards mainly focus on evidence derived from controlled devices such as computers and mobile storage, providing limited guidance for social media–based intelligence. Evidence captured from online platforms is often volatile, editable, and difficult to verify, which raises doubts about its authenticity and admissibility in court. To address these challenges, this study proposes a systematic and legally compliant open-source intelligence framework aligned with digital forensics principles. The framework comprises five stages: identification, acquisition, authentication, preservation, and validation. By integrating blockchain-based notarization and image verification mechanisms into existing forensic workflows, the proposed system ensures data integrity, traceability, and authenticity. The implemented prototype demonstrates the feasibility of conducting reliable and legally compliant open-source intelligence investigations, providing law enforcement agencies with a standardized operational guideline for social media–based evidence collection.

Keywords: blockchain-based framework; open-source intelligence; digital forensics; evidence integrity; user identification

1. Introduction

Social media has become an indispensable component of modern communication, shaping how individuals interact, share content, and disseminate information. Popular platforms such as Facebook, Instagram, WhatsApp, and others collectively host billions of active users worldwide [1]. This extensive and diverse ecosystem provides a vast reservoir of publicly accessible information that can be leveraged for investigative purposes, particularly in open-source intelligence (OSINT) contexts.

Law enforcement agencies have increasingly utilized social media content to identify suspects and reconstruct criminal activities. A well-known example occurred after the 2021 U.S. Capitol riot [2], when the Federal Bureau of Investigation (FBI) extensively used Facebook data and user-generated posts to identify participants and gather supporting evidence. Similarly, in the 2015 United Kingdom cyclist insurance fraud case [3], investigators analyzed Facebook photos to reveal inconsistencies between the suspect's claims and his actual activities, ultimately leading to a criminal conviction. These cases demonstrate the evidentiary potential of open-source information available through Facebook.

Although digital forensics standards such as ISO/IEC 27037:2012 [4] and ISO/IEC 27043:2015 [5] provide comprehensive guidance for handling evidence from computers, mobile devices, and storage media, they offer limited applicability to open-source intelligence. Evidence obtained from social media platforms is typically volatile, may be altered or deleted remotely, and is often captured as screenshots without verifiable provenance. Such conditions make it difficult to prove authenticity and maintain evidential integrity in judicial proceedings. Consequently, ensuring the reliability of social media evidence has become a significant challenge in digital forensics and legal practice.

To address these challenges, this study advances three core contributions that differentiate it from prior OSINT or blockchain-based forensic research. First, we operationalize a complete OSINT evidence lifecycle that incorporates multi-factor user identity correlation before acquisition, ensuring that only attributable and case-relevant social media content is collected. This design responds directly to recurring judicial concerns about the inability to link social media accounts to real-world individuals. Second, the proposed system supports both on-site OSINT acquisition for first responders and office-based OSINT processing, reflecting the realities of criminal investigation workflows and enabling consistent handling of OSINT across operational contexts. Third, we introduce on-acquisition notarization through a permissioned judicial blockchain infrastructure, providing immediate and cross-agency verifiable proof of evidentiary integrity, authenticity, and timing for volatile OSINT. These three components collectively close the OSINT-to-courtroom gap and establish an end-to-end, legally oriented framework for handling social media evidence.

To realize this design, this study proposes a blockchain-based forensic framework for OSINT evidence collection and user identification. The framework integrates established digital forensics standards with blockchain-supported notarization and timestamping mechanisms to ensure the integrity, authenticity, and traceability of collected intelligence across various social media and online platforms.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed system architecture. Section 4 discusses experimental results. Section 5 concludes the study and outlines future research directions.

2. Related Work

2.1. Digital Evidence Standards and Challenges in OSINT

To ensure the rigor, consistency, and reliability of forensic procedures, the International Organization for Standardization (ISO) has developed a comprehensive series of standards related to this field. ISO/IEC 27037 provides essential guidelines for managing the digital-evidence lifecycle [4], offering detailed instructions for identification, collection, acquisition, and preservation. Its primary objective is to preserve evidence integrity from the outset and to prevent contamination caused by improper handling. ISO/IEC 27041:2015 builds on this foundation by emphasizing the evaluation of the suitability and effectiveness of forensic methodologies and tools [6]. ISO/IEC 27042:2015 focuses on the analysis and interpretation of digital evidence, improving analytical reliability and the consistency of results [7]. Furthermore, ISO/IEC 27043:2015 defines overarching principles and an integrated procedural framework for digital-evidence investigations, encompassing the entire process from incident response and evidence management to final analytical reporting [5]. Collectively, these standards provide a systematic foundation for digital-forensics activities, particularly in cases involving computers, mobile devices, and storage media.

While these ISO/IEC standards have been widely adopted in traditional digital forensics environments, their applicability to open-source intelligence remains limited. Evidence obtained from social media platforms is typically collected in the form of screenshots or

captured web content, which presents significant challenges in judicial evaluation. Courts must determine whether such evidence truly originates from the platform or has been modified. Unlike evidence stored on local devices, online information is highly volatile and subject to platform updates, user-side changes, and remote deletions. These characteristics complicate the preservation and verification of evidence. As open-source intelligence becomes an increasingly important component of digital investigations, establishing reliable mechanisms to ensure the authenticity and credibility of OSINT evidence has become a pressing issue for both forensic practice and judicial proceedings.

2.2. Judicial Practices and Admissibility of OSINT Evidence

In United States courts, the admissibility of open-source intelligence is primarily governed by Rule 901 of the Federal Rules of Evidence, entitled Authenticating or Identifying Evidence [8]. Under this rule, the proponent bears the burden of demonstrating that the offered evidence is what it is claimed to be. The most reliable way to meet this standard is to prepare a comprehensive digital forensics collection report that documents cryptographic hash values, source URLs, precise timestamps, and the identity of the personnel involved in the acquisition process. A broad corpus of judicial literature indicates that courts are increasingly relying on open-source intelligence, particularly social media content, in both criminal and civil proceedings [9–12].

One of the earliest and most influential decisions, Lorraine v. Markel American Insurance Company (2007) [12,13], established a five-factor framework for authenticating electronically stored information. The ruling emphasized the importance of collection integrity, metadata verification, authorship attribution, and complete procedural documentation. Subsequent rulings continued to reinforce methodological rigor. In Capitol Records v. Thomas Rasset (2011) [12,14], the court held that electronically stored information-derived evidence must be reproducible, independently verifiable, and supported by disclosures regarding tool limitations. In People v. Price (2015) [15], a New York appellate court ruled that Facebook screenshots were inadmissible because they were captured eighteen months after the alleged event, and neither authorship nor authenticity could be demonstrated to satisfy Rule 901. In United States v. Farrad (2017) [12,16], the court addressed the authentication of social media evidence collected through open-source intelligence techniques and clarified that screenshots may be admissible only when adequately authenticated. The court required the proponent to explain how the content was accessed, what tools were used, the steps taken to preserve data integrity, and any methodological limitations that might affect reliability [12]. These requirements highlight the importance of transparent acquisition procedures and robust verification practices when collecting social media evidence.

Civil proceedings have reached similar conclusions. In Montanez v. Future Vision Brain Bank (2021) [12], the court scrutinized OSINT-based evidence under principles of authenticity, chain of custody, reliability, and proportionality. This case showed that when the collection methodology lacks clarity or sufficient documentation, opposing parties may request forensic examination of the investigator's device to verify data integrity and the reliability of the acquisition process. Consistent with this trend, many judicial decisions from both civil and criminal contexts affirm that social media evidence is not automatically admissible [17]. Courts frequently dismiss digital content that lacks proper authentication, sufficient metadata, or a verifiable chain of custody. Because social media posts can be easily modified, deleted, or removed from their original context, judges routinely insist on reliable methods for preserving the original content, proving authorship, and documenting the forensic steps taken during collection. Numerous rulings involving fraud claims, personal injury disputes, and criminal investigations show that improperly collected

social media evidence is susceptible to challenges related to authorship, manipulation, or incompleteness. These patterns indicate that without rigorous authentication procedures, metadata preservation, and chain-of-custody documentation, social media evidence may be ruled inadmissible [17].

Case law from multiple jurisdictions further illustrates the risks associated with improper handling of open-source intelligence. In Canada, the ruling in R v. Hamdan (2017) [12,18] criticized investigators for relying on basic screenshot tools that did not record metadata or generate hash values, concluding that such practices weaken evidentiary reliability. A related Canadian decision, Regina v. Othman Ayed Hamdan, emphasized that open-source intelligence cannot transition from raw intelligence into admissible evidence without rigorous digital forensic procedures, including metadata extraction, cryptographic verification, and comprehensive chain-of-custody documentation. Similar considerations appear in Taiwan Supreme Court Judgment Number 320 (2023) [19], where the court held that when original digital data cannot be produced, substitute evidence such as screenshots is admissible only when supported by reliable verification methods demonstrating consistency with the original information.

These judicial decisions show that the admissibility of open-source intelligence depends not on the type of evidence but on how it is collected, authenticated, preserved, and documented. Although courts frequently admit social media evidence when proper procedures are followed, many cases demonstrate exclusion due to procedural weaknesses such as unverifiable screenshots, missing metadata, inconsistent timestamps, or insufficient documentation. These findings underscore the importance of standardized and forensic-grade OSINT workflows.

2.3. Technical Challenges in OSINT Evidence Collection and Identification

Beyond judicial case law, recent academic studies also acknowledge that the increasing reliance on open-source intelligence has created an urgent need for standardized practices in OSINT evidence collection and identification. Basumatary et al. [20] conducted a holistic review of social media forensics and identified three major admissibility challenges: integrity, data provenance, and probative value. Integrity concerns arise because social media content can be easily fabricated through the use of fake profiles or unauthorized account manipulation, making it difficult to establish that the collected data has not been altered. Issues of data provenance further complicate admissibility, as courts require clear traceability of ownership, custody, and modification history; however, provenance management in social media environments remains underdeveloped. Probative value also depends heavily on metadata such as timestamps, geolocation information, and modification history, which conventional web crawling or device-based extraction tools often fail to capture. These findings reinforce that without consistent procedures for metadata extraction, authenticity assessment, and chain-of-custody documentation, social media evidence may lack the reliability required for legal acceptance.

Gupta et al. [10] also examined methods for recovering digital traces from social media platforms and identified two central challenges. The first is the absence of a standardized methodology tailored for social media forensics, since existing frameworks do not address platform-specific constraints. The second is the lack of specialized tools capable of integrating social media artifacts into traditional forensic workflows, as most conventional tools cannot effectively process social media data. These limitations highlight the continued need for dedicated methodologies and tools that support reliable social media forensic analysis.

To address such challenges, several studies propose technical frameworks to support forensic OSINT workflows. Vidya et al. [21] introduced a system with acquisition and analysis functionalities that can crawl websites, preserve digital objects, and generate reports for both acquisition and analysis. Although this approach offers strong webpage archiving capabilities, it often collects excessive amounts of irrelevant data when applied to social media platforms, which makes it difficult for investigators to reconstruct case-relevant narratives. Greco et al. [22] proposed a certification framework based on a controlled browser operated by a certification authority to capture screenshots with comprehensive metadata. This approach enhances integrity but faces practical restrictions in real investigations because social media platforms frequently limit content access through privacy controls that require authenticated sessions.

