Independent University **Bangladesh (IUB)**

IUB Academic Repository

Computer Science and Engineering | Undergraduate Thesis

2023-01

Blockchain-Based Intrusion Detection Systems (IDS) using Decentralized Threat Intelligence Sharing

Hasan, Mostafa

IUB

http://ar.iub.edu.bd/handle/11348/988

Downloaded from IUB Academic Repository



Blockchain-Based Intrusion Detection Systems (IDS) using Decentralized Threat Intelligence Sharing

January 2023

Prepared by:

Mostafa Hasan
ID: 1910800

Moinul Alam
ID: 2120306

Arvil Nath Akash
ID: 2120263

Department of Computer Science and Engineering
Independent University, Bangladesh

Supervised by

Sanzar Adnan Alam
Sr.Lecturer, Department of CSE
Department of Computer Science and Engineering
Independent University, Bangladesh

Attestation

We are aware of the fact that plagiarism is strictly prohibited and it conflicts with our university's rules and regulations. We guarantee the authenticity of our work. We have used some copyrighted material and models of others in our project which has been properly cited following the international standards proper guidelines.

Author Name:

. . . . . .

Signature:

. . . . . .

Author Name:

. . . . . .

Signature:

. . . . . .

Author Name:

. . . . . .

Signature:

. . . . . .

Evaluation Committee

Supervisor

Name: _____ Signature: _____

Internal Examiner 1

Name: _____ Signature: _____

Internal Examiner 2

Name: _____ Signature: _____

External Examiner

Name: _____ Signature: _____

Acknowledgement

We would like to express sincere gratitude to our respected faculty, Sanzar Adnan Alam, Senior Lecturer, Department of Computer Science and Engineering, Independent University, Bangladesh. His dedicated guidance, insightful suggestions, remarkable patience, and generous investment of his valuable time played a pivotal role in the successful completion of our project. Our Professor's expertise and mentorship greatly enhanced the quality and depth of our work, and we are truly appreciative of his unwavering support throughout our academic journey.

Our heartfelt thanks go to the CCDS Lab (Center for Computational & Data Sciences) for their extraordinary help and consistent support. The collaborative and resourceful environment provided by CCDS Lab significantly contributed to the development and execution of our project. Moreover, The encouragement and assistance have been instrumental in shaping our academic and research field. We are grateful for the opportunities and resources extended to us during our research, which undoubtedly enriched our learning experience.

We acknowledge and appreciate the contributions of all individuals who directly and indirectly supported us during this endeavor.

Abstract

This paper presents a theoretical analysis of a blockchain-based Intrusion Detection System (IDS) designed to improve cybersecurity through decentralized threat intelligence sharing. The proposed IDS framework integrates the Isolation Forest algorithm, a machine learning model, with blockchain technology to enable anomaly detection and automated threat response across a network. Using the Isolation Forest algorithm, the system effectively identifies anomalies in network traffic, while smart contracts on the blockchain allow for autonomous actions, such as node quarantine and malicious IP blocking. This approach aims to reduce the frequency of false positives and enhance response times, providing an efficient solution for scalable and collaborative cybersecurity applications. By examining the the potential of combining machine learning with blockchain and smart contracts, this paper contributes to the advancement of adaptive IDS solutions that leverage decentralized architectures for enhanced threat detection and rapid response.

Contents

1 Introduction **8**
- 1.1 Background of the project . . . 8
- 1.2 Problem Statement . . . 9
- 1.3 Outcome of the project . . . 9

2 Literature Review **11**
- 2.1 Existing Methods: . . . 11
- 2.2 Gaps within the Existing Methods: . . . 12

3 Theoretical Framework **14**
- 3.1 Decentralized Threat Intelligence Sharing . . . 14
- 3.2 Anomaly Detection with the Isolation Forest Algorithm . . . 15
- 3.3 Automated Response with Smart Contracts . . . 15
- 3.4 Synergistic Integration of Technologies . . . 15

4 Methodology **17**
- 4.1 Research design and Approach . . . 17
- 4.2 Data Pre-processing Steps . . . 19
  - 4.2.1 Removing Duplicates . . . 19
  - 4.2.2 Normalization . . . 19
  - 4.2.3 Imputing Missing Values . . . 20
- 4.3 Implementation of the Isolation Forest Algorithm . . . 20
- 4.4 Blockchain Architecture and Consensus Mechanism . . . 21
  - 4.4.1 Blockchain Design and Role . . . 22
  - 4.4.2 Consensus Mechanism . . . 23
  - 4.4.3 Scalability and Fault Tolerance . . . 24
- 4.5 Smart Contract Design and Functionality . . . 24
  - 4.5.1 Design Principles . . . 25
  - 4.5.2 Functional Workflow . . . 25
  - 4.5.3 Pseudocode for Smart Contracts . . . 25
  - 4.5.4 Benefits and Challenges . . . 26
  - 4.5.5 Integration with the System . . . 27

5 Performance Evaluation **28**
- 5.1 Metrics Used . . . 28
  - 5.1.1 Detection Accuracy . . . 28
  - 5.1.2 Precision . . . 29
  - 5.1.3 False Positive Rate . . . 29
  - 5.1.4 Response Time . . . 29
- 5.2 Scalability and Fault Tolerance of the Proposed IDS . . . 30

5.2.1 Scalability . . . 30
5.2.2 Fault Tolerance . . . 31
5.3 Theoretical Analysis of System Robustness . . . 32

6 Discussion **34**
6.1 Advantages of decentralized threat intelligence sharing . . . 34
6.2 Machine learning's role in improving anomaly detection . . . 35
6.3 Limitations and trade-offs in performance. . . . 36

7 Conclusion and Future Works **38**
7.1 Conclusion . . . 38
7.2 Proposed advancements . . . 39
7.2.1 Deep learning . . . 39
7.2.2 Layer-2 scaling . . . 39
7.2.3 Hybrid storage . . . 40

8 Submission History **41**

Bibliography **41**

List of Figures

1 Architecture of the Proposed Blockchain-Based Intrusion Detection System (IDS) 18
2 Process diagram of Isolation Forest algorithm 21
3 Functional diagram of Smart Contract 26
4 Scalability and Fault Tolerance Flowchart 30

1 Introduction

1.1 Background of the project

As we move into an era of pervasive interconnectivity, cybersecurity threats are growing ever more complex and widespread. However, this rapid evolution of threats precludes traditional Intrusion Detection Systems (IDS) from keeping pace with the dynamic behavior of contemporary cyberattacks. Most conventional IDS frameworks rely on centralized architectures to monitor and detect malicious activities. While this approach has worked to an extent, this approach is inherently vulnerable, especially against large-scale Distributed Denial of Service (DDoS) attacks and in scaling [1]. Moreover, these systems, having a centralized nature, possess a single point of failure which in case of a breach can result in a disastrous consequence, that is, they can expose sensitive data and disrupt an entire network infrastructure. The limitations emphasize the necessity of more sophisticated, and decentralized, systems capable of reacting on the fly to evolving threats.

Lacking to detect novel and advanced attacks, centralized IDS solutions have numerous limitations. However, these systems rely on static, rule-based detection methods, which are incapable of handling emerging attack patterns. In addition, they produce a large number of false positives, consuming resources and the chance of missing actual threats. Their inefficiencies coupled with the latency in threat detection and response only make matters worse, and leave networks vulnerable during critical periods. With the increasing complexity and frequency of cyberattacks, cybersecurity research is focusing particularly on these challenges.

Blockchain technology appears to provide a promising path forward in the integration of blockchain technology into IDS frameworks. Supplying a decentralized, secure, and immutable layer for storing data and threat intelligence, blockchain can make IDS architectures more resilient. As IDS solutions are based on distributed ledger technology, single points of failure are entirely eliminated resulting in enhanced system reliability and scalability. Blockchain also facilitates real-time sharing and collaboration of the threat information between network nodes and fosters a collaborative and cohesive defense mechanism against cyber threats.

Although machine learning (ML), especially the Isolation Forest model [4], could be incorporated into the IDS framework to improve detection accuracy and reduce false positives, blockchain technology complements it. The Isolation Forest algorithm works well for detecting unusual traffic patterns in large-scale networks because it was designed for such things. It lets you monitor network traffic in a decentralized and optimal manner if deployed at individual network nodes. The ML and blockchain combination overcomes the core weaknesses of conventional IDS solutions by providing higher detection and faster response rates.

Furthermore, the IDS framework is fortified by smart contracts: automating response actions. In real time these contracts can execute predefined tasks such as isolating any compromised nodes or limiting malicious IP addresses. The use of this automation re-

duces the response latency between detection and taking countermeasures, aiming to minimize the damages. By leveraging machine learning, blockchain, and smart contracts, synergy is formed between the above which is a robust and scalable framework for solving cybersecurity issues of today.

1.2 Problem Statement

Due to the increasing sophistication and frequency of cyber attacks, traditional Intrusion Detection Systems (IDS) cannot cope with this challenge. Due to the inherent shortcomings of such centralized IDS frameworks such as single points of failure, static rule-based detection methods, and high false positive rates, they are ill-equipped to address these evolving threats. These deficiencies make centralized systems unable to detect novel attack patterns and lag response times, and networks are vulnerable at moments that demand response. These challenges will be addressed through emerging technologies such as blockchain and machine learning, but their integration into IDS frameworks has little been explored. Decentralized approaches are often handicapped by their difficulty in achieving high accuracy of detection at scale in a timely fashion without excessive false alarms. Moreover, most current solutions suffer from automated response mechanisms and effectively compromise the effectiveness of cyber threats mitigating.

Gaps in that research are bridged with the development of a decentralized IDS architecture integrating blockchain and Helix, the Isolation Forest anomaly detection algorithm, and the sharing of distributed threat intelligence. The proposed framework extends detection accuracy, decreases latency, and offers a scalable solution to modern cybersecurity challenges through smart contract-based automated response. The work addresses critical questions in current IDS design and develops a foundation for future decentralized cybersecurity practices.

1.3 Outcome of the project

The Intrusion Detection System (IDS) architecture proposed here seamlessly integrates machine learning and blockchain technologies into the decentralized framework for adaptive, efficient, and cost-effective cybersecurity. The system allows for real-time anomaly detection at individual nodes, bypassing the need for the aid of a central authority, by deploying the Isolation Forest algorithm at those nodes. An independent detection unit runs at each node, securely examining local traffic and sharing anomaly data across a blockchain network. By making this inherently decentralized approach, scalability and resilience are increased and the risk of having a single point of failure is reduced.

