National College of Ireland

A Blockchain-Integrated Detection System with Hash-Based Validation and AI under Zero Trust Security for APTs

MSc Research Project
MSc in Cybersecurity

RYAN CHERIAN ROY
Student ID: x23277220

School of Computing
National College of Ireland

Supervisor: Kamil Mahajan

National College of Ireland

MSc Project Submission Sheet
School of Computing

Student Name:	RYAN CHERIAN ROY
Student ID:	X23277220
Programme:	MSc in Cybersecurity **Year:** 2024-2025
Module:	MSc (Research) Practicum/Internship
Supervisor:	Kamil Mahajan
Submission Due Date:	01/09/2025
Project Title:	A Blockchain-Integrated Detection System with Hash-Based Validation and AI under Zero Trust Security for APTs
Word Count:	7770 **Page Count :** 22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: RYAN CHERIAN ROY

Date: 31/08/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable)	

A Blockchain-Integrated Detection and Prevention System with Hash-Based Validation and AI under Zero Trust Security for APTs

Ryan Cherian Roy / x23277220

Abstract

Intrusion Detection Systems (IDS) play a critical role for enterprises for monitoring and detecting unauthorized entry points or malicious attacks. However, the current IDS systems with signature-based detection engine raise an alarm whenever network traffic matches any signature. This process results in significant challenges when dealing with the modern zero-day attacks and other fileless and insider attack threats. They often bypass conventional detection methods by finding ways to avoid the known pattern of detection and thus exploiting system vulnerabilities in novel ways. So, to address this issue, this paper introduces a Blockchain-Enabled Intrusion Detection System which is designed to enhance detection capabilities through the integration of blockchain-based file integrity verification system and anomaly detection using a Random Forest model within a Zero Trust security framework. This system makes use of blockchain technology to store verified endpoint data in the form of SHA256 hashes on its blocks which is then used to verify the system integrity in comparison to the hashes collected from real-time processes and files from the endpoint agent. If a file hash is absent or altered, it is flagged for further analysis. Subsequently, the flagged program's behavioral patterns are assessed using machine learning techniques to identify potential advanced threats or any persistent activities by making use of Sysmon operational logs. This dual-layer approach enables real-time detection, tamper-proof logging, and enhanced threat response. The use of blockchain ensures immutable and verifiable integrity records, while the Random Forest model effectively distinguishes between benign and malicious activity. BIDPS's modular design supports distributed deployment, making it adaptable to varied environments, including SMEs, enterprise networks, and resource-constrained IoT systems.

Keywords- IDS, Blockchain, Hash Verification, Zero Trust, Anomaly Detection

1 Introduction

In today's digitally interconnected world, the increasing reliance on networked systems has significantly amplified the risk and impact of cyberattacks(Mallick and Nath 2024). From mass-scale ransomware attacks to covert Advanced Persistent Threats (APTs), the unfolding threat landscape has revealed the inadequacies of legacy Intrusion Detection Systems (IDS). These legacy IDS, especially signature-based systems, are effective only against known attack patterns and cannot detect new or zero-day attacks. This gap introduces severe vulnerabilities in enterprise networks, critical infrastructures, and intelligent environments such as IoT and edge devices(Iyer 2021).

Current methodologies have started embedding ML and AI into IDS to improve their capability to detect anomalous behaviour. These systems tend to have high false positive rates, black-box decision-making, and low adaptability in resource-limited scenarios(Kocher and Kumar 2021). Concurrently, blockchain technology has risen as an influential element in cybersecurity because of its decentralized nature, unalterable data storage, and auditability. These features position it to be a good bet for maintaining data integrity, especially for securing system logs and authenticating execution events(Ahmad Ali et al. 2022). However, current solutions do not adequately merge the strengths of blockchain and AI to address real-time, endpoint-level threats. Specifically, there has been little focus on combining hash-based file validation with blockchain smart contracts for identifying unauthorized or tampered executables. Additionally, fileless attacks, where few traces remain on disk are seldom handled in current frameworks despite being one of the most debilitating categories of contemporary threats. This motivates the development of a Blockchain-Enabled Intrusion Detection and Prevention System (BIDPS) that combines the blockchain cryptographic trust, Machine learning anomaly detection and hash-based file verification precision to protect endpoint devices. The system is most suited for implementation in accordance with the bases of the Zero Trust security paradigm, and it is appropriate for use in dynamic, distributed, and high-risk environments.

1.1Research Gap

In spite of the fast improvements in intrusion detection systems (IDS), existing solutions continue to lag behind in identifying sophisticated cyber threats such as zero-day attacks, fileless malware, and insider threats(Zaid and Garai 2024). Signature-based IDS are speedy but not effective against unfamiliar patterns, whereas machine learning (ML)-based IDS models suffer from potential high false positives and lack explainability in decision-making(Neupane et al. 2022). Blockchain technology, with its capability for offering decentralized, tamper-proof records, holds significant promise for increasing IDS security and credibility(Dommari and Vashishtha 2025). Current blockchain-based IDS platforms mainly centre on secure storage of logs or high-level event recording without emphasizing fine-grained file verification by real-time hash comparison. Most studies are devoid of practical deployment of hash-based detection solutions at the endpoint.