These limitations suggest that although existing automated frameworks offer valuable features, they are insufficient for practical use in real-world criminal investigation environments. Investigators often work with devices seized at the crime scene that are already logged into the suspect's social media accounts, and they must preserve this authenticated environment without altering the original digital state of the device. In other situations, investigators may need to obtain OSINT evidence from an office environment rather than from a suspect's device, which further underscores the need for a consistent, user-friendly, and operationally flexible tool that supports frontline personnel in both field-based and office-based OSINT acquisition tasks. In addition, many automated collection architectures generate large volumes of irrelevant social media data that still require manual examination to determine evidentiary value. This creates a significant analytical burden for investigators and may delay urgent investigations. Excessive automation without effective relevance filtering does not necessarily improve operational efficiency and can increase the workload for personnel in real criminal cases.

2.4. Blockchain Applications in Judicial Evidence Preservation and OSINT

Blockchain technology has emerged as a vital solution for secure data preservation and evidence management, largely due to its decentralized architecture and inherent security characteristics. As noted in [23], blockchain systems exhibit several essential attributes, including decentralization, autonomy, data integrity, immutability, verifiability, fault tolerance, anonymity, auditability, and transparency. Through the use of cryptographic hashing algorithms such as SHA-256 and digital signatures, blockchain ensures high data trustworthiness, preserving integrity and preventing unauthorized alteration.

In practical applications, blockchain technology effectively addresses key challenges in digital evidence preservation, particularly within judicial contexts. For example, Ref. [24] proposed a blockchain-enabled judicial evidence framework that integrates the Interplanetary File System (IPFS). In this model, the IPFS distributed storage system is used to store digital evidence, while blockchain records and secures the corresponding hash values and transactional metadata. This integration enhances the security, authenticity, and integrity of judicial evidence. Similarly, Ref. [25] demonstrated that combining blockchain with cloud-storage environments through distributed-ledger mechanisms and Provable Data Possession (PDP) protocols can ensure the long-term credibility, integrity, and accessibility of electronic records, providing a reliable foundation for sustainable evidence management. To clearly distinguish the contributions of our work, Table 1 provides a comparative analysis of this study against these aforementioned blockchain-based digital evidence preservation approaches.

To strengthen digital transformation and improve data integrity in the judicial sector, Taiwan established the Blockchain-applied Judicial Alliance for Digital Era (b-JADE) in

2022 [26]. The consortium blockchain was jointly developed by five founding institutions in Taiwan: the Ministry of Justice (MOJ), Judicial Yuan (JY), Taiwan High Prosecutors Office (THPO), National Police Agency (NPA), and Ministry of Justice Investigation Bureau (MJIB) as shown in Figure 1. The b-JADE infrastructure is based on the Hyperledger Fabric framework and employs a permissioned blockchain architecture that requires authorized participation.

Table 1. Comparison of this study with other blockchain-based digital evidence preservation approaches.

Work (Year)	Primary Focus	Identity Correlation Capability	Legal Compliance Features	Storage and Preservation Model
Deng et al. (2019) [25]	Trusted preservation of electronic records in cloud environments	Not supported	Ensures integrity and availability through PDP	Cloud storage combined with PDP and distributed ledger techniques
Liu & Zheng (2024) [24]	Judicial evidence preservation using blockchain and IPFS	Not supported	Discusses authenticity and integrity requirements	IPFS for evidence storage with hashes recorded on chain
This work	OSINT acquisition and transformation into court admissible evidence (social media focused)	Supported (multi-factor cross-platform correlation)	Alignment with Taiwan Judicial Alliance practice and explicit mapping to U.S. FRE	Off-chain forensic evidence vault with hash values and metadata recorded on b-JADE

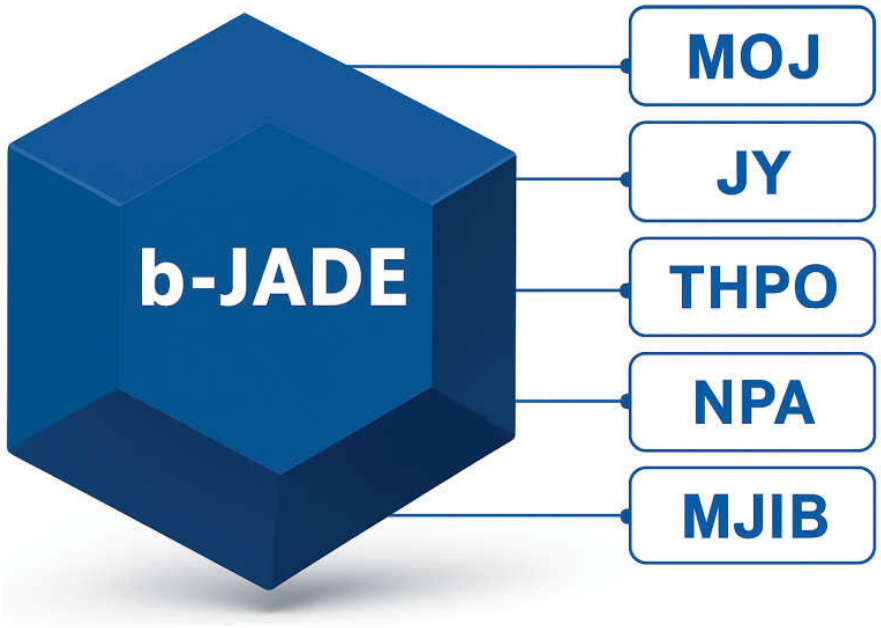


Figure 1. Blockchain-applied Judicial Alliance for Digital Era.

Operationally, once digital evidence is collected, the system automatically generates a cryptographic hash of the file. The hash value, along with relevant case information, timestamps, and metadata, is encapsulated into a transaction and uploaded to the blockchain for notarization. Each transaction must be endorsed by validation nodes, which are independently managed by consortium members, through a consensus algorithm. After consensus is reached, the transaction is permanently recorded on the shared ledger [27].

Building on these implementations, blockchain technology offers significant advantages when applied to OSINT evidence collection. By generating cryptographic hashes and recording OSINT data on-chain at the time of acquisition, the proposed framework ensures immutability, verifiability, and a tamper-evident record. This real-time notarization mitigates risks associated with data volatility, unauthorized modification, or deletion, thereby improving the evidential reliability and legal admissibility of OSINT in judicial proceedings.

While this study focuses on the integrity and preservation of OSINT data, recent advancements in 2025 have significantly expanded the scope of blockchain forensics, particularly in attribution and fraud detection. Emerging research has demonstrated the efficacy of AI-enhanced models and dynamic graph neural networks (GNNs) for identifying illicit activities on public ledgers. For instance, recent studies such as Sheng et al. (2025) [28] and Tian et al. (2025) [29] have proposed novel architectures like Dynamic Feature Fusion and DiT-SGCR (Directed Temporal Structural Representation), which integrate global graph structures with local semantic features to detect sophisticated fraud patterns in real-time. Similarly, advancements in large-scale data processing presented in [30,31] leverage dynamic graph-based approaches to analyze transaction flows with greater granularity, while [32] highlights the potential of combining machine learning with massive blockchain datasets to trace complex cybercriminal behaviors. Although our framework primarily targets the preservation of social media evidence rather than on-chain financial fraud detection, these advanced analytical methods offer a complementary perspective. They suggest that future forensic frameworks could integrate such AI-driven attribution models to further correlate OSINT findings with on-chain transactional evidence, thereby strengthening the link between digital identities and real-world actors.

2.5. Social Network User Identification

Although blockchain-based solutions ensure the immutability, verifiability, and tamper-evident preservation of OSINT data, they do not inherently reveal the real-world identities of account holders. Even when digital evidence is securely notarized on-chain, identifying the individual behind a social media account remains one of the most challenging tasks in forensic investigations. Solving this problem requires systematic strategies that integrate multiple sources of user-generated data and behavioral indicators to enable probabilistic or deterministic inference of user identities. Existing user-identification techniques on social networks can generally be divided into two main categories: profile-based and network-structure-based approaches.

2.5.1. Profile-Based User Identification

User profile information consists of data voluntarily provided or selected during social-network registration, such as usernames, gender, birth dates, and other personal attributes that are publicly available online. Prior research has shown considerable similarity in user-profile features across different social networks, revealing a degree of cross-platform consistency that can be exploited for user identification.

Zafarani et al. [33] were among the earliest to propose a cross-platform identification approach by comparing usernames across distinct social-networking sites. Wang et al. [34] extracted alphanumeric and date-related patterns from usernames to improve matching accuracy. Li et al. [35] analyzed variations in username-selection behaviors across networks and applied machine-learning algorithms to detect accounts likely belonging to the same user. Motoyama et al. [36] crawled and analyzed user profiles from multiple platforms, representing them as keyword sets and calculating similarity scores between these sets to estimate account correspondence. Raad et al. [37] designed a matching method using the Friend-of-a-Friend

(FOAF) ontology, converting profile data into standardized FOAF descriptors and applying decision algorithms to quantify inter-account similarity. Iofciu [38] introduced a subjective-weighting model that combined usernames and user-generated tags to compute weighted similarity, whereas Ye et al. [39] adopted an objective-weighting scheme derived from subjective evaluations to measure attribute-based similarity among profiles.

2.5.2. Network-Structure-Based User Identification

Network-based identification relies on the topology of social connections, particularly friendship relations, to match users by analyzing node-similarity metrics. The interaction networks generated by user activities often exhibit structural symmetry across different social platforms. Because friendship networks tend to remain stable over time, cross-platform user identification can be achieved by examining overlaps among mutual acquaintances detected on multiple networks.

Narayanan et al. [40] first demonstrated successful identification using network-topology information alone. Building on this foundation, Cui et al. [41] integrated user-profile data with graph-similarity measures to link accounts between email-based networks and Facebook's social-network structure.

Both profile-based and network-based methods have inherent limitations when used independently, particularly in criminal investigations where user data are sparse, fragmented, or incomplete. Combining the two approaches allows investigators to leverage complementary strengths, resulting in more reliable and higher-confidence identification. This integrated strategy provides a robust analytical framework for associating OSINT-derived digital traces with real-world identities while maintaining the integrity, traceability, and admissibility of the corresponding evidence.

3. Proposed OSINT Evidence Collection Methodology

To address these challenges, this study establishes a systematic and legally compliant framework for the collection and preservation of OSINT evidence. The framework is developed in accordance with international digital-forensics standards, including ISO/IEC 27037, 27041, 27042, and 27043, which collectively define procedures for the identification, acquisition, authentication, preservation, and validation of digital evidence. The proposed methodology follows this structure to ensure procedural clarity and maintain consistency with established forensic best practices.

While ISO/IEC 27037 formally defines four primary stages of digital evidence handling (identification, collection, acquisition, and preservation), this study extends the framework by incorporating the authentication and validation concepts outlined in ISO/IEC 27041 and ISO/IEC 27043. This enhancement ensures procedural transparency, auditability, and legal defensibility throughout the entire OSINT investigation workflow, aligning the proposed framework with the most recent international standards for end-to-end digital forensic processes.

To realize this framework in practice, a digital-evidence collection system named CIB-Triage III (Figure 2) was developed. The system operates the framework by supporting modular acquisition, metadata extraction, hashing, and verification within a unified workflow. It enables evidence collection from mobile devices, computers, and cloud platforms, and its automated processes minimize human intervention, thereby reducing the risk of contamination or tampering.