With blockchain technology, threat intelligence can be shared securely among nodes, on a secure and immutable platform. Through a decentralized ledger, all nodes are ensured to derive collective insights and improve dynamic and adaptive cybersecurity. Through smart contracts integrated into the blockchain, the system is also empowered by automated threat response actions. Stated differently, smart contracts can isolate compromised nodes, restrict access to malicious IP addresses, or implement any other pre-defined countermeasures in real time when they detect or sense a breach. However, this automation significantly reduces the response latency, minimizing the potential damages, and deploying mitigation against the threats as quickly as possible. The theoretical outcomes of this framework include:

- **Improved Detection Accuracy:** By leveraging the Isolation Forest algorithm for anomaly detection, the system can effectively identify unusual traffic patterns and emerging threats while reducing false positives.
- **Enhanced Scalability and Resilience:** The decentralized nature of the system eliminates dependency on a central authority, allowing it to scale efficiently and remain operational even under large-scale attacks such as Distributed Denial of Service (DDoS).
- **Reduced Response Latency:** Smart contracts automate threat response actions, enabling near-instantaneous execution of mitigation strategies and ensuring timely protection against cyberattacks.
- **Dynamic and Adaptive Cybersecurity:** The system's collective intelligence, supported by blockchain's secure data-sharing capabilities, allows for continuous learning and adaptation to evolving threats.
- **Comprehensive Threat Management:** The integration of machine learning, blockchain, and automated response mechanisms creates a holistic framework capable of addressing modern cybersecurity challenges.

This innovative architecture has the potential to revolutionize how networks defend against cyber threats, providing a scalable and efficient solution for enhancing cybersecurity in increasingly interconnected environments.

2 Literature Review

Recent research highlights the potential of blockchain and machine learning for enhancing IDS frameworks:

2.1 Existing Methods:

- As demonstrated by Al-E'Mari et al. Blockchain transforms IDSs by providing higher precision and reliability by leveraging the advanced detection model of the K-means algorithm to reduce false positives and increase true positives. BAD and ADvISE use the blockchain's tamper-proof property to share attack signatures and reduce false alarms by propagating attack signatures through the blockchain so as to accurately detect repetitive patterns across hosts. DPoS mechanisms and time-triggered patterns improve detection reliability and reduce errors [1].

- This work is primarily concerned with outlier elimination using the Isolation Forest algorithm for achieving higher classification accuracy in cybersecurity environments. We use this approach which pre-processes the data to normalize features and remove noisy instances using recursive feature elimination and isolation mechanisms. We introduce iForestCAD, a novel method for outlier detection in the presence of fraud on mixed attribute data which relies on conditional anomaly scores. An anomalous user detection in Enterprise datasets using an enhanced version of the Isolation Forest algorithm [4].

- To help intrusion data become immutable and decentralized storage, Blockchain technology increasingly finds use to help augment IDS. Although overhead cost remains a concern, this ensures the integrity and traceability of the data. In this work, a hybrid intrusion detection system (HIDT) that uses machine learning (ML) algorithms such as support vector machines (SVM), random forests (RF), and k-nearest neighbors (KNN) for anomaly detection is designed [5].

- This paper combines the power of Ethereum for smart contracts with decentralized storage tools like IPFS to automate and make transparent document share and version control. For example, existing methods like Sweden's blockchain-based land registry system use blockchain for secure document management [10].

2.2 Gaps within the Existing Methods:

While existing methods showcase innovative approaches, several challenges persist:

- We identify the limitations of existing blockchain-based methods for document management and version control such as those based on Ethereum and IPFS for real-time detection tasks. Consensus mechanisms that lead to high latency limit the adoption of these systems in immediate action scenarios. Real-time responsiveness is limited by scalability issues, as more participants or more transactions lead to a loss of scalability. Also, they are not well integrated with AI/ML algorithms which don't allow them to detect and respond to anomalies dynamically. While real-time detection requires efficient management of highly frequent data, current methods essentially handle static data. The problem also has high resource consumption and privacy concerns, particularly in sensitive domains, such as healthcare. In addition, the lack of domain-specific implementation restricts them to use in limited dynamic environments such as IoT and industrial automation. To fill these gaps blockchain's potential for application-specific real-time, decentralized, and secure detection could be enhanced. However, existing blockchain-based methods using Ethereum and IPFS suffer from significant scaling bottlenecks while being applied to real-time detection scenarios. As the number of participants and transactions grows, these systems start experiencing poor performance, with slower consensus and larger latency. Consensus mechanisms like Proof of Work increase these problems because of their high computational and energy demands and are thus impractical for real-time applications that require immediate action. Additionally, the efficiency of a blockchain system is concerned with scalability, first of all, it manages static data superbly but lags in updating dynamic data, especially when it is updated in real-time. However, by restricting their usability to applications with low throughput demands, such as IoT networks or industrial monitoring systems, this limitation limits the usefulness of the proposed solutions for applications with high throughput. To make blockchain technology suitable for real-time detection, scalability cannot be ignored as it can be used to handle bigger networks, faster transaction processing, and real-time modification in data without compromising performance or security [5].

- The research gap in this paper [10] is that prevailing centralized and decentralized document version control systems have problems of security risks, trust breakdown, and low efficiency. Currently, centralized systems have the inherent disadvantage of single control, manipulation, and opaqueness, whereas conventional decentralized systems do not guarantee immutability and consensus approaches for approval. Further, there are no detailed realizations regarding the use of blockchain for establishing a secure, immutable, distributed system for managing documents and versions of documents. This paper fills this gap by presenting a blockchain solution based on Ethereum Smart Contracts and IPFS to bring an effective and novel solution to address the challenges of multi-user document collaborations and versioning, along with decentralization, trust, and immutability.

- Today, agricultural processes do not assign a comprehensive approach to the diagnostics of product quality in real-time and dissemination of information across their logistic chains. Current methods are either expensive or depend heavily on centralized systems and do not fully incorporate protection measures such as anti-tampering during data acquisition. There are still numerous systems that employ blockchain but lack time-bound IA monitoring, decentralized storage, or SC solutions like QR code traceability [8]. Besides, there are often no effective automated data entry mechanisms. Also, there is no reliable reputation scoring for producers. Such gaps call for developing a system that will efficiently augment agricultural transparency and data credibility cost-effectively without compromising the accessibility of scale farmers.

- Real-world detection applications require versatility that is available but is not yet in the currently deployed blockchain-based systems. They are tuned for immutable, static data management, and it is hard to incorporate this in dynamic environments in which data changes rapidly [9]. Consensus mechanisms are also rigid and are unable to adapt to diverse real-time scenarios like introducing new participants or maintaining efficient workloads. At the same time, current methods are not modular or adaptable, imputing a lack of deployment in various domains such as IoT and healthcare, as well as industrial monitoring. Yet, hybrid architectures (combining blockchain with traditional systems or AI-driven analytics) are not supported, further reducing their application in dynamic use cases. To address these limitations, a framework that provides additional flexibility through adaptable frameworks, seamless integration with external systems, and a high degree of dynamic configuration capabilities is necessary. This would allow blockchain solutions to better solve diverse and evolving real-time detection problems at hand.

Our framework advances these studies by utilizing the Isolation Forest algorithm for unsupervised anomaly detection and incorporating smart contracts for automated threat responses.

3 Theoretical Framework

This thesis builds upon a theoretical framework that joins blockchain technology with machine learning algorithms to develop a decentralized Intrusion Detection System (IDS). This architecture addresses the shortcomings of a traditional centralized IDS architecture prone to single points of failure, inefficiently handling incoming threats, and is scalable. The proposed system combines the machine learning algorithm Isolation Forest for anomaly detection and the blockchain's centralized and indestructible ledger. The combination of these technologies defines a robust, scalable, and adaptive framework for the real-time detection and response to cyber threats.

3.1 Decentralized Threat Intelligence Sharing

Typically, traditional IDS systems depend on centralized servers for monitoring network traffic and for identifying anomalies [10]. This is effective in some contexts, and however, it creates significant vulnerabilities. As an example, if the central server gets compromised, the whole system can also fail. Additionally, centralized systems can find it difficult to efficiently scale in the face of growth in networks resulting in data processing and threat response bottlenecks. To address these challenges, a proposed framework decentralizes threat intelligence sharing on blockchain technology. Blockchain stands as a distributed ledger that is always kept synchronized and tamper-proof between all nodes in the network. The chain can be used for each node to independently detect threats and share findings with its other nodes. With this design, there is no longer a need to rely on a central authority to scale or even tolerate failures. Key features of blockchain relevant to this framework include:

- **Immutability:** It does record the detected threat and what actions were taken, they log on to the blockchain so nobody can tamper with or delete historical data.
- **Transparency:** The same data is available to all nodes which creates a foundation for trust and collaboration among the network.
- **Consensus Mechanisms:** For validity, a quick and efficient way to confirm transactions is algorithms like Proof of Authority (PoA) or Delegated Proof of Stake (DPoS) that do it quickly and efficiently with less delay.

Together, these features provide the blockchain component, the real-time, secure sharing of threat intelligence, which allows nodes to respond more accurately and quickly to emerging cyber threats.

3.2 Anomaly Detection with the Isolation Forest Algorithm

The proposed IDS framework depends on the Isolation Forest algorithm as a critical component. It is an unsupervised machine learning model that fills the data space in feature space to isolate anomalies [11]. Isolation Forest is different than traditional clustering methodology which utilizes density or proximity metrics to isolate anomalies by creating random partitions in the data. Rare and distinct (anomalous) data points are less difficult to separate from normal points of data. The algorithm's key advantages include:

- **Efficiency:** For large-scale numbers of network links and high traffic, its computational efficiency is suited.
- **Independence from Labeled Data:** Unlike other methods, it is an unsupervised method that does not always rely on pre-labeled datasets in order to discover known or previously unknown attack patterns.
- **Scalability:** It can be deployed over individual nodes on the network for distributed anomaly detection.

In the proposed framework, each node runs an Isolation Forest algorithm instance that analyzes local network traffic in real time. Those anomalies are detected and shared with other nodes in the blockchain. By using a distributed approach to this, we know that even if one of the nodes gets compromised or goes down the system keeps on running.