Unlike traditional digital forensics, OSINT data are typically obtained from open networks and social media platforms rather than devices owned by suspects. As such, linking online evidence to a specific individual in the real world presents a significant challenge. Establishing

this connection requires a combination of behavioral analysis, profile correlation, and network structure assessment to verify that the collected information accurately represents the activities of the identified person. This verification process is essential for ensuring that OSINT evidence is both authentic and attributable before it is formally acquired and preserved.



Figure 2. (**a**) CIB-Triage III integrates evidence collection tools for computers, mobile devices, and cloud data, along with mechanisms for on-chain evidence registration and verification. (**b**) The screenshot and recording functions within the digital forensic toolkit allow investigators to capture visual data from social networking platforms as digital evidence. (**c**) Each captured image is automatically annotated with metadata, including the timestamp and file size, and a cryptographic hash value is generated to verify the integrity and authenticity of the evidence.

Within the context of OSINT, the Evidence Identification Phase differs from traditional digital-forensics practice. Although ISO/IEC 27037 defines identification as the process of recognizing potential sources of digital evidence, OSINT extends this phase to include the evaluation of publicly available materials distributed across social media and online ecosystems. This phase focuses on identifying and validating OSINT data that is relevant to the specific investigation target. By examining contextual relationships, user interactions, and data provenance, the process determines whether the collected online content can be legitimately associated with the subject under investigation. This refined identification phase ensures that subsequent acquisition, authentication, preservation, and validation procedures are performed on evidence that is both relevant and verifiably connected to the intended individual or incident.

After OSINT authenticity has been verified and the social media account has been credibly linked to a real-world individual, the corresponding evidentiary data are collected using CIB-Triage III. This system performs modular acquisition of verified OSINT artifacts from online platforms, ensuring that each item is accompanied by metadata, precise timestamps, and cryptographic hash values (e.g., SHA-256, acquisition timestamp, collector ID, content length) to maintain data integrity and verifiability. The evidence bytes remain off-chain in the evidence vault. The verified and authenticated evidence, along with its metadata and hash values, is subsequently uploaded to b-JADE for blockchain notarization. By leveraging the decentralized and immutable architecture of blockchain, the framework preserves authoritative evidence states, enables immediate verification of authenticity, and mitigates risks associated with the volatility of online content. Integration with the Police Digital Evidence Chain further enhances inter-agency trust and procedural transparency, strengthening the legal defensibility of OSINT-derived evidence in judicial and administrative proceedings.

The following sections provide a detailed explanation of the proposed framework, elaborating on the identification, acquisition, authentication, preservation, and validation phases (Figure 3). Each phase is discussed in terms of its objectives, procedures, and mechanisms, illustrating how the framework systematically implements OSINT evidence management in a transparent and legally defensible manner. To further enhance clarity, Figure 4 visualizes the complete data flow from OSINT collection through identity correlation, off-chain storage, and on-chain notarization. This architectural illustration highlights the interaction between CIB-Triage III and b-JADE, the separation of trust boundaries, and the integration between on-chain and off-chain components, providing a clear understanding of how evidence moves through the system.

3.1. Identification Phase

3.1.1. Platform and Tool Preparation

The platform and tool preparation sub-phase ensures that all technical resources are ready and that pre-existing intelligence is systematically organized, establishing a solid foundation for subsequent investigative activities. All hardware and software required for evidence collection are configured and validated according to the specific objectives of the investigation. Core tools include screen-capture applications, Optical Character Recognition (OCR) utilities for converting text embedded in images into machine-readable formats, and analytical software such as mind-mapping tools used to conduct relational analysis and visualize complex intelligence structures.

Integrated forensic platforms, such as CIB-Triage III, further enhance operational rigor by enabling continuous recording of digital interactions, automatic extraction of textual content, and secure computation and storage of cryptographic hash values alongside es-

sential metadata, including timestamps and source identifiers. Collected evidence can be uploaded either immediately or in batches to b-JADE, leveraging blockchain's decentralized, immutable, and auditable characteristics to ensure authenticity and integrity across the entire chain of custody. By formalizing these preparatory procedures and aligning them with forensic and legal standards, the framework enhances both operational efficiency and the reliability of the resulting evidence.

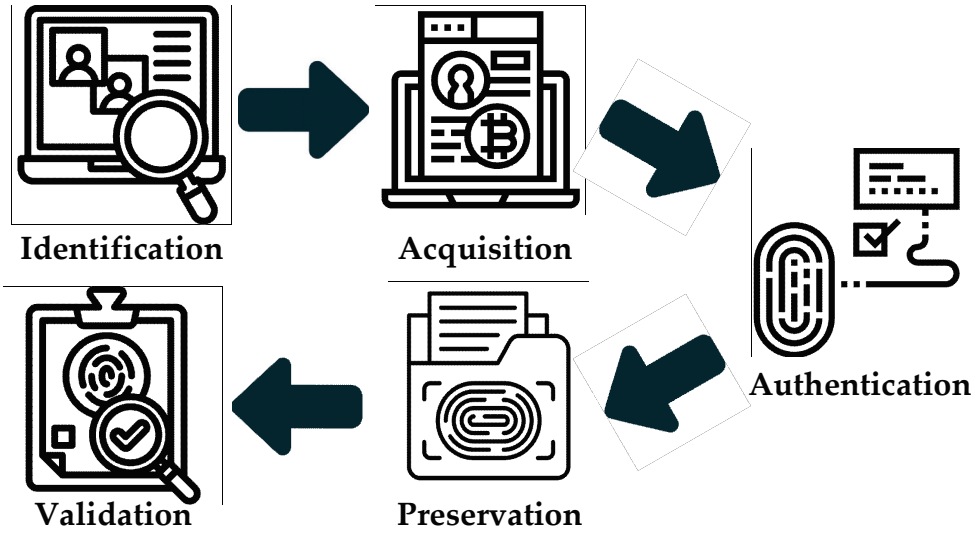


Figure 3. Operational workflow of the proposed OSINT evidence collection framework.

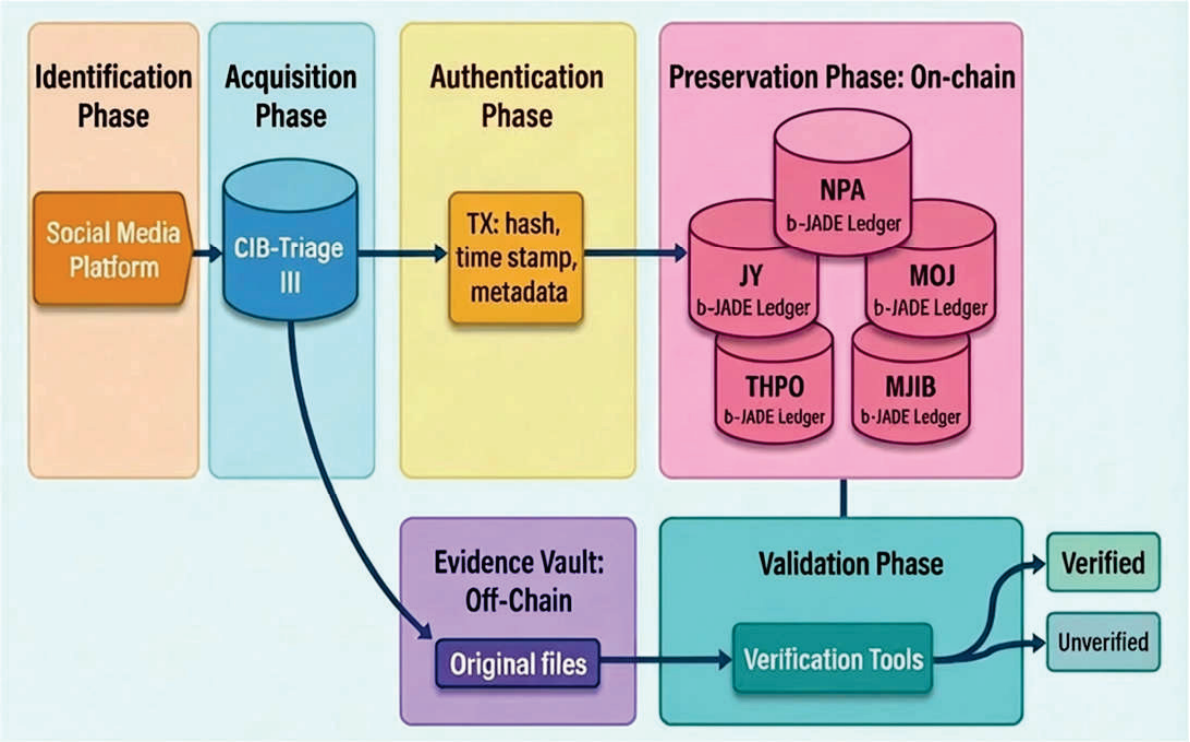


Figure 4. System architecture of the proposed OSINT evidence framework.

3.1.2. Target Background and Relevant Information

Before initiating active OSINT collection, investigators systematically compile comprehensive background information on the target of the investigation. This preparatory

step ensures that subsequent data acquisition and analysis are conducted with precision, legality, and a clear understanding of the context. The compiled information includes personal identifiers such as full names, known aliases, demographic characteristics, professional and educational backgrounds, contact details, frequently visited locations, and photographic records.

The data compilation also extends to individuals closely associated with the target, including family members, colleagues, and other relevant affiliates, along with any supporting documentation, such as news articles, social media posts, or public records. This structured dataset serves as a critical foundation for subsequent stages of OSINT investigation, facilitating correlation, behavioral analysis, and identity verification. By conducting this process under standardized forensic and legal protocols, investigators ensure that all preparatory activities are performed within a methodologically sound and legally defensible framework.

3.1.3. User Identity Correlation Phase

The User Identity Correlation Phase focuses on verifying and correlating online identities with real-world individuals through a progressive multifactor comparison mechanism. The process begins with surface-level indicators available on social media and gradually advances toward more detailed and personalized information that provides stronger corroboration. This layered approach ensures that identity attribution proceeds from low-risk, easily observable factors to high-confidence, evidence-based verification, thereby enhancing both accuracy and legal defensibility.

By integrating multiple identification criteria in a structured sequence, OSINT data are cross-validated across diverse sources to minimize misidentification risks and strengthen investigative precision. The verification process consists of four levels, each increasing in depth and reliability, building upon the previous one.

1. Identical Account or Name Comparison

This method represents the entry point of the analysis, identifying potential users based on identical or highly similar usernames across multiple platforms. For example, if a person uses the account name A1 on platform Alpha and a nearly identical username on platform Beta, there is a strong probability that both accounts are operated by the same individual. The reliability of this approach increases with the uniqueness or rarity of the username.

2. Relationship Network Comparison

The next level examines social relationships and connections between followers and friends within online platforms. If user A is known to have siblings named Alpha and Beta, and those accounts appear in the friend list of a particular Facebook profile, it can be reasonably inferred that the profile is likely controlled by user A. This method leverages stable interpersonal ties as mid-level corroborative evidence, enhancing confidence in user identification.

3. Special Event Comparison

At this stage, the analysis incorporates identifiable events with distinct temporal markers such as birthdays or anniversaries. For instance, if user A's birthday is known to be March 3 and a Facebook account receives numerous birthday messages precisely on that date, the likelihood that the account belongs to user A increases significantly. Event-based temporal correlation thus provides contextual validation linking digital activity with real-world timelines.

4. Individualized Content Comparison

The final and most in-depth layer focuses on highly personalized information voluntarily or inadvertently disclosed online. Examples include photographs with recognizable locations, visible vehicle license plates, or images related to banks. If user A posts a photograph that clearly displays identifiable features such as a car license plate or a distinctive environment, it provides compelling corroborative evidence that the account is indeed operated by that individual. This method yields the highest level of confidence, provided that the evidence is obtained and preserved in compliance with legal and forensic standards.

3.2. Acquisition Phase

The Acquisition Phase translates the outputs of the Identification Phase into actionable evidence collection. Building on the previously established verified OSINT targets and sources, this stage focuses on the controlled acquisition of relevant data while maintaining strict forensic soundness. All collection activities are conducted in accordance with ISO/IEC 27037 guidelines to ensure that each acquired item retains its authenticity, integrity, and traceability throughout the process.

3.2.1. Collection

The Collection sub-phase involves the systematic acquisition of publicly available data that have been pre-validated as relevant to the investigation. Data sources include major social media platforms (Facebook, Instagram, and X), instant messaging applications, online forums, blogs, and cryptocurrency exchanges. To preserve authenticity and integrity, investigators use digital screenshots, continuous screen recording, and other non-intrusive methods that do not alter the original data.

Key elements of interest include usernames, account identifiers, profile attributes, user-generated content, social relationships (friends, followers, commenters), temporally defined activities (birthday posts, event participation), and individualized digital artifacts such as images, videos, vehicle license plates, and geolocation data. Formalizing and standardizing these procedures ensures reproducibility, completeness, and compliance with evidentiary requirements.

3.2.2. Analysis

After collection, the acquired data are organized and filtered to remove redundancy and irrelevant information. The analysis aims to extract critical, individualized intelligence components that contribute to establishing evidential value. Each dataset is examined for authenticity, integrity, and temporal accuracy before being prepared for authentication and preservation.

Analytical tools assist this process: Google Lens can detect duplicate images across platforms, while deep-fake-detection tools such as Sensity [42] and Reality Defender [43] verify content authenticity. Captured images are processed with OCR software to facilitate textual indexing, and visualization applications such as Xmind [44] help investigators map and understand complex relationships within the collected intelligence.

3.3. Authentication Phase

The authentication phase ensures that all OSINT evidence collected in the acquisition stage is genuine, unaltered, and traceable to its original source. This phase verifies the integrity and authenticity of each data item by applying forensic validation techniques and establishing a clear chain of custody. The authentication process is particularly critical for

OSINT because the information originates from open and dynamic online environments, where digital content can be modified, deleted, or fabricated without notice.

Verification begins by confirming the provenance and consistency of each evidence item. Metadata such as timestamps, URLs, device information, and digital signatures are analyzed to confirm their coherence with the original collection records. Cryptographic hash values generated during acquisition are recalculated and compared to the originals to verify that no alterations occurred. Any discrepancy triggers an integrity review, and the affected data are either revalidated or excluded from further evidentiary use.

To further enhance reliability, blockchain-based notarization mechanisms such as b-JADE are utilized to record hash values and essential metadata on a decentralized ledger. This ensures long-term immutability and auditability, allowing investigators and judicial authorities to independently verify the authenticity of evidence at any point in time. Additionally, all authentication activities are documented within the forensic management system to ensure procedural transparency and compliance with ISO/IEC 27001:2022 [45] and 27002:2022 [46] requirements.

Through this systematic verification process, the framework guarantees that every authenticated OSINT artifact meets the evidentiary standards of integrity, originality, and reproducibility required for legal admissibility.

3.4. Preservation

The Preservation Phase focuses on maintaining the integrity, authenticity, and continuity of OSINT evidence after authentication. To reinforce evidentiary reliability, the proposed framework integrates the preservation process directly into the CIB-Triage III platform. This integration enables automated computation of unique cryptographic hash values for each digital artifact, ensuring that all evidence is systematically verified and securely recorded. Immutable hash records serve as digital fingerprints, preventing post-collection modification and maintaining an unbroken chain of custody.

By embedding OSINT preservation procedures into an established forensic platform, the framework extends conventional digital-evidence management practices. It provides investigators with a standardized, reproducible, and legally defensible methodology for long-term intelligence preservation, fully compliant with the principles outlined in ISO/IEC 27042 and 27043.

3.5. Validation Phase

The Validation Phase consolidates and formalizes all outputs from the preceding collection, analysis, and identification stages into a structured, auditable, and transparent forensic record. Within this phase, all investigative findings, analytical reasoning, and operational methodologies are systematically documented to ensure reproducibility, traceability, and evidentiary validity.

Following the principles outlined in the Berkeley Protocol on Digital Open Source Investigations [47], the framework structures the final report to include:

1. Clearly defined investigative objectives;
2. Comprehensive description of the methodology, including procedural steps and adherence to legal and forensic standards;
3. A chronological account of activities conducted during collection, analysis, and identification;
4. Detailed documentation of data sources and provenance to maintain transparency and traceability;

5. Acknowledgment of uncertainties, data gaps, or limitations encountered during the investigation; and
6. Evidence-based findings supported by actionable recommendations.

Furthermore, the framework incorporates visual analytic representations, such as network diagrams, charts, and relational maps, to illustrate complex interconnections and facilitate the interpretation of analytical conclusions. By formalizing the report generation process within this structure, the Validation Phase ensures that OSINT investigations are systematic, transparent, and legally defensible, while also providing a replicable methodology for knowledge transfer and future investigative or judicial use.

3.6. Legal Admissibility and Compliance Mapping

This section maps the OSINT lifecycle employed in this study, which consists of Identification, Acquisition, Authentication, Preservation, and Validation, to the evidentiary requirements of the United States Federal Rules of Evidence (FRE). This alignment ensures that each stage of the proposed framework satisfies specific legal thresholds required for admissibility:

1. Authenticity (FRE 901): The requirement of Rule 901(a) is satisfied through a documented, repeatable process. The system captures comprehensive logs, precise timestamps, and the SHA-256 hash of the original data at the moment of capture. This digest, along with minimal metadata, is immediately notarized on the b-JADE consortium ledger to freeze the state of evidence.
2. Self-Authentication (FRE 902): Requirements under Rules 902(13) and 902(14) regarding certified records of generated electronic information are met through a dual-verification mechanism. This involves a custodian's certificate describing the system's operation and a verifiable match between the hash of the preserved object in the off-chain vault and the immutable digest recorded on-chain (referenced by transaction ID).
3. Best Evidence Rule (FRE 1001–1003): The framework complies with the "Best Evidence" principle by preserving the original binary data in a Write-Once-Read-Many (WORM) evidence vault. It allows for the production of the original or a bit-for-bit duplicate, with integrity demonstrated through deterministic re-hashing and on-chain proof verification.

Through these mechanisms, our framework ensures that OSINT evidence collected through the proposed lifecycle meets the legal standards for authentication, integrity, provenance, and evidentiary sufficiency required in judicial proceedings. By embedding admissibility considerations directly into the forensic workflow, the system strengthens the reliability and defensibility of OSINT evidence in both investigative and courtroom contexts.

4. Implementation of the OSINT Evidence Collection Framework

This section presents a prototype implementation of the proposed framework that demonstrates feasibility and workflow integration; large-scale, real-world deployment and evaluation are left to future work. The implementation follows the five sequential phases established in Section 3, namely identification, acquisition, authentication, preservation, and validation. Each phase has been modularized into dedicated software components and operational procedures to ensure that the investigative workflow remains systematic, legally compliant, and forensically verifiable.

Given its large user base, Facebook [1] is used here as a case study platform for implementing and demonstrating the pipeline. However, the system architecture is designed to be

applicable to other OSINT sources such as X (Twitter), Instagram, and additional social network platforms. This platform, once fully developed, can potentially serve as a foundation for testing and validating the proposed framework. The system architecture is designed to be extensible and can be expanded in future versions to incorporate other OSINT sources.

The implementation emphasizes procedural rigor and reproducibility while integrating cryptographic verification and blockchain-based preservation mechanisms to ensure the integrity of digital evidence throughout the forensic process.

4.1. Identification Phase

In the identification phase, investigators prepare the technical environment and compile relevant background information on the investigative target. This preparation ensures that all required tools and intelligence resources are properly organized before active evidence collection begins. The identification phase, therefore, serves as the foundation of the entire OSINT workflow, linking investigative planning with the operational procedures described in subsequent stages.

4.1.1. Platform and Tools Preparation

The platform and tools preparation phase ensures that investigators are equipped with the essential technical resources, software platforms, and baseline intelligence required to perform efficient and legally compliant OSINT evidence collection. All tools incorporated into the framework correspond to the five functional phases of the OSINT evidence collection process, namely identification, acquisition, authentication, preservation, and validation. Excluding the validation phase, seven primary tools have been implemented to support the core operations of the system. These tools establish a standardized and reproducible technical environment that aligns with the principles of ISO/IEC 27037, supporting accurate, transparent, and legally defensible OSINT operations within a forensic context.

Among these tools, CIB-Triage III serves as the central platform, providing comprehensive functionality for screen capture, continuous screen recording, Optical Character Recognition (OCR), and blockchain-based preservation through b-JADE. The Google Lens utility, accessible directly via the Chrome browser, enables rapid image correlation and cross-platform verification. Xmind (version 26.01) facilitates structured visualization and relationship mapping, while deepfake detection tools such as Sensity and Reality Defender are employed to verify the authenticity of visual content (Figure 5).

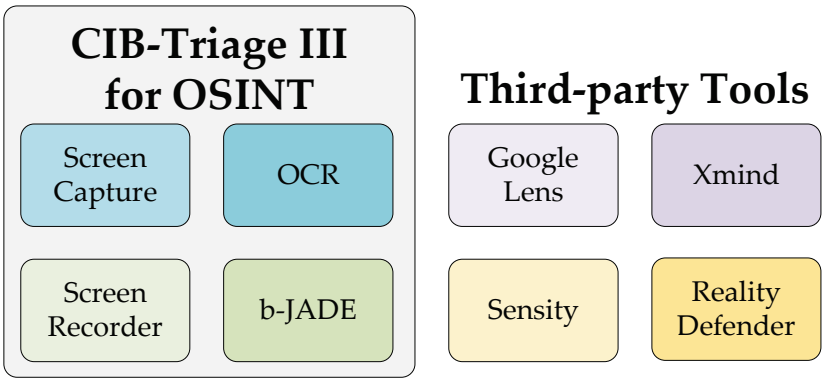


Figure 5. Tools utilized in the implementation of the OSINT evidence collection framework.

4.1.2. Target Background and Relevant Information

During this stage, investigators compile comprehensive background and familial information concerning the target. Police information systems are accessed to retrieve a list

of the target's relatives, including the collection of fundamental biographical data such as names, photographs, and dates of birth for all individuals within the fourth-degree kinship range. Public academic databases such as Google Scholar, along with advanced search queries using filters like "site:edu.xx" or "site:ac.xx," are also utilized to identify the target's academic background, research expertise, and potential classmates or collaborators.

Additional data sources, such as stop and search records, co-travel information from immigration databases, and visitation logs from detention facilities, are examined to uncover the target's close associates or possible accomplices. All gathered intelligence is subsequently organized with Xmind to construct a comprehensive mind map of the target (Figure 6), providing a visual overview that facilitates subsequent data analysis and operational assessment.