3.3 Automated Response with Smart Contracts

The use of blockchain-based smart contracts to automate threat response is a unique feature of this IDS framework. Self-executing programs stored on the blockchain which run as soon as predefined conditions are met are called smart contracts. Within the proposed IDS, smart contracts provide a mechanism for reactive responses to detected threats, reducing the delay of user responses.

- **Node Quarantine:** Prevention of an activity from spreading in a network by isolating compromised nodes is called.
- **Traffic Blocking:** Denying traffic to IP addresses whose detected anomalies are so severe, that traffic is denied to it.
- **Alerts:** Notifying system administrators of automatically recorded threats.

This automation greatly speeds up the system's response time and prevents the danger of widespread damage by threats. Furthermore, smart contracts employ the guarantee that Reflected Actions are consistent and verifiable, in that all Reflected Actions executed are recorded on the blockchain.

3.4 Synergistic Integration of Technologies

When blockchain and machine learning are integrated it becomes a synergistic setup that takes advantage of the good sides of each technology[12]. A Proof of Concept implementation is made using blockchain to provide a secure, decentralized platform for data

sharing and response coordination, and using the Isolation Forest algorithm to perform accurate and efficient detection of anomalies. Several critical challenges in IDS design are addressed by this synergy.

- **Reducing False Positives:** The system can improve the likelihood of false positives by cross-referencing the detected anomalies with data coming from other nodes.
- **Enhancing Scalability:** This is due to the distributed nature of the framework which grows naturally with the network
- **Improving Resilience:** The decentralized architecture ensures that the system remains operational even if individual nodes are compromised.

In this thesis, we present a theoretical framework for a decentralized, scalable, and adaptive IDS. The proposed system integrates Blockchain technology with the Isolation Forest algorithm and smart contracts to resolve critical problems of the traditional IDS architectures. Beyond simply improving the accuracy and efficiency of threat detection, this framework enables real-time, automated responses to further make our network infrastructures more resilient and secure. Moreover, future research should tackle the issues of the system's performance optimization and expand the applicability of the system to many cybersecurity contexts at hand.

4 Methodology

The methodology used for designing and implementing a blockchain-based intrusion detection system (IDS) coupled with the Isolation Forest machine learning algorithm is reported in this chapter. This includes blockchain architecture, smart contract design, research design and approach, data pre-processing, and algorithm implementation.

4.1 Research design and Approach

A decentralized intrusion detection system (IDS) using blockchain technology and machine learning is proposed in the research design to replace the centralized ones and address their limitations. A key goal of the design we present is the construction of a system capable of efficiently detecting anomalies in network traffic and autonomously responding to these threats with high accuracy and low response times.

- **Centralized vs. Decentralized Approaches:** Generally speaking traditional IDS frameworks typically depend on centrally tuned architectures for monitoring and reactions to network threats. Even though these systems can handle particular types of attacks well, there are serious drawbacks: a single-point failure, limited scalability, and slow response. These problems become critical as networks get larger and cyber threats become more sophisticated. Decentralized systems spread the work of data processing and threat intelligence across multiple nodes, leaving them entirely dependent on one server. It helps both the IDS' resilience and scalability.

- **Blockchain Integration for Threat Intelligence Sharing:** The proposed system is based on blockchain technology. It leverages threat intelligence as a decentralized ledger securely recording and sharing threat intelligence in all network nodes. Blockchain differentiates itself from centralized systems where an attacker can compromise the entire network simply by breaking the central server but data immutability and security are provided in blockchain using cryptographic techniques [**?**]. The network allows for dynamic and real-time collaboration in identifying and mitigating threats because each node has the ledger exposed to the network to collaborate and transactions.

  For instance, a node detects an anomaly in the local network traffic of its node and records the anomaly details, i.e., severity level, affected IP address, and timestamp in the blockchain. A consensus mechanism is used to validate this information and add it to the distributed ledger to ensure that all nodes in the network have synchronized and reliable threat intelligence. Under this collaboration model, nodes can enhance by means of shared insights and cut down on detection delays and false positives.

- **Machine Learning for Anomaly Detection:** The detection component of the IDS made use of machine learning critically. An unsupervised machine learning

model, the Isolation Forest algorithm is implemented at each node to analyze network traffic and detect anomalies. It can operate efficiently in distributed environments, and is particularly well suited for decentralized environments: it does not need labeled training data [13] [14]. An Isolation Forest model is trained, at each node, independently on locally observed network traffic and can be used for anomaly detection tailored to each node's environment. The algorithm isolates the anomalies by partitioning data points randomly and it measures the number of splits needed to isolate a given point. Anomalies are sparse and highly discriminative, so they are isolated with fewer splits, having higher anomaly scores. In the decentralized design, each node works independently of a central processor and with high scalability in the detection process.

- **Automated Responses with Smart Contracts:** The integration of smart contracts into Blockchain is leveraged to detect threats in real time. Self-executing programs that automatically execute predefined security actions to find anomalies are smart contracts. Isolation of compromised nodes and blocking malicious IP as well as the notification of those who have been affected by the incident are one of these actions. The system reduces this time for two reasons: first, the system automates these responses, eliminating the delays of manual intervention, and second, early response is faster if the threat is automated instead of having to wait for someone to respond. As an example, if the Isolation Forest algorithm flags an anomaly, a smart contract that handles the anomaly is triggered. The contract's logic guarantees that the response is proportional to the severity of the threat, and thus can provide for dynamic fine-tuning of the treatment of minor vs critical anomalies. On top of that, the smart contract takes action which is recorded on the blockchain for future audit and system improvement.

- **Proposed IDS Architecture:** These components are combined in the architecture of the proposed system. For anomaly detection, we take the approach that each node of the network runs its own Isolation Forest model, thus functioning as an autonomous unit. The incoming anomalies are validated and stored on the blockchain in which they are broadcasted if they are detected. In dynamic cybersecurity environments, smart contracts are integrated into the blockchain in order to automate responses, making the system robust and adaptable.

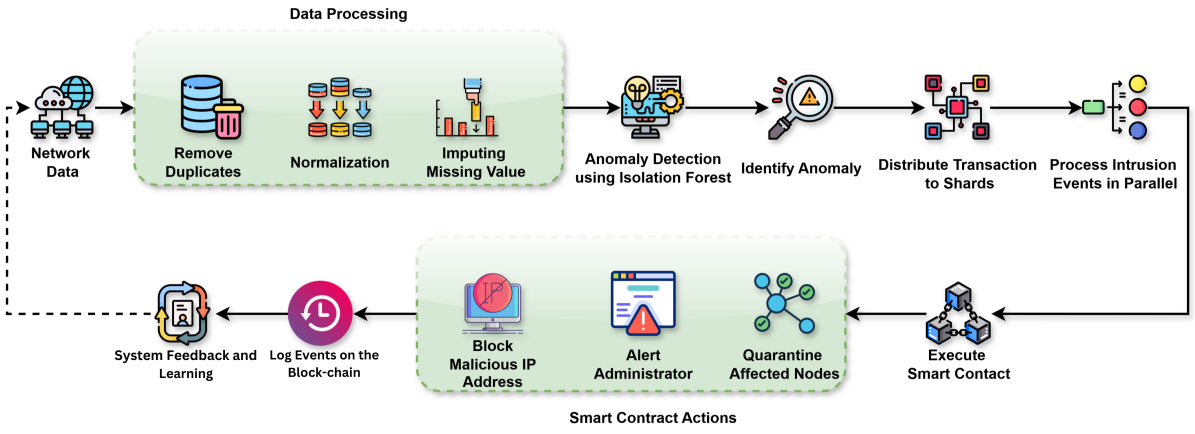

Figure 1: Architecture of the Proposed Blockchain-Based Intrusion Detection System (IDS)

4.2 Data Pre-processing Steps

The proposed Intrusion Detection System's effectiveness and accuracy depend greatly on data pre-processing. Before Isolation Forest can process the data to detect anomalies, the raw network traffic data has to be cleaned, normalized, and prepared. However, the model's performance typically degrades as a result of noise, inconsistency, or incomplete data, without proper pre-processing. In this system, pre-processing consists of three essential steps: This was in fact removing duplicates, normalizing, and imputing missing values.

4.2.1 Removing Duplicates

The first step, finding and deleting duplicate records from the network traffic data directory is required. Repetitive entries can be derived from various types, including logs produced multiple times by network monitoring programs, and redundant data packets ingress at transmission time. However, these duplicates can skew the learning process of the Isolation Forest algorithm, biasing or misleading the anomaly detection. This system ensures that each data point represents unique network activity in order to lower the risk of over-fitting and thus improve the system's ability to generalize to different traffic patterns.

Typically, a duplicate removal process strikes through a dataset and scans with identical values across key attributes like source IP address, destination IP address, packet size, protocol type, and timestamp. Duplicates are returned if any are found, and if found they are only retained and the rest are thrown away. That way, the dataset is concise, representative, and redundancy-free.

4.2.2 Normalization

The transformation of network traffic data into a consistent scale is referred to as normalization. Features such as packet size and transmission duration in raw form can have wide-ranging values, with few values dominating others simply by their magnitude. For example, the packet can have variable sizes such as less than a few bytes, up to several kilobytes and the transmission duration can vary from milliseconds to seconds. These variations can mask the feature importance to the machine learning model.

In order to address this issue, we normalize all numerical features relative to a common range, say between 0 and 1, or using z-score standardization. This scaling ensures that no single feature will overly impact the anomaly detection process. For instance, the packet size of a traffic record is scaled using the formula:

$$X_{\text{scaled}} = \frac{X - \mu}{\sigma}$$

This is where we used the original value X, the mean of the corresponding feature μ, and the standard deviation σ. This transformation makes it equally treatable for all the features by the Isolation Forest algorithm and hence has more robust anomaly detection results.

4.2.3 Imputing Missing Values

Missing values are often present in network traffic data because these contain; incomplete packet captures, logging errors, or incomplete metadata collection. Such gaps in data tend to impede the Isolation Forest algorithm in identifying anomalies because the algorithm relies on complete and consistent input to achieve effective pattern establishing and isolation of unusual behavior.

Statistical imputation techniques are applied to handle missing values. For categorical (or numerical) features, missing values can be plugged with the mean, median, mode, or any other value on the feature (as long as this value will not be used for imputation). In categorical features, the most common value is usually substituted with. For instance, if there is no protocol type in the record, we can infer the most common protocol used in the dataset which is TCP. This makes it a guarantee that this dataset can be consistently and truthfully learned during learning, reducing the odds of introducing bias in the learning Bias.