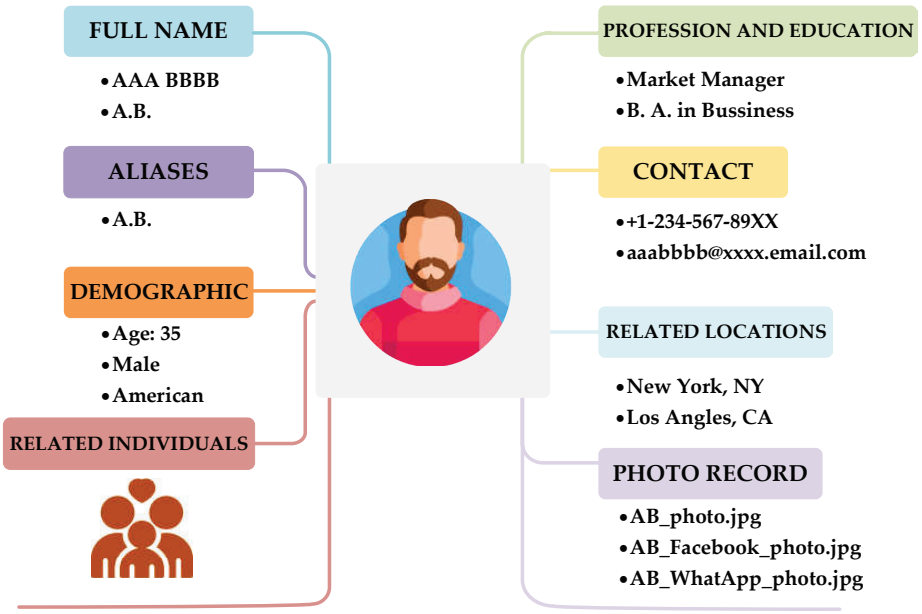


Figure 6. Map of target basic and relative information.

4.1.3. User Identity Correlation Phase

A multi-factor verification approach is employed to confirm account attribution. Relying on a single factor, such as a matching photograph, provides limited confidence due to the potential for impersonation or coincidental similarity. To address this limitation, the verification process follows a probabilistic reasoning model inspired by Bayesian theory. Each independent indicator contributes to the posterior probability of correct attribution according to the formula:

$$P(A \mid B_1, B_1, \ldots, B_N) = k \cdot P(A) \cdot \prod_{i=1}^{N} P(B_i \mid A)$$

where $P(A \mid B_1, B_1, \ldots, B_N)$ represents the probability that the account truly belongs to the target given N independent corroborating factors $B_1, B_1, \ldots, B_N$. The constant k is a normalization factor that ensures the total probability across all potential account candidates equals one. In practical applications, investigators are primarily concerned with relative probabilities rather than the explicit value of k, since the normalization term does not affect comparative attribution results.

In this operational framework, the specific weights $P(B_i \mid A)$ are assigned according to the evidentiary strength of each category. For example, 'Individualized Content Compar-

ison' (high weight) receives a substantially higher weight than 'Identical Account Name' (low weight) because the likelihood of accidental duplication is far lower for physical identifiers such as license plates or geolocated imagery than for usernames. This hierarchical weighting ensures that identity correlation is grounded in high-confidence indicators prior to final attribution.

As additional consistent evidence is incorporated, such as overlapping friend networks, recurring event timestamps, and unique identifiers, the cumulative probability of accurate attribution increases multiplicatively. This quantitative framework mirrors fingerprint identification principles, in which each new point of correspondence raises the overall confidence level and significantly reduces the likelihood of false attribution.

To operationalize this framework, investigators apply the four OSINT evidence classification methods introduced in Section 3.1.3. Representative examples are shown in Figures 7–11 and are described as follows:

1. Identical Account or Name Comparison. Investigators begin by using keywords or names to locate accounts with identical or similar usernames. For instance, Facebook's search function can identify accounts displaying public profile pictures, posts, and interaction histories. LINE is a messaging and social platform widely used in East Asia, and its functionalities are similar to those of WhatsApp and Facebook Messenger. LINE allows account searches using either user IDs or phone numbers. As shown in Figure 7, correlation is strengthened when both platforms use the same account name, such as the username "PKNEVXXXXXE."
2. Relationship Network Comparison. Public friend lists and post-interaction records, such as likes and comments, can be analyzed to map social networks (Figures 8 and 9). Even when friend lists are private, repeated engagement from specific accounts may indicate closer relationships.
3. Special Event Comparison. Posts associated with birthdays, memorials, or anniversaries (Figure 10) often reveal key personal relationships and life events, helping to corroborate identity.
4. Individualized Content Comparison. Posts may inadvertently disclose distinctive information such as locations, vehicle license plates, or household environments. Although each element may appear minor, these details become valuable when corroborated with other intelligence (Figure 11).

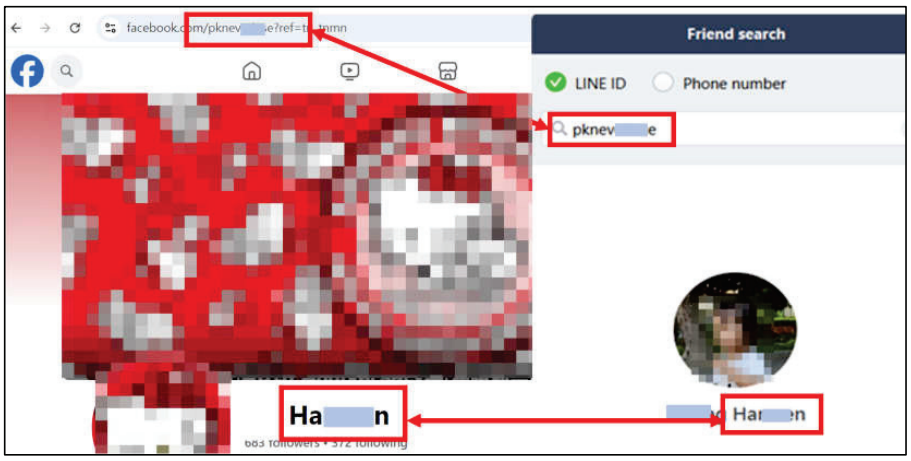


Figure 7. LINE and Facebook accounts using the same username "PKNEVXXXXXE," demonstrating cross-platform identity correlation.

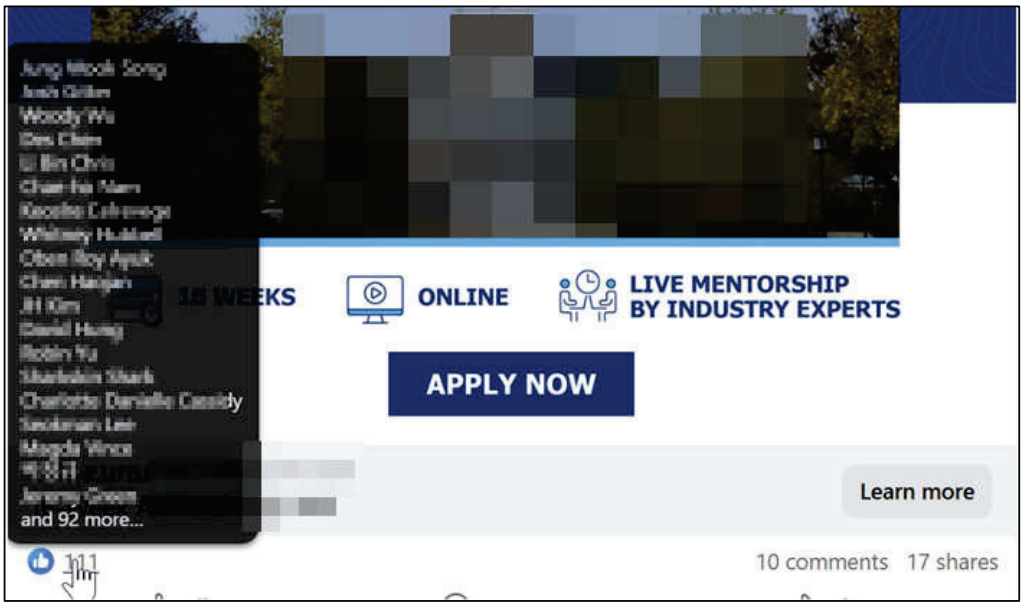


Figure 8. Facebook "Like" interaction records illustrating relationship mapping through engagement patterns.

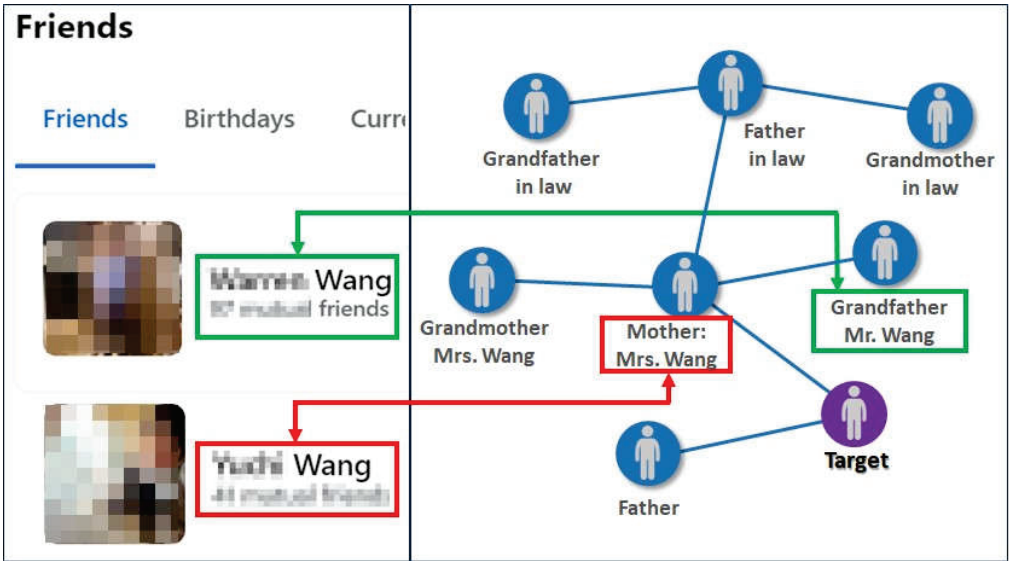


Figure 9. Comparison between the target's social media connections and known family and friend relationships.

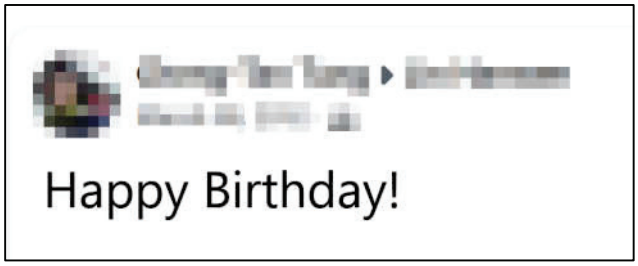


Figure 10. Facebook post containing a special event message "Happy Birthday!", showing temporal and relational cues useful for identity verification.