4.3 Implementation of the Isolation Forest Algorithm

We propose a novel Intrusion Detection System (IDS) with the Isolation Forest algorithm, a core component, to detect anomalies in network traffic. The Isolation Forest is unlike (and better than) traditional clustering or density-based anomaly detection methods, it is designed for unsupervised anomaly detection. It recursively partitions data points in a feature space isolating anomalies based on their rarity and distinctness native property. Thus, the algorithm is ideally suited for identifying unusual patterns in network traffic without supervision.

The implementation starts with a training phase where the model is trained using network traffic data delivered locally at each node. When passed through each node, it acts as an independent detection unit, continuously monitoring traffic and looking for features like source IP addresses, packet sizes, and protocol types [13]. The Isolation Forest algorithm takes the form of the following: It randomly selects features, and split points, and creates binary trees, thus partitioning the data. Anomalies are rare and differ from normal data significantly such that they appear nearer to the root, i.e., require fewer splits to isolate.

In the detection phase, the trained Isolation Forest model computes at runtime incoming network traffic. We assign each data point an anomaly score, which is the time it takes for an anomaly point to be isolated within the forest of binary trees. Anomalies with fewer splits can be isolated as points, and they get higher anomaly scores that indicate they're more likely to be anomalies. How do we know if a data point is anomalous, we use a predefined threshold. If the anomaly score exceeds a threshold value, the anomaly score indicates one or more potential intrusions. Fig. **??** illustrates the steps involved in the algorithm, including random feature selection, anomaly score calculation, and anomaly classification. This visual representation helps in understanding how anomalies are isolated and classified during runtime.

Intrusion Detection System (IDS) is a core architecture of the proposed method and the Isolation Forest algorithm is one of the core elements of the IDS, which is used for detecting the anomalies in the network traffic. The Isolation Forest is designed to detect anomalies in a way that traditional clustering and density-based anomaly detection methods do not. It partitions recursively in a feature space and isolates anomalies that are inherently rare and distinct. The algorithm is thus well suited to uncovering unusual patterns in network traffic without the need for training data, as with any supervised

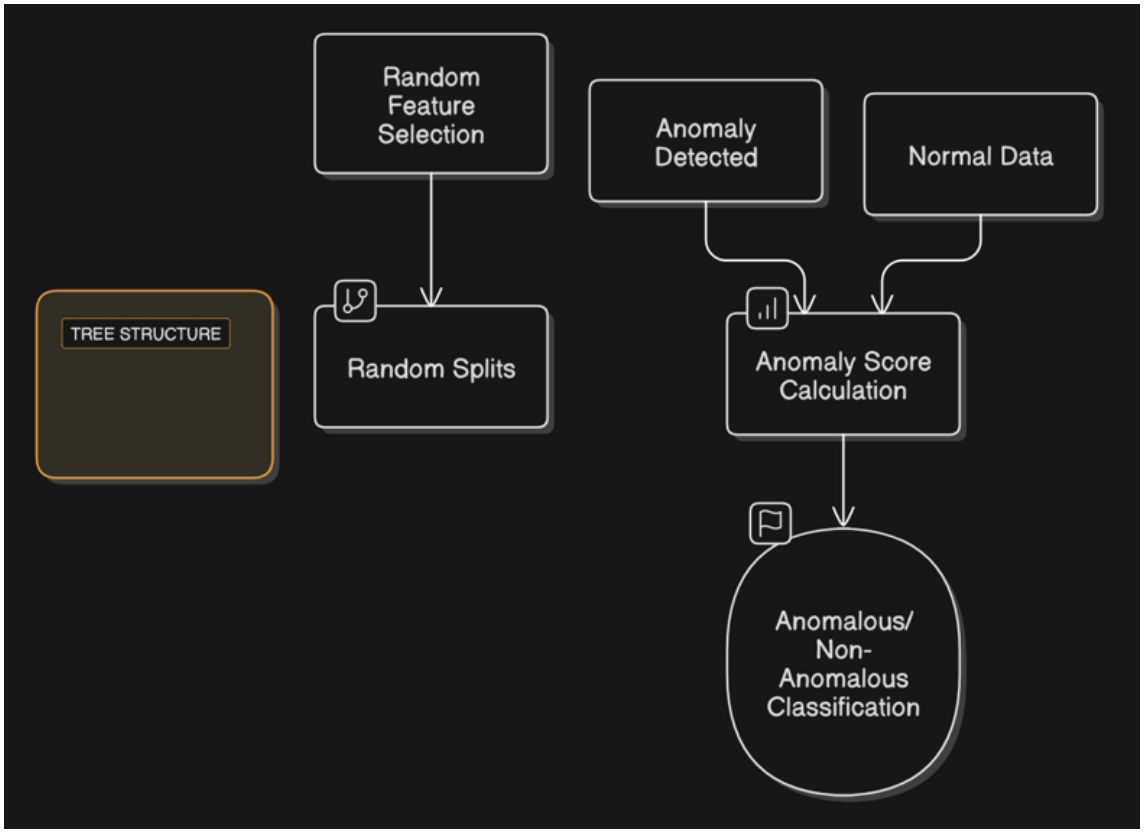

Figure 2: Process diagram of Isolation Forest algorithm

learning method.

The algorithm deals with network traffic data in batches in order to optimize performance and retain scalability in high-volume environments. Integrating blockchain-based threat intelligence sharing further enhances the detection process, giving the results of more than one node. By employing this collaborative approach, nodes derive from collective insights, ensuring both greater detection accuracy and decreased number of false positives.

The Isolation Forest algorithm provides a step-by-step illustration of how to implement the pseudocode. The training proceeds from the initialization of an empty forest and performs the actual training of the algorithm by constructing trees for anomaly detection. Anomalies scores of incoming data points are calculated during the evaluation phase, and anomalous points are marked for further action.

Below is the pseudocode for the Isolation Forest algorithm:

The main advantage of the Isolation Forest algorithm is the way in which it performs unsupervised learning tasks. Unlike supervised models which require large datasets of labeled examples, the Isolation Forest can detect novel attack patterns. Moreover, its tree-based structure guarantees rapid performance even on large datasets, thus making it an effective tool for real-time detection of intrusion in dynamic network environments.

4.4 Blockchain Architecture and Consensus Mechanism

The proposed intrusion detection system (IDS) is based on the blockchain architecture, which provides secure, decentralized, and tamper-resistant sharing of threat intelligence

Algorithm 1 Isolation Forest Anomaly Detection

Require: Network Traffic Data
Ensure: Anomalous Data Points
Initialize an empty Isolation Forest (IF).
for each node in the network **do**
    Collect network traffic data (e.g., source IP, packet size, protocol type).
    Train IF model on collected data.
end for
for each data point in the network traffic **do**
    Calculate anomaly score using IF.
    if anomaly score > predefined threshold **then**
        Mark data point as anomalous.
    end if
end for
return list of anomalous data points.

among the network nodes. This section discusses the components, operation, and consensus mechanism of the blockchain, More details its impact and contribution towards the growth of scalability, robustness, and data integrity.

4.4.1 Blockchain Design and Role

The proposed IDS uses the blockchain throughout as a distributed ledger where every network node keeps a synchronized copy of the chain [14]. By design, all nodes have the capability of accessing verified, immutable records of detected anomalies thus allowing decentralized collaboration without the presence of a central authority.

- **Distributed Ledger:** Each anomaly detected by a node is packaged as a transaction, containing details such as:
  - Detection results (D_i)
  - Metadata (M_i, including severity level and affected IPs)
  - Timestamp (t_i)

  Once these transactions occur, they are broadcasted to the blockchain network and added to the chain in new blocks.
- **Immutable Storage:** Once a transaction is completed, the transaction is added to a block and managed by the immutable ledger. It maintains records so that the threat intelligence data integrity and reliability cannot be tampered with and this is achieved.
- **Shared Intelligence:** This intelligence is available in real-time to all nodes in the network, and those nodes can collaborate to improve their ability to detect and respond to threats.

Equation:
The blockchain ledger updates its state when a node broadcasts an anomaly detection result:

$$B_{t+1} = B_t \cup T_i$$

Where:

- B_t: Blockchain ledger at time t
- T_i: New transaction containing D_i, M_i, t_i

This ensures that all nodes maintain an up-to-date copy of the ledger:

$$\forall j \in N,\, B_j = B$$

Where:

- N: Set of all nodes
- B_j: Blockchain ledger copy maintained by node j

4.4.2 Consensus Mechanism

With a Proof of Authority (PoA) consensus mechanism powering transactions and ledger consistency, the blockchain sits. This balance between security, efficiency, and scalability is this choice.

- **Validation Process:**
  - Validator nodes implicitly receive the role to validate transactions.
  - Validators meet and reach a consensus on which transactions are permissible and will be carried forward to the next block.
- **Advantages of PoA:**
  - **Low Latency:** PoA takes less time to validate, unlike computationally intensive mechanisms like Proof of Work (PoW).
  - **Reduced Overhead:** PoA makes the system highly efficient even in high-traffic environments, as the system is lightweight.
- **Consensus Efficiency**

  This means that the computational complexity of the consensus mechanism determines how long should the time be to validate a block. For PoA, this is relatively low, as shown by:

  $$\text{Validation Time} \propto \frac{1}{C}$$

  Where C represents the computational overhead of the consensus process.
- **Threat Intelligence Dissemination:**
  - New threat intelligence is distributed upon block validation across the network, to remain consistent while offering rapid response capabilities.

4.4.3 Scalability and Fault Tolerance

- **Scalability and Fault Tolerance Features**

  This distributed blockchain architecture enables our growing nodes and network traffic to scale without bottlenecks to the centralized network. As long as most validators behave honestly, it keeps its operation going even if some nodes fail or are directly malicious.

- **Threat Data Integrity**

  Blockchain records are immutable and can't be modified or deleted. Mathematically:

$$\text{Integrity}(B) = \text{True}, \forall t, \nabla B_t = 0$$

  where ∇B_t represents any attempted change to the blockchain at time t.

So basically, A blockchain-based architecture underpins the proposed IDS which enhances the secure and decentralized sharing of the threat intelligence. The system achieves low latency validation, scalability, and fault tolerance through the use of its PoA consensus mechanism. The framework, by combining anomaly detection with blockchain's immutable storage, presents a strong collaborative and adaptive cybersecurity solution.