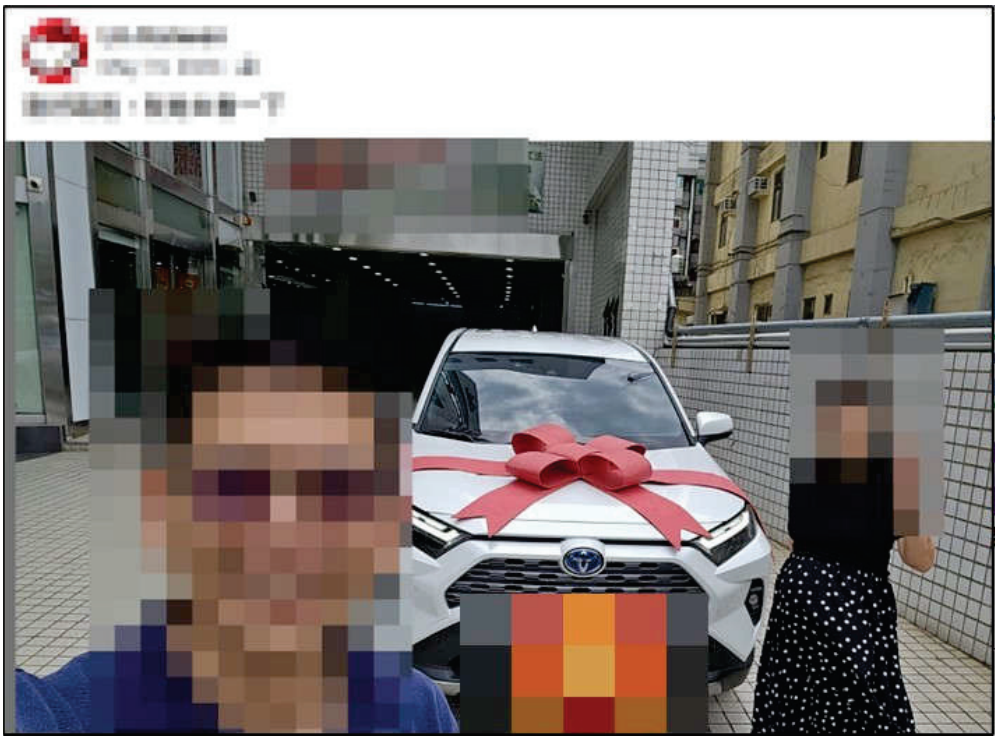


Figure 11. Facebook post displays a car license plate, representing individualized content that supports identity corroboration.

Once multi-factor identification confirms the actual user of a social media account, investigators then transition back to the Collection sub-phase to conduct broader data acquisition. At this stage, the CIB-Triage III system functions as the core acquisition platform, providing real-time screen capture and continuous recording capabilities. These functionalities support a comprehensive collection of both static and dynamic data across websites, social media platforms, and other publicly accessible sources, ensuring completeness and traceability of collected intelligence. Intelligence directly related to the target, such as check-in records or tagged locations, provides valuable insights into the target's activities, timeline, and associations. Finally, all information relevant to the investigation is securely preserved through the Preservation sub-phase by recording integrity-protected artifacts, such as hash values, onto the b-JADE blockchain, ensuring long-term traceability and evidentiary admissibility.

4.2. Acquisition Phase

4.2.1. Collection

The Collection sub-phase involves the systematic acquisition of publicly available data that have been pre-validated as relevant to the investigation. The CIB-Triage III system serves as the core acquisition platform, providing real-time screen capture and continuous recording capabilities (Figure 12). To preserve authenticity and integrity, investigators employ digital screenshots, continuous screen recording, and other non-intrusive techniques that do not modify the original data.

These functionalities support comprehensive acquisition of both static and dynamic data across websites, social media platforms, and other publicly accessible sources, ensuring completeness and traceability of collected intelligence. By centralizing these operations within a single platform, the framework achieves consistent data-handling procedures that

comply with ISO/IEC 27037 guidelines and maintain forensic soundness throughout the acquisition process.



Figure 12. Real-time screen capture and continuous recording are performed using the CIB-Triage III platform.

4.2.2. Analysis

After collection, the acquired data are systematically organized, filtered, and verified to remove redundancy and irrelevant information. The analysis aims to extract critical and individualized intelligence components that contribute to establishing evidential value. Each dataset is examined for authenticity, integrity, and temporal accuracy before being prepared for authentication and preservation.

To enhance analytical rigor and efficiency, multiple complementary tools are integrated into the framework:

- OCR within CIB-Triage III converts textual content embedded in images into searchable formats, enabling rapid extraction of relevant information from large datasets (Figure 13).
- Mind-mapping software, such as Xmind, provides structured visualization and relational analysis, allowing investigators to systematically synthesize fragmented intelligence (Figure 14).
- Reverse image search tools, such as Google Lens, enable verification of whether visual content is already publicly available, misappropriated, or altered.
- Deepfake detection platforms, such as Sensity and Reality Defender, are employed to evaluate the authenticity of images and videos, minimizing the risk of incorporating manipulated content into the investigation.

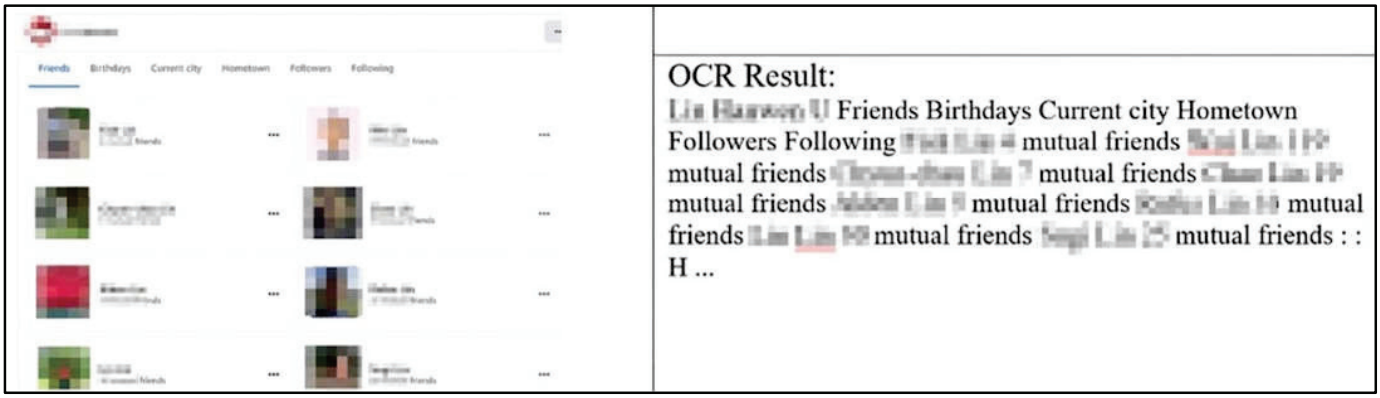


Figure 13. Extraction and analysis of Facebook friend lists using the OCR function within CIB-Triage III.

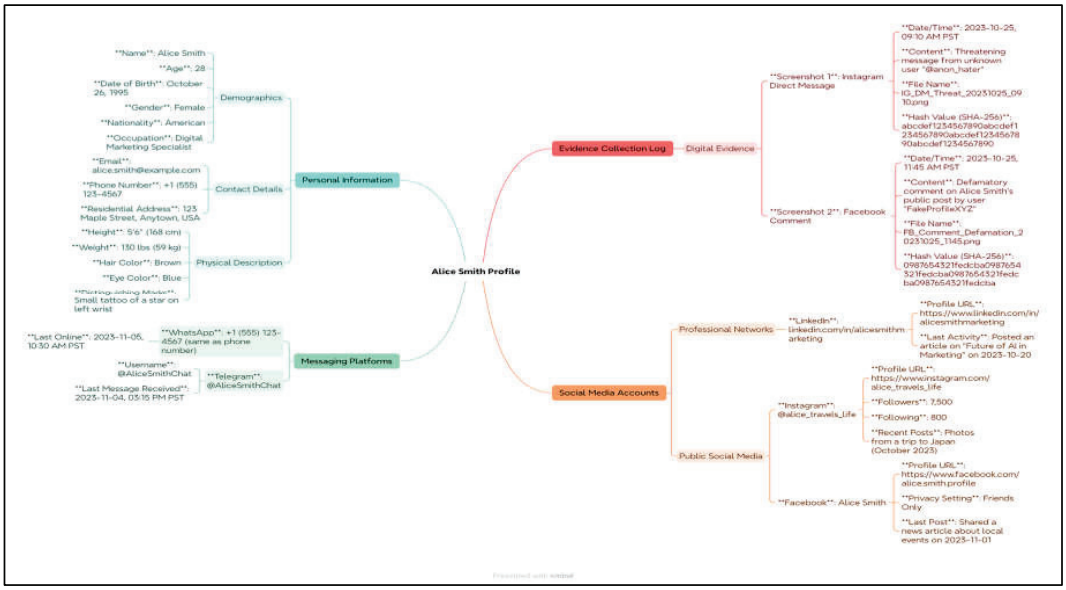


Figure 14. Mind mapping of the investigative target and associated relationships created with Xmind.

After the analytical process is completed, verified datasets are forwarded to the Authentication Phase, where hash values and metadata are generated to ensure evidentiary integrity and forensic verifiability.

4.3. Authentication and Preservation Phase

In the Authentication Phase, CIB-Triage III (Figure 15) automatically generates cryptographic hash values for all captured screenshots and video recordings. These hash values, together with associated metadata such as timestamps and collector identity, are securely stored in b-JADE (Figure 16). This procedure ensures that each piece of evidence can be independently verified during judicial review and remains tamper-proof throughout its lifecycle.

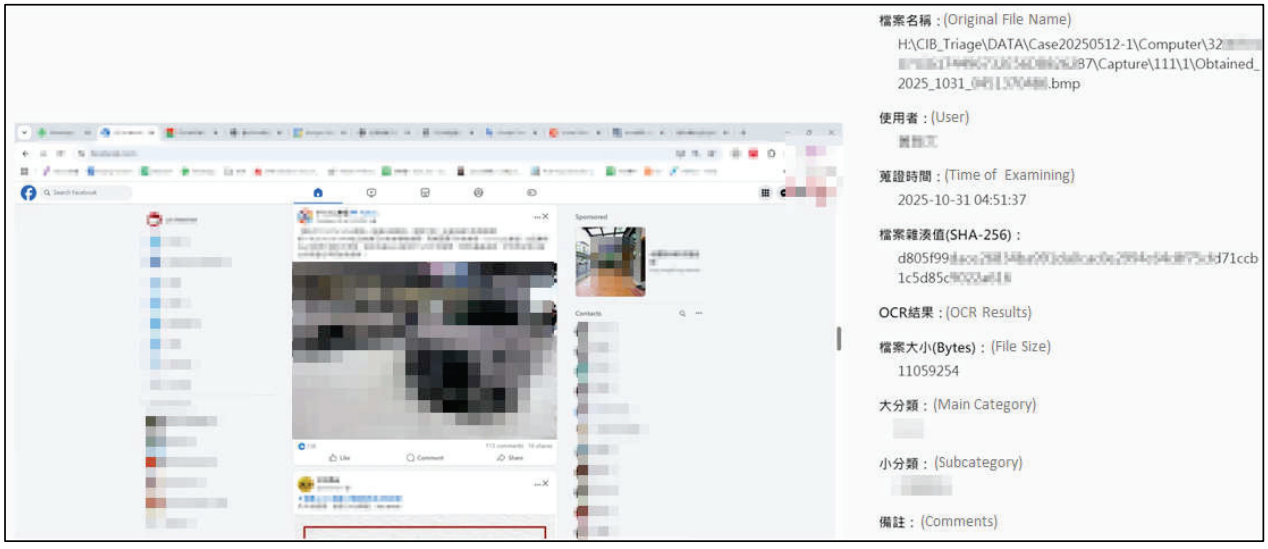


Figure 15. Hash value generation and metadata recording are performed within the CIB-Triage III platform.