4.5 Smart Contract Design and Functionality

The proposed IDS is built around the use of smart contracts that supply real-time, automated responses to detected anomalies. The system achieves efficient and secure threat management in the face of detected threats without delay through the integration of these self-executing contracts into the blockchain architecture [15]. This section further details the design principles of the proposed IDS, how smart contracts fit into the IDS' functional workflow, and the practical implementation of the IDS as an extension to the Ethereum platform.

4.5.1 Design Principles

Instructions or designs of smart contracts in this system are drawn by the requirement of speed, security, and reliability [16]. It sends the triggers that each contract responds to, each of which is programmed to perform specific tasks for themselves when they are invoked by predefined conditions. There are these tasks: isolating compromised nodes, blocking bad IP addresses, and notifying administrators that there has been an attack. Blockchain immutability guarantees that these contracts are executed in this way and that they no longer have to be tampered with or modified in any unauthorized way [17].

The contracts are written inside of a blockchain framework that exploits this distributed nature to achieve consensus across all nodes. That ensures that security control is consistently protected along the network. In addition, these diminish dependence on central authority and therefore increase the scalability and robustness of the system.

4.5.2 Functional Workflow

Smart contracts operate through a systematization process. If the Isolation Forest algorithm running on a network node detects an anomaly, the details of the anomaly – the type of anomaly, severity level, affected IP addresses, to name a few – are sent as a transaction to the blockchain. This transaction is verified, then added to the ledger, and triggers the smart contract associated with it.

Anomaly data is evaluated by the smart contract that runs, and depending on the conditions written by the smart contract logic in it, the appropriate action will be determined. Say, the severity level is above a critical level, then the contract can choose to isolate the compromised node in the network and also block the traffic from the attacked IP address, as depicted in Fig. 3. All this is happening when, at the same time, an alert is sent to admins who can then further take actions if needed.

All of this happens without manual intervention, placing minimal latency along the way. Only with this transparency and audibility can the administrators of the blockchain review each action taken by the smart contracts and take responsibility for the actions being taken in the system.

4.5.3 Pseudocode for Smart Contracts

To implement smart contracts the logic needs to incorporate specific rules and actions to be performed. Below is an example pseudocode that illustrates the design of a smart contract for handling detected anomalies:

Algorithm 2 SmartContract:Intrusion Response

Require: Anomalous Data Detected

1: **if** anomaly is detected **then**
2: Execute predefined security actions:

1. Isolate the affected node from the network.
2. Block traffic from the suspicious IP address.
3. Alert system administrators via notification.

3: **end if**
4: Log the response on the blockchain for audit and system enhancement.

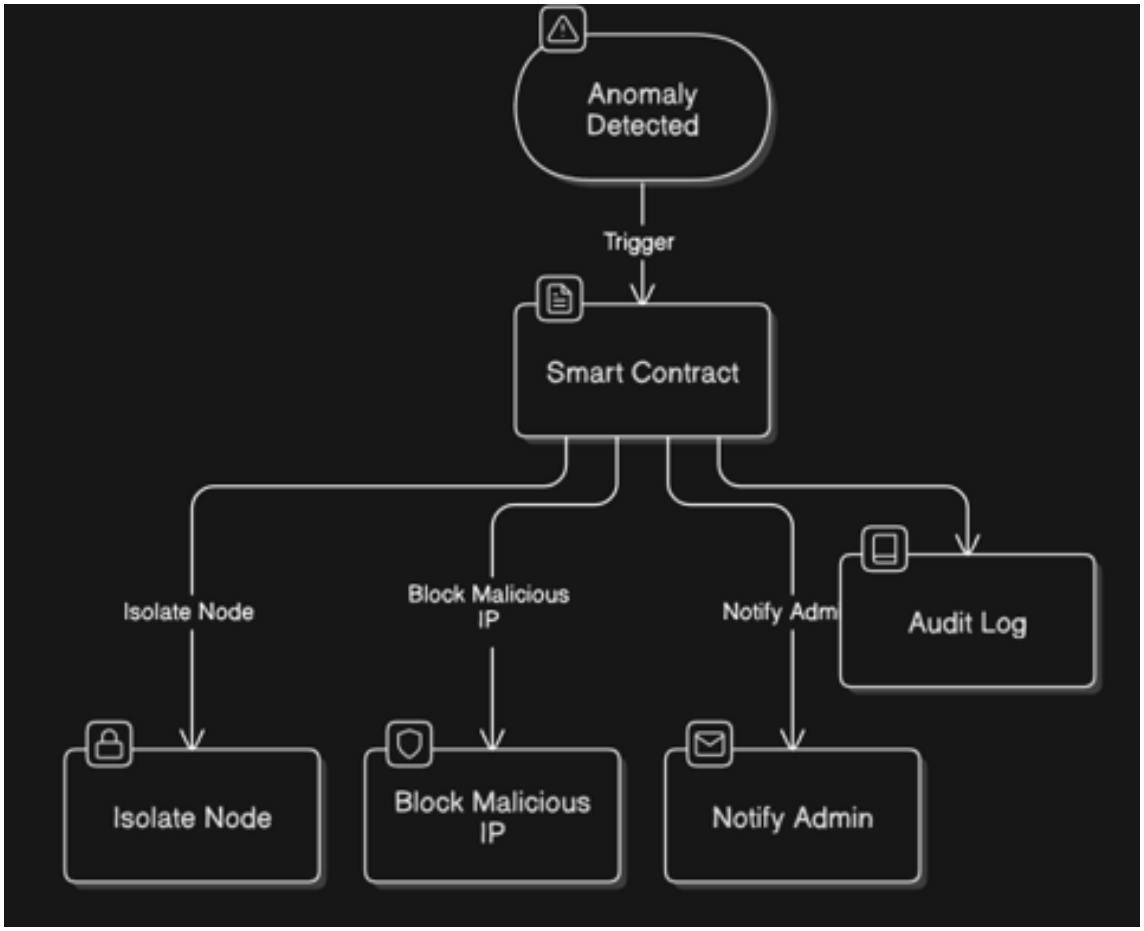

Figure 3: Functional diagram of Smart Contract

At the start of the given pseudocode, initially, the contract checks if an anomaly has been found. Next, it evaluates the severity of the anomaly against a predefined threshold. In the case of high-severity anomalies, the contract takes some action like quarantining the node that is targeted and blocking traffic from the sources that appear to be malicious. It logs less critical incidents to the blockchain as events and notifies administrators, leaving human intervention open for less critical incidents.

4.5.4 Benefits and Challenges

This brings some advantages or disadvantages to the system through the use of smart contracts. The contracts in essence make the reaction time to cyber attacks much shorter, taking these down to nothing [18]. They also guarantee security enforcement consistently without any human error. Apart from that, the blockchain itself being decentralized will make the contracts tamper-proof and thereby will make the overall system robust.

However, accomplishing this has some associated challenges. Once the smart contract is deployed it is immutable which implies that the logic of good smart contracts is not modifiable to accommodate evolving threat scenarios. That limitation imposes a severe limit on how the design can be, and a need to test carefully before deploying it. The execution of smart contracts introduces another potential challenge, particularly in regard to the computational overhead associated if these smart contracts are executed by resource-constrained nodes in the network. The issues explored above can be addressed in the future by dynamic smart contracts or off-chain execution.

4.5.5 Integration with the System

The Isolation Forest algorithm is seamlessly embedded with blockchain and smart contracts to build a solid intrusion detection framework [19]. They bridge the gap between anomaly detection and the security action they represent as an integral part of the system's automated response mechanism. For the purposes of the proposed IDS, they are indispensable in their ability to execute predefined responses rapidly and securely.

5 Performance Evaluation

The performance evaluation of the proposed blockchain-based intrusion detection system (IDS) is presented in this chapter. It is an analysis of the metrics that define its effectiveness and scalability as well as its robustness. They give you a complete grasp of the IDS' strengths and weakness areas. The work of the research is mainly carried out on the theoretical level and, for the most part, based on the findings of the research and on the conceptual design and framework developed before.

5.1 Metrics Used

The performance of the proposed blockchain-based Intrusion Detection System (IDS) is evaluated using four fundamental metrics: precision, response time, false positive rate, and detection accuracy. They serve as a total insight into how well the system serves at detecting and solving network intrusions.

5.1.1 Detection Accuracy

One of the most important criteria for the system is to detect accuracy which is important in ensuring that the system correctly recognizes intrusions among all the anomalies that are flagged. It indicates how well the IDS can identify abnormal network behaviors. A machine learning-based approach for anomaly detection which is an efficient method of isolating outliers from data Isolation Forest is used by the proposed system for detecting the anomaly. Given this, an unsupervised algorithm is used to create many decision trees to split away anomalous points from the bulk of data. This stands in contrast to traditional methods, which require pattern pre-definition, and consequently is appropriate for recognizing new or evolving threats in a dynamically changing environment.

Furthermore, the integration of blockchain technology into the system improves the accuracy. Decentralized threat intelligence sharing with collaborative validation of detected anomalies is based on contributions from each network node. The advantage of this collective approach is that it substantially lowers the probability of overlooking advanced or invisible intrusion efforts. The combination of the Isolation Forest algorithm and blockchain guarantees that the system has high detection, even when the system is complex.

5.1.2 Precision

Precision refers to a system's capacity to correctly classify legitimate intrusion and poorly classify normal anomalies as threats. The importance of this metric lies in the fact that it can signal whether the IDS generates useful alarms for administrators, without too much distraction. The proposed IDS shows great precision because of its granular analysis of network traffic. Isolation Forest algorithm scores the data points according to their deviation from the normal traffic pattern and prioritizes the most suspicious anomalies for further study. The reasoning behind this method is that it reduces the probability that benign activity will be mislabeled.

Furthermore, precision is further refined by the decentralized architecture which allows for cross-verification of anomaly data between nodes. The collective intelligence stored on the blockchain is validated by one node when it's found that the node is under attack. Such a consensus-based mechanism guarantees the true threats while lowering the false alarms, so the IDS would be highly reliable.

5.1.3 False Positive Rate

The false positive rate measures the ratio of the number of errors that the IDS detects intrusion and there is no intrusion. A lower false positive rate means a better, more dependable system, as it reduces the frequency of false alarms and reduces the administrator's attention needed to focus on real threats. The proposed IDS uses the Isolation Forest algorithm to use it efficiently isolate unusual patterns and it does not rely on stringent, rigorously defined rules to achieve a low false positive rate. It allows this flexibility to adapt to variations of the network traffic and avoid over-classifying benign anomalies.