Figure 16. Evidence preservation and blockchain-based verification performed through the integration of CIB-Triage III with the b-JADE.

4.4. Validation Phase

This phase is the final operational step of the framework. It consolidates and validates outputs from identification, acquisition, authentication, and preservation. The system performs consistency checks, revalidates evidence integrity, and generates structured exports to support report preparation for judicial or administrative use.

1. Data integrity revalidation

The platform rechecks hash values, timestamps, and blockchain notarization identifiers produced during authentication and preservation. Hash records stored in b-JADE are compared with the local evidence registry to confirm that no alterations occurred during storage or transfer (Figure 17).

2. Procedural traceability review

Operational logs are compiled and linked to their corresponding digital objects. Logs include acquisition timestamps, collector identifiers, analytic actions, and version information. These linkages establish an auditable chain of custody suitable for judicial review.

3. Reporting support and structured export

Validated findings, metadata, and integrity references are organized into exportable fields aligned with Section 3.5 and the Berkeley Protocol. The framework does not autogenerate the final report. Investigators can directly use exported fields, such as key findings and identity correlation evidence, as well as the associated acquisition and authentication metadata, cryptographic hashes, and blockchain transaction IDs. Additionally, they can appendices that list artifact manifests and hash indexes.

Through these validations and structured outputs, the Validation Phase completes the forensic workflow, ensuring that OSINT evidence managed by CIB-Triage III is verifiable, traceable, and suitable for legal admissibility.

4.5. Performance Evaluation and Scalability Analysis

Social media evidence is characterized by high velocity and volatility, which requires the underlying notarization mechanism to respond rapidly and remain stable under varying workloads. To evaluate whether the proposed blockchain-based framework meets these

operational requirements, we conducted a series of performance and scalability experiments focusing on the hash-to-on-chain process in the b-JADE environment.

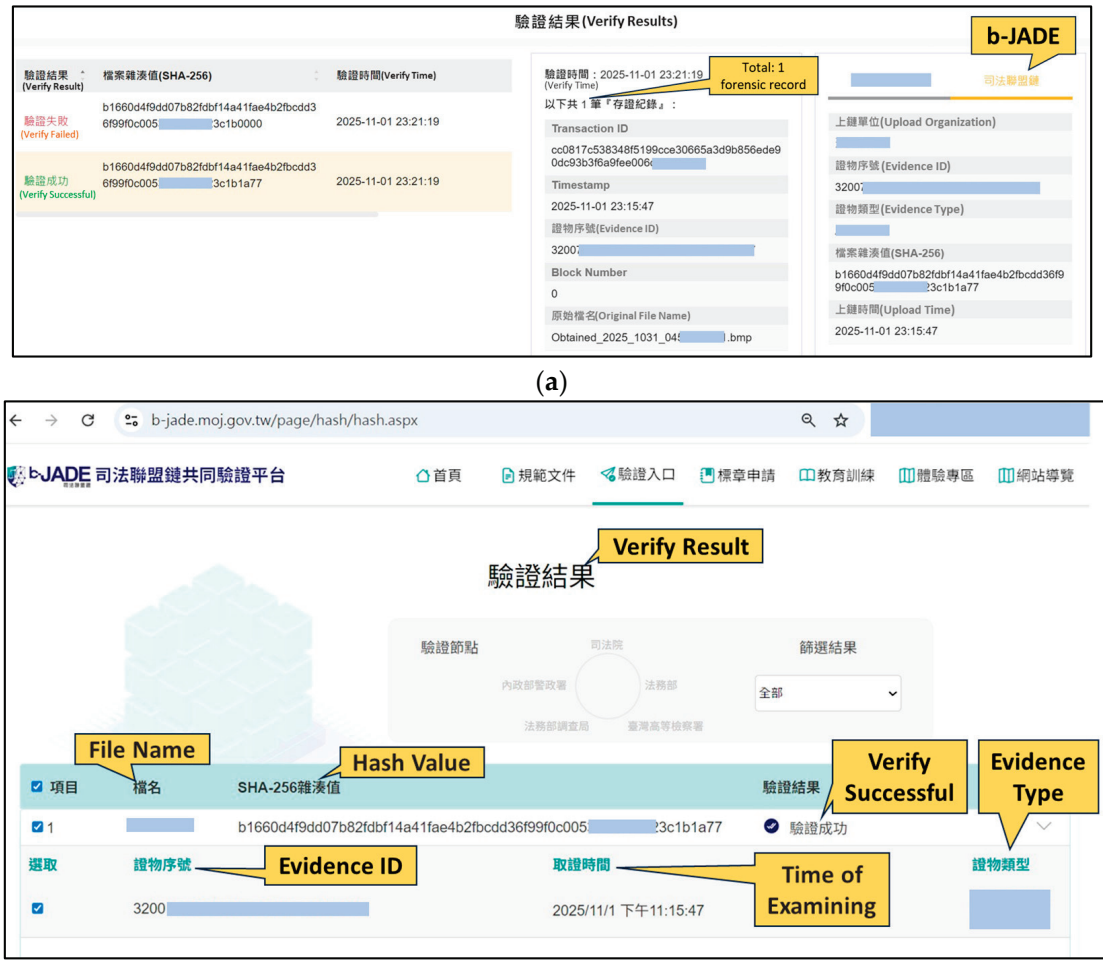


Figure 17. Verification of digital evidence integrity on the b-JADE blockchain. (**a**) Verification performed through the connection between the CIB-Triage III forensic platform and b-JADE. (**b**) Verification conducted via the b-JADE Common Verification Platform web interface (https://b-jade.moj.gov.tw/page/hash/hash.aspx) (accessed on 24 November 2025).

4.5.1. Experimental Setup and Scenarios

The experiments were performed on VMware ESXi virtual machines equipped with 4 virtual CPUs, 8 GB of RAM, and a 120 GB solid-state drive, running Ubuntu 20.04 and Hyperledger Fabric. The blockchain network was configured with a batch timeout of 2 s, a maximum message count of 500 transactions per block, and an absolute maximum block size of 10 megabytes. These parameters reflect a representative configuration for permissioned blockchain deployments in practical forensic environments.

To evaluate the impact of increasing data volume on system latency, we generated a comprehensive set of hash values and conducted write-latency tests across four batch sizes: 10, 100, 1000, and 10,000 entries. Because Hyperledger Fabric does not explicitly record block-finalization timestamps, the completion time for each test was determined using the header timestamp of the final transaction included in the last confirmed block, which provides millisecond-level precision. To ensure reliability and reduce the influence

of outliers, each test scenario was executed 5 times, and the average completion time was used for analysis.

4.5.2. Experimental Results and Analysis

The results summarized in Table 2 demonstrate a clear relationship between batch size and write latency while confirming the operational stability of the system. For small batches of 10 hashes, the average completion time was 42 milliseconds. As the batch size increased to 100 and 1000 hashes, the average latency rose to 2.24 s and 51.25 s, respectively. The largest batch of 10,000 hashes required 555.21 s (approximately 9 min).

Table 2. Test result for 10 to 10,000 hash values.

Test Case	10 Transactions	100 Transactions	1000 Transactions	10,000 Transactions
Case 1	38 ms	2111 ms	49,471 ms	563,705 ms
Case 2	24 ms	2462 ms	51,474 ms	547,638 ms
Case 3	79 ms	2023 ms	53,490 ms	555,367 ms
Case 4	38 ms	2229 ms	51,614 ms	556,447 ms
Case 5	31 ms	2355 ms	50,178 ms	552,896 ms
Average	42 ms	2236 ms	51,245 ms	555,211 ms

These increases in latency reflect the computational overhead inherent to block ordering and validation within a permissioned blockchain network. Nonetheless, the performance remains within acceptable limits for forensic operations. In practical investigative contexts, the ability to notarize thousands of digital artifacts, such as chat logs or image collections, within minutes is sufficient to preserve volatile social media evidence before suspects can alter or delete content. These results demonstrate that the b-JADE infrastructure provides the throughput and stability required for real-world investigative scenarios that demand timely OSINT preservation.

5. Conclusions

This research establishes an integrated digital-forensics framework that transforms OSINT from a loosely organized investigative activity into a legally admissible evidentiary process. By combining standardized identification, controlled acquisition and analysis, cryptographic authentication, and blockchain-based preservation, the proposed model operationalizes the full OSINT lifecycle in accordance with forensic and legal standards.

Through the implementation of the CIB-Triage III prototype, the framework demonstrates that OSINT evidence can be recognized, collected, analyzed, authenticated, and validated within a single coherent workflow. This integration closes the long-standing gap between open-source information gathering and judicially recognized digital evidence. The b-JADE blockchain infrastructure further extends this capability by providing immutable notarization and cross-agency verification, creating an auditable chain of custody that meets the evidentiary requirements of international courts and law enforcement cooperation.

The study's methodological contribution lies in formalizing a replicable, end-to-end process that aligns the principles of the Berkeley Protocol on Digital Open Source Investigations with ISO/IEC 27037 and related standards. Technically, it delivers a prototype that unifies multi-factor identity correlation, integrity verification, and digital preservation within a single forensic platform. Practically, it offers judicial institutions a verifiable path to admit OSINT-derived evidence while maintaining transparency, reproducibility, and ethical compliance.

By consolidating these theoretical, technical, and procedural advances, the framework establishes a new paradigm for OSINT forensics. It provides an actionable model that enhances investigative credibility, strengthens transnational evidentiary reliability, and promotes judicial confidence in the lawful use of open-source intelligence worldwide. Future research should further evaluate whether the deployment of a production-level version of CIB-Triage III can empirically improve the judicial acceptance and evidentiary weight of OSINT in real court proceedings.

While the current framework establishes a robust foundation for data integrity using permissioned blockchains, future iterations will integrate advanced privacy-preserving technologies and intelligent analysis modules. Zero Knowledge Proofs (ZKPs) represent a promising direction for privacy-preserving OSINT, enabling investigators to prove the possession or validity of evidence without disclosing sensitive collection methods or source identities on the ledger. Furthermore, as discussed in Section 2.4, incorporating AI-driven anomaly detection, deepfake detection, and dynamic graph analysis will enhance automated identification of manipulated content and fraudulent on-chain behaviors. Integrating these smart and privacy-preserving analytical components within the b-JADE infrastructure represents a natural evolution of the framework, transforming it from a passive preservation mechanism into an active, intelligent forensic ecosystem.

Author Contributions: All authors discussed the content of the manuscript and contributed equally to its preparation. Specifically, H.-W.H. performed the experiments; C.-H.S. and C.-Y.L. designed and supervised the study; H.-Y.T. analyzed the experimental results; and C.-Y.L. assisted in interpreting the results. C.-H.S. and C.-Y.L. reviewed the manuscript, while C.-Y.L. and H.-W.H. drafted the initial version. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The CIB-Triage III prototype used in this study is not publicly available due to copyright and intellectual property restrictions. No new datasets were generated or analyzed beyond those incorporated within the framework's controlled testing environment.

Acknowledgments: During the preparation of this manuscript, the authors used ChatGPT-5, ChatGPT-4o, and Grammarly (version 1.2.198.1762) to assist with text refinement and language editing through human–machine–human iterations. Furthermore, ChatGPT-5 and Freepik were utilized to optimize the visual aesthetics, specifically the color schemes and structural layout, of Figures 1 and 4 based on the original designs. The authors have carefully reviewed and edited all outputs and take full responsibility for the content of this publication.