Also, the blockchain-based authentication process is essential in terms of eliminating false positives. Since nodes share information and corroborate their findings, the system allows flagged anomalies to be checked against the rest of the network. Improving upon individual nodes generating too many excessive false alarms, this collaborative approach mitigates the risk.

5.1.4 Response Time

The critical response time (how quickly the IDS catches the detected threat) is a metric that responds instantly and tries responses. Now, in cybersecurity, these are important because they help minimize the damage done by intrusions. Smart contracts deployed on a blockchain enable the proposed IDS to reach a rapid response time. In these smart contracts, it is defined by whom and under which conditions an action will be taken for isolated compromised nodes or block IP addresses from malicious addresses. With no manual intervention, the system minimizes the variability involved with delay and consistency in response.

Beyond that, response efficiency is further augmented due to the decentralized nature of the blockchain. In contrast to traditional systems that rely on a central authority to coordinate responses, the proposed IDS entrusts each node to act autonomously. Such decentralization stops bottlenecks and delays, ensuring that threats are extinguished quickly. Response times' comparison with respect to different traffic conditions would be valuable information regarding the system performance. With such a figure, we would demonstrate the IDS's ability to respond rapidly, even in traffic scenarios.

The proposed IDS exhibits its robustness and reliability by excelling across these metrics. Being relatively highly accurate and precise and responding efficiently, it effectively

balances accuracy, precision, and response efficiency with minimum false positives providing optimum network protection.

5.2 Scalability and Fault Tolerance of the Proposed IDS

The scalability and fault tolerance of the proposed blockchain-based Intrusion Detection System (IDS) are two critical attributes that contribute to its adaptability and reliability in dynamic and large-scale network environments, as illustrated in Fig. 4. These features address the inadequacies of the typical IDS frameworks which are plagued by bottlenecks and weaknesses within a centralized model.

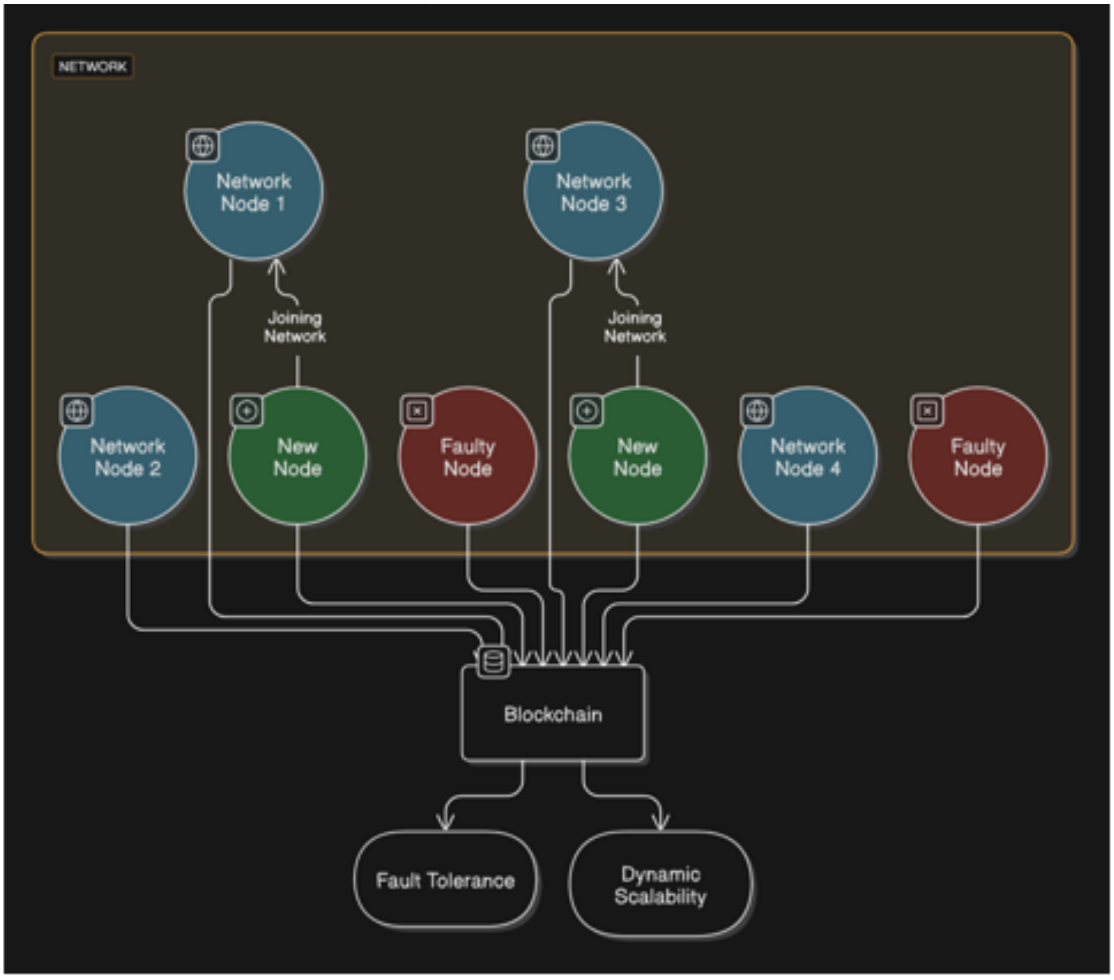

Figure 4: Scalability and Fault Tolerance Flowchart

5.2.1 Scalability

The scalability of the proposed IDS is enabled by the decentralized architecture that enables it to scale the network without a commensurate increase in latency or the consumption of resources. In conventional IDS setups, scalability is limited by the use of a central server to process and store threat data. However, as the network size grows this central hub becomes overwhelmed resulting in delayed detection of the threat and response.

On the other hand, the blockchain-based IDS allows both the data processing and storage responsibilities to multiple nodes across the network. It consists of handfuls of nodes, each of which operates entirely independently to detect anomalies from local traffic, sharing those findings with one another through the blockchain. The nodes are all independent and each of them analyzes local traffic for any anomalies that may be occurring and spreads those findings across the network through the blockchain. In this case, a decentralized approach is used, ensuring that no node becomes a bottleneck, allowing the system to have high performance even with a great number of devices or network traffic. Moreover, the scalability of the system is supported by the modularity. The network can be seamlessly extended with nodes, and their computational capacities adjust to meet special requirements. Thus, for example, high-traffic nodes may be outfitted with more durable hardware capable of processing data quickly, while low-traffic nodes could live with few resources. The flexibility of the IDS permits its deployment in various environments from smaller, connected local area networks to scaled, connected industrial systems.

However, scalability is hardly an easy thing. With time, the blockchain's growing ledger size elevates the storage requirements of a node, and consensus algorithms can cause delays due to more transactions. But thanks to lightweight mechanisms like Proof of Authority (PoA) or Delegated Proof of Stake (DPoS), the system handles these issues. However, these algorithms make sure transactions get processed quickly and the integrity of the blockchain remains secure. Beyond that, further scalability can be obtained by offloading the load of the main blockchain through integration with layer 2 solutions like 'sidechains' or 'sharding'.

5.2.2 Fault Tolerance

Another cornerstone technology of the proposed IDS consists of fault tolerance, meaning that the IDS continues operating even under node failures or malicious nodes. Centralized IDS frameworks are exceptionally prone to single points of failure. Traditional centralized IDS frameworks are vulnerable to a single point of failure. The whole system is rendered ineffective if the central server is compromised or has technical problems, exposing the network to attacks. The proposed IDS blockchain-based design eliminates this vulnerability. The blockchain ledger is a copy of both that is stored in each node to keep the core functions of the system in place should some of the nodes fail: data storage, threat detection, and intelligence sharing. Thus, the redundancy is decentralized and a bad or compromised node cannot take the whole system down.

Additionally, the reliability of blockchain records comes from the fact that they are immutable. Blockchain-based storage of threat intelligence data cannot be altered or deleted to achieve the integrity of historical records. This feature is especially important when it comes to collaborative threat detection, as in the absence of it, each node will operate on unverified data. Finally, the consensus mechanism allows the system to resist node failures. If a subset of nodes behave maliciously or do not participate in the network, but the rest of the nodes continue to strive in the network, agreement for the validity of transactions can be reached between them. The IDS is also resilient to accidental failure, as well as sabotage.

In a way, fault tolerance also has its own set of problems. With multiple copies of the blockchain ledgers being maintained this increases the storage requirements and the system has to constantly verify that our data is right on storage to avoid errors propagating. However, the benefits of decentralization which are security, reliability, and flexibility exceed these challenges.

5.3 Theoretical Analysis of System Robustness

A critical measure of a system's robustness indicates its resistance to hostile environments, mitigation of environmental change, and operational ability to carry on despite continually changing threats. Based on these unique combinations of decentralized architecture, advanced machine learning techniques, and automated response mechanisms, the proposed blockchain-based Intrusion Detection System (IDS) proposes to be robust. These components are completely interrelated to form a system that is resilient, adaptive, and highly reliable.

Decentralized threat intelligence sharing capability is one of the system's central pillars towards its robustness. Centralized IDS frameworks suffer from the introduction of single points of failure, where a single compromised centralized authority can become a source of a catastrophic breach. However, unlike the blockchain architecture used in this case, data are spread across all participants in this architecture, and thus such vulnerabilities are eliminated. When a node observes an anomaly recurrence, it logs and broadcasts the findings into the blockchain to the rest of the network. The first thing to understand is the immutable characteristic of this shared ledger that any generated threat cannot be deleted or changed. This immutability guarantees that every node can see 100% reliable and consistent historical data, which facilitates learning from past incidents and decreases the likelihood that the system suffers the unfortunate fate of repeated attacks. Additionally, the blockchain lends itself toward a collaborative aspect, allowing for live updates requiring that no nodes are out of sync and subsequently in the dark of emerging threats.

The Isolation Forest algorithm serves as a core component of the proposed IDS, which increases significantly the system's resiliency to new as well as evolving cybersecurity threats. Compared to the conventional supervised Machine learning models, the Isolation Forest is an unsupervised model. In environments where new types of attacks can arise without examples, the ability to learn such examples is particularly valuable. The algorithm is great at isolating the anomalies in the data using the unique trait of outliers being easier to isolate than normal data points. The system is also highly flexible and robust to a wide range of intrusion scenarios, by its ability to dynamically identify novel attack patterns. Furthermore, the algorithm maintains low computational overhead, made possible by having low overhead, so that anomaly detection remains efficient when traffic grows.