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

References

1. Statista. Biggest Social Media Platforms by Users 2025. Available online: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/ (accessed on 17 November 2025).
2. AP News. Capitol Rioters' Social Media Posts Influencing Sentencings. Available online: https://apnews.com/article/media-prisons-social-media-capitol-siege-sentencing-0a60a821ce19635b70681faf86e6526e (accessed on 10 July 2025).
3. Insurance Business, UK. Another Fraudulent Insurance Claim Foiled by Social Media. Available online: https://www.insurancebusinessmag.com/uk/news/breaking-news/another-fraudulent-insurance-claim-foiled-by-social-media-193531.aspx (accessed on 10 July 2025).
4. *ISO/IEC 27037:2012*; Information Technology—Security Techniques—Guidelines for Identification, Collection, Acquisition, and Preservation of Digital Evidence. ISO: Geneva, Switzerland, 2012.
5. *ISO/IEC 27043:2015*; Information Technology—Security Techniques—Guidelines for the Analysis and Interpretation of Digital Evidence. ISO: Geneva, Switzerland, 2015.
6. *ISO/IEC 27041:2015*; Information Technology—Security Techniques—Incident Investigation Principles and Processes. ISO: Geneva, Switzerland, 2015.

7. *ISO/IEC 27042:2015*; Information Technology—Security Techniques—Guidance on Assuring Suitability and Adequacy of Incident Investigative Method. ISO: Geneva, Switzerland, 2015.
8. US Law Rule 901. Authenticating or Identifying Evidence. Available online: https://www.law.cornell.edu/rules/fre/rule_901 (accessed on 14 June 2025).
9. CourtListener. (n.d.). *Search Results for Courts: All Query: Social Media Evidence*. Available online: https://www.courtlistener.com/?type=r&q=social%20media%20evidence&type=r&order_by=score%20desc (accessed on 21 November 2025).
10. Gupta, K.; Oladimeji, D.; Varol, C.; Rasheed, A.; Shahshidhar, N. A Comprehensive Survey on Artifact Recovery from Social Media Platforms: Approaches and Future Research Directions. *Information* **2023**, *14*, 629. [CrossRef]
11. Patzakis, J. Social Media Evidence Key Factor in Estimated 500,000 Litigation Cases Last Year. Available online: https://blog.pagefreezer.com/social-media-evidence-500000-litigation-cases (accessed on 22 November 2025).
12. Wright, P. Neal Ysart Legal Scrutiny of OSINT Evidence and the Access to Digital Devices in Court: Key Case Law and Best Practices. Available online: https://www.linkedin.com/pulse/legal-scrutiny-osint-evidence-access-3gbjf/ (accessed on 22 November 2025).
13. 06-1893—Lorraine et al v. Markel American Insurance Company—Content Details. Available online: https://www.govinfo.gov/app/details/USCOURTS-mdd-1_06-cv-01893 (accessed on 22 November 2025).
14. 06-1497—Capitol Records, Inc et al v. Thomas-Rasset—Document in Context. Available online: https://www.govinfo.gov/app/details/USCOURTS-mnd-0_06-cv-01497/context (accessed on 22 November 2025).
15. People v. Price 2017 New York Court of Appeals Decisions New York Case Law New York Law U.S. Law Justia. Available online: https://law.justia.com/cases/new-york/court-of-appeals/2017/58.html (accessed on 10 July 2025).
16. United States v. Farrad, No. 16-6730 (6th Cir. 2018) Justia. Available online: https://law.justia.com/cases/federal/appellate-courts/ca6/16-6730/16-6730-2018-07-17.html (accessed on 22 November 2025).
17. Can Social Media Be Used in Court? 23 Court Cases That Prove Social Media Evidence Can Make or Break a Case. Available online: https://blog.pagefreezer.com/social-media-digital-evidence-forensics-court-cases (accessed on 22 November 2025).
18. Supreme Court of British Columbia, C.C. ICD—R. v. Hamdan—Asser Institute. Available online: https://www.internationalcrimesdatabase.org/Case/3306/ (accessed on 14 June 2025).
19. Taiwan Supreme Court Judgment No. 320. Available online: https://web.archive.org/web/20251129025708/https://judgment.judicial.gov.tw/FJUD/data.aspx?ty=JD&id=TPSM,112%2c%e5%8f%b0%e4%b8%8a%2c320%2c20230817%2c1 (accessed on 24 November 2025).
20. Basumatary, B.; Kalita, H.K. Social Media Forensics—A Holistic Review. In Proceedings of the 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 23–25 March 2022; pp. 590–597. [CrossRef]
21. Vidya, V.; Saly, K.; Balan, C. Forensic Acquisition and Analysis of Webpage. In Proceedings of the 2022 2nd International Conference on Intelligent Technologies (CONIT), Hubli, India, 24–26 June 2022. [CrossRef]
22. Greco, C.; Ianni, M.; Seminara, G.; Guzzo, A.; Fortino, G. A Forensic Framework for Screen Capture Validation in Legal Contexts. In Proceedings of the 2024 IEEE International Conference on Cyber Security and Resilience (CSR), London, UK, 2–4 September 2024; pp. 127–132. [CrossRef]
23. Guo, H.; Yu, X. A Survey on Blockchain Technology and Its Security. *Blockchain Res. Appl.* **2022**, *3*, 100067. [CrossRef]
24. Liu, S.; Zheng, Q. A Study of a Blockchain-Based Judicial Evidence Preservation Scheme. *Blockchain Res. Appl.* **2024**, *5*, 100192. [CrossRef]
25. Deng, Z.; Ren, Y.; Liu, Y.; Yin, X.; Shen, Z.; Kim, H.J. Blockchain-Based Trusted Electronic Records Preservation in Cloud Storage. *Comput. Mater. Contin.* **2019**, *58*, 135. [CrossRef]
26. Ministry of Justice, Taiwan. Blockchain-Applied Judicial Alliance for Digital Era Common Verification Platform. Available online: https://web.archive.org/web/20250505075608/https://b-jade.moj.gov.tw/Default.aspx (accessed on 24 November 2025).
27. Transaction Flow—Hyperledger Fabric Docs Main Documentation. Available online: https://hyperledger-fabric.readthedocs.io/en/latest/txflow.html (accessed on 10 July 2025).
28. Sheng, Z.; Song, L.; Wang, Y. Dynamic Feature Fusion: Combining Global Graph Structures and Local Semantics for Blockchain Phishing Detection. *IEEE Trans. Netw. Serv. Manag.* **2025**, *22*, 4706–4718. [CrossRef]
29. Tian, Y.; Song, L.; Qian, P.; Wang, Y.; Sun, J.; Jia, Y. DiT-SGCR: Directed Temporal Structural Representation with Global-Cluster Awareness for Ethereum Malicious Account Detection. *arXiv* **2025**, arXiv:2506.20123.
30. Liu, T.; Wang, Y.; Sun, J.; Tian, Y.; Huang, Y.; Xue, T.; Li, P.; Liu, Y. The Role of Transformer Models in Advancing Blockchain Technology: A Systematic Survey. *Eng. Appl. Artif. Intell.* **2026**, *163*, 112968. [CrossRef]
31. Jia, Y.; Wang, Y.; Sun, J.; Tian, Y.; Qian, P. LMAE4Eth: Generalizable and Robust Ethereum Fraud Detection by Exploring Transaction Semantics and Masked Graph Embedding. *IEEE Trans. Inf. Forensics Secur.* **2025**, *20*, 10260–10274. [CrossRef]

32. Sun, J.; Jia, Y.; Wang, Y.; Tian, Y.; Zhang, S. Ethereum Fraud Detection via Joint Transaction Language Model and Graph Representation Learning. *Inf. Fusion* **2025**, *120*, 103074. [CrossRef]
33. Zafarani, R.; Liu, H. Connecting Corresponding Identities across Communities. In Proceedings of the Third International AAAI Conference on Weblogs and Social Media, San Jose, CA, USA, 17–20 May 2009.
34. Wang, Y.; Liu, T.; Tan, Q.; Shi, J.; Guo, L. Identifying Users across Different Sites Using Usernames. *Procedia Comput. Sci.* **2016**, *80*, 376–385. [CrossRef]
35. Li, Y.; Peng, Y.; Ji, W.; Zhang, Z.; Xu, Q. User Identification Based on Display Names Across Online Social Net-works. *IEEE Access* **2017**, *5*, 17342–17353. [CrossRef]
36. Motoyama, M.; Varghese, G. *I Seek You: Searching and Matching Individuals in Social Networks*; ACM Digital Library: New York, NY, USA, 2013; ISBN 9781605588087.
37. Raad, E.; Chbeir, R.; Dipanda, A. User Profile Matching in Social Networks. In Proceedings of the 13th International Conference on Network-Based Information Systems, Takayama, Japan, 14–16 September 2010; pp. 297–304. [CrossRef]
38. Iofciu, T.; Fankhauser, P.; Abel, F.; Bischoff, K. Identifying Users Across Social Tagging Systems. In Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, Barcelona, Spain, 17–21 July 2011.
39. Na, Y.; Yinliang, Z.; Lili, D.; Genqing, B.; Liu, E.; Clapworthy, G.J. User Identification Based on Multiple Attribute Decision Making in Social Networks. *China Commun.* **2013**, *10*, 37–49. [CrossRef]
40. Narayanan, A.; Shmatikov, V. De-Anonymizing Social Networks. *Proc IEEE Symp Secur Priv* **2009**, 173–187. [CrossRef]
41. Cui, Y.; Pei, J.; Tang, G.; Luk, W.-S.; Jiang, D.; Hua, M. Finding Email Correspondents in Online Social Networks. *World Wide Web* **2012**, *16*, 195–218. [CrossRef]
42. Sensity AI: Best Deepfake Detection Software in 2025. Available online: https://sensity.ai/ (accessed on 17 June 2025).
43. Deepfake Detection—Reality Defender. Available online: https://www.realitydefender.com/ (accessed on 17 June 2025).
44. Xmind—Mind Mapping App. Available online: https://xmind.com/ (accessed on 17 June 2025).
45. *ISO/IEC 27001:2022*; Information Security, Cybersecurity and Privacy Protection—Information Security Management Systems—Requirements. ISO: Geneva, Switzerland, 2022.
46. *ISO/IEC 27002:2022*; Information Security, Cybersecurity and Privacy Protection—Information Security Controls. ISO: Geneva, Switzerland, 2022.
47. United Nations Office of the High Commissioner for Human Rights (OHCHR) and Human Rights Center, University of California, Berkeley School of Law. The Berkeley Protocol on Digital Open Source Investigations: A New Standard for Online Investigations. Office of the High Commissioner for Human Rights, Geneva, Switzerland, 2020. Available online: https://www.ohchr.org/en/publications/policy-and-methodological-publications/berkeley-protocol-digital-open-source (accessed on 24 November 2025).

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

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

Future Internet Editorial Office
E-mail: futureinternet@mdpi.com
www.mdpi.com/journal/futureinternet



Disclaimer/Publisher's Note: The title and front matter of this reprint are at the discretion of the Guest Editors. The publisher is not responsible for their content or any associated concerns. The statements, opinions and data contained in all individual articles are solely those of the individual Editors and contributors and not of MDPI. MDPI disclaims responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.





Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-8976-1