The smart contract-based automated response mechanisms are another key element of robustness. In traditional systems, in response to actions, human intervention introduces delays and inconsistency. As opposed to that, in the proposed system, responses will be automated automatically using the pre-defined rules, and smart contracts based on the blockchain. Consider that if a node detected a very severe anomaly, the smart contract could isolate the compromised node or block the malicious IP address upon detection of an anomaly. This robotic automation ensures that response actions are quickly performed and minimizes the window of vulnerability against an attack. The system also records in the blockchain every response action, making the record transparent and verifiable of all events. The transparency offered not only adds to accountability but also has important lessons to learn to further refine response strategies in the future.

Despite these strengths, the robustness of the system is not unlimited. This reliance on blockchain technology adds computational and storage overhead: we need each node to both process and store a copy of the blockchain. In high-traffic scenarios or nodes with constrained resources, this can be a challenge. In fact, smart contracts give you a consistent, fast response, but the immutability of smart contracts means that the pre-defined response rule cannot be quickly changed to respond to unexpected situations. In order to

tackle these problems, future frameworks could lean on hybrid architectures that combine in-chain and off-chain orchestration, or alternative dynamic smart contracts that dynamically evolve based on the threat landscape. In conclusion, we show that the proposed IDS benefits from a high degree of robustness delivered by a decentralized architecture, adaptive anomaly detection, and automated response capabilities. The complexity of modern security threats calls for secure data sharing, dynamic detection capability by using a well-defined Isolation Forest algorithm, and rapid response through smart contracts.

6 Discussion

6.1 Advantages of decentralized threat intelligence sharing

The proposed Intrusion Detection System (IDS) takes advantage of blockchain technology for decentralized threat intelligence sharing while dispensing with the limitations of traditional centralized methods. Available today is the traditional IDS architecture which consists of a single central server collecting, analyzing, and disseminating threat information. This approach works well in small-scale systems but introduces vulnerabilities, inefficiencies, and scalability challenges for modern networks. On the other hand, the decentralized approach cares about how each node can operate in isolation but contribute to the collective intelligence of the network. Below are the detailed advantages of decentralized threat intelligence sharing:

- **Enhanced Resilience and Fault Tolerance:** Centralized systems have only one point of failure. The entire IDS is compromised if the central server is attacked or not available because of system failure, which exposes the network to attack. By distributing data storage and processing across multiple nodes, the blockchain-based decentralized architecture does away with this risk. It is still working even if a few nodes are compromised or fail.
- **Data Integrity and Trustworthiness:** The shared threat intelligence is immutable by using blockchain technology. Once the chain is placed in fact, a block containing threat data that is added to the chain cannot be changed or removed. It ensures the data will be authentic and reliable across the network. That's because each node can trust the threat intelligence it gets and it's necessary for coordinated and efficient threat responses.
- **Real-Time Collaboration and Adaptive Response:** In decentralized systems, each node is an active participant in threat detection and distribution. The information is then instantly broadcasted to the blockchain as anomalies are detected locally. This verified information is available to other nodes right away and can make changes to their detection models and response strategies. The resulting system is highly adaptive to emerging cyberattacks, thanks to the collaborative approach that accelerates the detection and mitigation of the threats.
- **Scalability for Large and Dynamic Networks:** Decentralized threat intelligence sharing can be scaled as such by its very nature. As the number of nodes grows the larger the network becomes and new nodes can join without a central server and overload a bottleneck. Each node contributes to the fact that the system's total processing and storage of threat data. The system is best suited to large and dynamic networks like IoT ecosystems, smart grids, or traditional centralized systems, but would be ill-suited to performance.

- **Efficient Resource Utilization:** A decentralized model distributes the processing and storage load across all nodes to optimize the resource as minimally as possible. With local anomaly detection performed by each node, the contributions to the blockchain reduce the computational load for each component. In addition, this distribution avoids centralized processing power, a common inefficiency in traditional systems.

- **Security Against Targeted Attacks:** Attackers are often targeting the central server in centralized IDS frameworks to disrupt the system. The risk of the central server is mitigated by removing it from the central server altogether. The distributed ledger allows nodes to operate independently and maintains that threat intelligence is available and secure even in attacks against different parts of the network coordinated by an adversary.

6.2 Machine learning's role in improving anomaly detection

The intrusion detection system based on the proposed Intrusion Detection System (IDS) includes an important innovation: the integration of machine learning (ML) with the Isolation Forest algorithm. Machine learning (ML) advances the IDS' capability to detect anomalies beyond the current rule-based paradigm and empowers dynamic, data-based decision-making. This section elaborates on how ML contributes to improving anomaly detection in the system:

- **Unsupervised Learning Approach:** An unsupervised learning approach using the Isolation Forest algorithm is particularly suitable in cybersecurity scenarios in which labeled data for training is not available or not very abundant. Unlike supervised methods that require massive amounts of training datasets of known attacks, this algorithm isolates anomalous observations based on their intrinsic characteristics. It allows for a high degree of adaptability in response to a variety of dynamic and changing network environments with new, or previously unknown threats emerging regularly.

- **Focused Detection of Anomalies:** The isolation Forest algorithm's core principle is to isolate anomalies in the above distinct outliers in the data space. The algorithm creates random partitions of the data points and tests how easy a point is to isolate, in distinguishing between normal and suspicious behavior. By targeting malicious activities, false negatives are measured much less and more accurately.

- **Reduction in False Positives:** A significant difficulty for existing IDS frameworks is that benign network behaviors are easily confused as threats, a phenomenon known as false positives. The problem of this is resolved by the Isolation Forest algorithm by refining the process of detection. The IDS relies on anomaly scores to prune cases on the fringes, alerting on activities that are clearly suspicious. The result of this is reduced unnecessary distraction and resource allocation to what is not critical.

- **Adaptability to Emerging Threats** Cyberthreats evolve all the time and inevitably adopt different patterns of attack, even compared with known attack signatures. The unsupervised nature of the Isolation Forest algorithm permits the IDS to examine such deviations in network behavior as threats that were previously

not classified. It is crucial that such adaptability allows for keeping strong defense mechanisms in environments where attacking vectors evolve rapidly, all the time.

- **Scalability and Efficiency in Large Networks**

  The Isolation Forest algorithm is efficient in large-scale network environments. The IDS processes traffic locally at each node, as well as securely sharing observed insights across the blockchain, maintaining capacity at each component to avoid overburdening anyone. By distributing the processing, the system is scaled and maintains high performance even as the network grows.

- **Augmentation of System Intelligence** Using machine learning, I enrich the IDS with a continuous learning mechanism. The algorithm works better as it gets more data and processes it, refining its understanding of how normal and abnormal behaviors act in the network. Not only does this iterative improvement improve detection accuracy, but the IDS can also always be a step ahead of new threats. The proposed IDS leverages machine learning to greatly improve the anomaly detection capabilities through these mechanisms, while guaranteeing accuracy and the ability to adapt to modern cybersecurity problems.

6.3 Limitations and trade-offs in performance.

Although the proposed blockchain-based IDS provides many advancements in intrusion detection, it comes with some limitations and trade-offs for its real-world network implementation. The challenges are primarily driven by the integration of blockchain and machine learning technologies and the operational mode required of a large-scale, dynamic environment.

Computational overhead is one of the most notable challenges that blockchain brings. Any form of decentralized ledger requires an immense amount of processing power — specifically, performing validation of transactions and syncing the blockchain to all nodes. In high-volume traffic networks, this may overstrain the resources of each individual node, although it would make sense that a network with fewer nodes would have more resources; obviously, it would also have less traffic. The overall system's efficiency may be hindered by delays in the nodes that have a limited computational capability.

The other key limitation of blockchain is its latency imparted from its consensus mechanism. Decentralization makes data safe and authentic while making the whole process of reviewing new blocks and networking consistency get delayed. In such environments where real-time threat response is imperative (i.e. financial institutions, healthcare networks) this latency can prevent timely detection and mitigation of a cyberattack. The proposed system auto-responds with smart contracts to automate them, yet the response time of the responses employed still relies on the blockchain's underlying infrastructure.

The system also has a scalability problem. With the increase in the number of nodes participating in the network as well as the volume of the threat intelligence data grows, leading to an expansion of the blockchain ledger. It can cause node storage burdens and slow down the synchronization processes as nodes grow. One specific situation where this can be a problem is when several nodes simultaneously upload threat intelligence data, without an effort to update the distributed ledger consistently; this delays updating the distributed ledger and results in that node being non-responsive overall.

Although the Isolation Forest algorithm works well for anomaly detection, it also heavily relies on trade-offs. To analyze a large dataset and identify patterns that are not normal, the algorithm suffers from the requirement of large computational power. This

computational demand can result in latency in finding anomalies, particularly if nodes must process data locally, in networks with heavy traffic. As the nature of the threat activity threatens periods of high level of activity, the balance between detection accuracy and processing speed has become critical since delays beyond a threshold can render the system ineffective. Next, the immutability of smart contracts, a key attribute of Blockchain, is a bottleneck when it comes to adaptability. Once these contracts are deployed, they cannot be altered, meaning we are unable to change our response strategy in the instance of new cyber threats. This immutability maintains the integrity of automated responses while leaving the system rigid when combined with evolving threat landscapes and the necessity for meeting current or evolving threats with adaptive measures.

However, though with these limitations, the proposed IDS is a compelling framework for decentralized and automated cybersecurity. Technological advancements to address these trade-offs, including lightweight blockchain architectures and hybrid on-chain / off-chain models could greatly increase its practical applicability. Similarly, optimization on the Isolation Forest algorithm for less computational resources further improves the detection accuracy over the challenge of anomaly detection on a high load. By addressing these limitations through understanding and mitigation, the proposed IDS can become a more robust and scalable IDS solution that would be able to confront the complexity of the current modern cybersecurity threats.

7 Conclusion and Future Works

7.1 Conclusion

This thesis presents a novel framework for enhancing cybersecurity through a decentralized intrusion detection system (IDS) based on the use of machine and blockchain learning technology. We present the core contribution in the integration of an unsupervised machine learning model, the Isolation Forest algorithm, to a blockchain-based architecture that supports a network to perform decentralized threat detection and intelligence sharing. This study fills a gap by providing necessary directions for more robust, scalable, and adaptive cybersecurity solutions, which were able to address existing key limitations of traditional IDS systems including high false positive rates, dependence on centralized structures, and slow response times.

The other important contribution of this work is the use of smart contracts to automate the response mechanisms. Traditional concentric defense framework involves a slow and human intervention-based approach to filtering out threats. While the proposed system implements smart contracts to perform the predetermined actions (e.g., isolate nodes that are compromised or block malicious IP addresses), it decreases the time response and enhances efficiency. In dynamic network environments, where a speedy response can spare serious harm, this automation is particularly handy.

Another important part of this research is the decentralization of threat intel sharing. The system uses blockchain's distributed ledger capability to enable secured and immutable storage of detected anomaly data and their pointers of metadata. Moreover, this approach helps to filter transparency as well as foster collective intelligence among network nodes, allowing network nodes to react adaptively and dynamically to emerging threats. In addition, Isolation Forest based on anomaly detection does not depend on labeled datasets and therefore suits well real-world applications with a lack of labeled datasets.

Further, the framework proposed also handles scalability and fault tolerance, two important challenges in cybersecurity. The system avoids single points of failure by distributing detection and response responsibilities across multiple nodes which can operate even in the presence of node failures or malicious actors. For large-scale networks, especially those relying on central systems, these decentralized designs are relevant.

This study has also contributed to the advancement of IDS solutions by blending the techniques of blockchain technology, learning machines, and automated smart contracts. It is a forward step towards creating adaptive and scalable systems of cybersecurity that can deal with the dynamic challenge of cyber threats.

7.2 Proposed advancements

The proposed framework provides significant improvement over currently available IDS systems and there remain two areas in which additional research and development is needed. Advancement includes also the leveraging of deep learning models, layer-2 scaling solutions, and hybrid storage architecture. Deep learning models, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or hybrid models, have demonstrated remarkable capabilities in handling complex and high-dimensional data.

7.2.1 Deep learning

In this framework, the performance of the Isolation Forest algorithm for anomaly detection is demonstrated to be effective but deeper learning techniques could improve its performance.

- **Improved Detection Accuracy:** Network traffic deep learning models can learn complex patterns and have natural capabilities in detecting advanced, evolving cyber threats with greater accuracy than traditional machine learning.
- **Adaptability:** Unlike the Isolation Forest, which relied mostly on anomaly detection, deep learning models can be trained to distinguish threats and deliver more in-depth knowledge about what the detected intrusions are.
- **Scalability:** Optimized with the right technology, deep learning models can ingest a volume of network data in the giant system and thus be used in a future implementation.

Future work may inherit the summary of combining the fast and precise but not deep pure detection mechanism of a deep learning hybrid mechanism with the strength of the Isolation Forest algorithm for fast anomaly detection. While these two approaches may cause more false positives than the two pure methods, they would also improve detection rates.

7.2.2 Layer-2 scaling

For data integrity and security, reliance on blockchain for decentralized threat intelligence sharing is a necessity, but as network size and transaction volumes expand, performance is still a challenge. A promising path forward to overcoming this limitation is layer-2 scaling solutions. The solution of these solutions consists of offloading some operations from the main blockchain, reducing the computational and storage burden on it, and keeping security and decentralization intact.

For instance, sharding splits the blockchain into smaller, manageable partitioning that each can handle transactions independently. This gives a way to increase the throughput of the blockchain network so that its capacity to handle such increased traffic figures is not compromised. Like sidechains — auxiliary blockchains that are separate from the main chain — one can solve congestion issues on the main chain by allowing sidechains to undertake a specific kind of transaction, or even store non-critical degree of data.

Another way is to use state channels, whereby off-chain transactions are made and settled in on-chain. State channels reduce latency by minimizing the amount of transactions present on the blockchain, improving transaction speeds. These solutions not only scale the blockchain but preserve the scalability of the IDS framework as large-scale,

high-traffic networks grow. Future work may consider the performance trade-offs of these scaling methods, and select the desired strategy for the proposed IDS.

7.2.3 Hybrid storage

However, as the volume of threat intelligence data grows, the ability to store and retrieve the data efficiently impacts the blockchain. The immutability and security of the data are secured by blockchain, but not for storing the large dataset. However, to combat this challenge, the adoption of a hybrid storage architecture seems to provide a possible solution by combining on-chain and off-chain storage schemes. Within this architecture, critical metadata, including anomaly scores, timestamps and a summary of detected threats can be stored (directly) onto the blockchain to leverage its security and transparency. Meanwhile, off-chain storage of detailed threat logs and raw network data in distributed databases or cloud storage systems is commonplace. Using this approach, the demands for storing data on the blockchain are drastically reduced without compromising the accessibility and integrity of important data.

The hybrid model enables efficient data management as well. The blockchain stays lightweight and agile by storing only the summarized information on the chain so that dissemination is faster as well as execution and retrieval times. At the same time, much of the data can reside in off-chain storage, allowing a comprehensive threat analysis without tying up the system. Secure interfaces between on-chain and off-chain components are achieved by integration.

The hybrid approach could be the subject of future implementations exploring innovative means for its optimization. For example, an InterPlanetary File System (IPFS) or the IDS framework adapted to a blockchain-compatible cloud storage service might help increase the scalability and integrity of the IDS framework. The hybrid storage architecture finds a real solution in balancing out the weaknesses of both on-chain security and off-chain scalability.

8 Submission History

The original paper of this research was first prepared for the ICCIT conference, but it was rejected. Based on the reviews given by the conference reviewers after some modifications, it was accepted by AINA-2025: The 39-th International Conference on Advanced Information Networking and Applications 2025 on the 7th of January 2025.

Bibliography

[1] S. Al-E'mari, M. Anbar, Y. Sanjalawe, S. Manickam, and I. Hasbullah, "Intrusion detection systems using blockchain technology: A review, issues and challenges," Computer Systems Science & Engineering, vol. 40, no. 1, pp. 1-15, 2022.

[2] K. Begum, M. A. I. Mozumder, M. I. Joo, and H. C. Kim, "BFLIDS: Blockchain-Driven Federated Learning for Intrusion Detection in IoMT Networks," Sensors, vol. 24, no. 14, p. 4591, 2024.

[3] G. Rathee, C. A. Kerrache, and M. A. Ferrag, "A blockchain-based intrusion detection system using Viterbi algorithm and indirect trust for IIoT systems," *Journal of Sensors and Actuator Networks*, vol. 11, no. 4, p. 71, 2022.

[4] R. C. Ripan, I. H. Sarker, M. M. Anwar, M. H. Furhad, F. Rahat, M. M. Hoque, and M. Sarfraz, An Isolation Forest Learning Based Outlier Detection Approach for Effectively Classifying Cyber Anomalies, p. 270–279. Springer International Publishing, 2021.

[5] S. Mishra, "Blockchain and machine learning-based hybrid ids to protect smart networks and preserve privacy, "Electronics, vol. 12, p. 3524, Aug. 2023.

[6] H. H. Pajooh, S. Demidenko, S. Aslam, and M. Harris, "Blockchain and 6G-enabled IoT," *Inventions*, vol. 7, no. 4, p. 109, Nov. 2022.

[7] B. Mondal, F. Faisal, Z. Tusnia Towshi, M. Fahad Monir and T. Ahmed, "A Gradient Boosted ML Approach to Feature Selection for Wireless Intrusion Detection," 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring), Florence, Italy, 2023, pp. 1-5, doi: 10.1109/VTC2023-Spring57618.2023.10199490.

[8] I. A. Toke, M. F. Monir and T. Ahmed, "Blockchain in Agriculture: A Scalable Solution for Crop Quality Assessment and Data Integrity," 2024 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Tbilisi, Georgia, 2024, pp. 205-210, doi: 10.1109/BlackSeaCom61746.2024.10646243.

[9] W. Zhang, Y. Yuan, Y. Hu, K. Nandakumar, A. Chopra, S. Sim, and A. De Caro, Blockchain-Based Distributed Compliance in Multinational Corporations' Cross-Border Intercompany Transactions: A New Model for Distributed Compliance Across Subsidiaries in Different Jurisdictions, p. 304–320. Springer International Publishing, Dec. 2018.

[10] G. Wood et al., "Ethereum: A secure decentralized generalized transaction ledger," Ethereum project yellow paper, vol. 151, no. 2014, pp. 1–32, 2014.

[11] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," IEEE Access, vol. 4, p. 2292–2303, 2016.

[12] M. Zhaofeng, W. Lingyun, W. Xiaochang, W. Zhen, and Z. Weizhe, “Blockchain-enabled decentralized trust management and secure usage control of iot big data,” IEEE Internet of Things Journal, vol. 7, no. 5, pp. 4000–4015, 2020.

[13] L. Deloitte, “Blockchain: Enigma, paradox, opportunity,” Retrieved, vol. 5, no. 21, p. 2021, 2016.

[14] T. A. Aadeef, M. F. Monir and S. A. Alam, ”Educational Certification on Ethereum blockchain: Analysis on How BCert and UZHBC differ,” IEEE EUROCON 2023 - 20th International Conference on Smart Technologies, Torino, Italy, 2023, pp. 337-342, doi: 10.1109/EUROCON56442.2023.10198929.

[15] I. Makhdoom, K. Hayawi, M. Kaosar, S. S. Mathew, and M. M. Masud, “Blockchain-based secure cids operation,” in 2021 5th Cyber Security in Networking Conference (CSNet), vol. 310, p. 103–106, IEEE, Oct. 2021.

[16] P. K. Naik, K. Mannanuddin, S. K, S. Sah, H. Patil, and F. A, “Enhancing iot security with self-adaptive blockchain-based intrusion detection system,” in 2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS), vol. 9, p. 1–5, IEEE, Dec. 2023.

[17] A. M. Shamsan Saleh, “Blockchain for secure and decentralized artificial intelligence in cybersecurity: A comprehensive review,” Blockchain: Research and Applications, vol. 5, p. 100193, Sept. 2024.

[18] [1] N. Alexopoulos, E. Vasilomanolakis, N. R. Ivánkó, and M. Mühlhäuser, “Towards blockchain-based collaborative intrusion detection systems,” in *Lecture Notes in Computer Science*, 2018, pp. 107–118.

[19] N. A. Dawit, S. S. Mathew, and K. Hayawi, “Suitability of blockchain for collaborative intrusion detection systems,” in 2020 12th Annual Undergraduate Research Conference on Applied Computing (URC), vol. 19, p. 1–6, IEEE, Apr. 2020.