In addition, work tends to neglect performance bottlenecks such as latency due to blockchain activity or computation on resource-limited devices such as IoT endpoints. There is also an extensive gap in assessing such systems in realistic attack settings, and few methods present a holistic architecture with adherence to Zero Trust principles. Therefore, a solid, lightweight, and blockchain-supported solution that mixes hash validation, AI-based anomaly detection, and Zero Trust enforcement is urgently required to bridge this gap of cybersecurity.

1.2Research Question

How can a blockchain based intrusion detection and prevention system integrated with machine learning technology enhance the security of endpoint devices against APTs within a zero-trust architecture?

From this question, the following questions will be answered by the end of this research:

1. How effectively can a private Ethereum blockchain smart contract perform tamper-proof hash verification for file integrity validation in endpoint systems?
2. To what extent can a Random Forest anomaly detection model trained on Sysmon logs using LMD 2023 dataset identify fileless malware and zero-day threats in real time?
3. What is the detection accuracy, latency, and resource efficiency of BIDPS compared to traditional endpoint security mechanisms?

1.3 Research Objective

- To design a hash-based verification mechanism using blockchain smart contracts for detecting unauthorized file executions and enhanced logging capabilities.
- To integrate Sysmon event logging with a Random Forest model for real-time anomaly and fileless attack detection.
- To evaluate BIDPS performance in terms of detection accuracy, blockchain latency, false positives/negatives, and resource efficiency.

1.4 Research Methodology Overview

The proposed methodology is the design, development, and evaluation of a Blockchain-Enabled Intrusion Detection and Prevention System (BIDPS) that combines hash-based file validation, blockchain smart contracts, and AI-driven anomaly detection in a Zero Trust security framework. The architecture of the system starts with the installation of Sysmon on endpoint machines to gather detailed system activity logs, such as process creation, file changes, and network activity. Other ETW sources can also be used, but here we went ahead with Sysmon due to its compatibility with Windows system for testing purposes. The Endpoint Agent, which is a custom-written agent, monitors these logs and retrieves meaningful event IDs for anomaly detection and also dynamically computes SHA-256 hashes of all new executed applications/files. These hashes are then locally checked against a pre-defined hash. If none is matched, the hash is sent to a smart contract that has been placed on a private Ethereum blockchain. The smart contract checks if the hash is already present in the secured ledger. If it is not found, the file is identified as suspicious, and an alert is triggered, with the occurrence being recorded immutably on the blockchain. Alongside this file-based detection mechanism, an anomaly detection engine based on machine learning analyses behaviour-based patterns from Sysmon logs which are captured real-time and modified into json format for the detection of zero-day malware and fileless APTs. System-level metrics are used to develop features, which are then input into trained models such as Random Forest for classification. Performance is measured in terms of detection accuracy, latency, resource usage, and false positive/negative rates under a controlled scenario. A dashboard UI facilitates real-time monitoring of alerts, actions, and blockchain transactions, with the added benefit of forensic transparency.

1.5 Structure of the Report

A detailed comparison of this study with existing literature is discussed in Section 2 (Related Work), highlighting existing methods of intrusion detection, machine learning, and blockchain implementation, and the gaps that the current research aims to fill. Section 3 (Research Methodology) describes the methods and strategies utilized to attain the research goals, such as the utilization of Sysmon logs, hash-based file checking, smart contract checking, and

anomaly detection using artificial intelligence. In Section 4 (Design and Implementation Specifications), we provide the architectural design and technical requirements utilized in developing the proposed Blockchain-Enabled Intrusion Detection System. Sections 5 discuss the implementation and testing of the system, including tools and platforms employed, followed by performance testing in terms of detection accuracy, response time, and resource consumption. The report is concluded with Section 6 (Conclusions and Discussion), which summarizes the most important findings, cites the limitations, and provides possible future directions for research and development**.**

2 Related Work

2.1 Intrusion Detection Systems: Traditional vs Modern Approaches

Mohanad Sarhan et al. (2022) introduce the HBFL framework, a hierarchical blockchain-supported federated learning system for IoT intrusion detection. The architecture combines edge, fog, and cloud layers, relying on smart contracts to govern updates to models securely while achieving immutability of common logs. Locally learned ML models produce updates that are checked on-chain prior to aggregation, maintaining privacy and decentralizing trust. Testing on typical IoT datasets reveals enhanced detection precision and tamper-resistance, proving the practicality of hybrid blockchain-federated methods in real-time IDS.

Critical Evaluation: Excellent privacy and tamper-resistance but the layered structure adds communication latency and complexity. Scalability to larger-scale IoT deployments and real-time requirements are yet to be tested.

Sáez-de-Cámara et al. (2023) introduce a federated learning architecture with clustering for anomaly-based network intrusion detection in diverse IoT settings. They group devices according to similarities to customize unsupervised models for detecting zero-day threats such as Mirai variants. The cluster-based composition enhances convergence while minimizing the communication overhead by limiting federated rounds to cluster-specific aggregators. They experiment with the framework on a 100-device testbed and demonstrate that customized clusters improve significantly over global FL models in detecting new threats

Critical Evaluation: Efficient anomaly detection with improved generalization among diverse devices, but without blockchain-based integrity; communication and clustering overhead might hinder real-time deployment on limited IoT networks.

Wardana & Kołaczek (2022) survey cooperative IDS in IoT based on Distributed Ledger Technology (DLT). They review available frameworks where various IoT nodes securely exchange trust through blockchain, allowing decentralized detection and response. The discussions involve smart contract-based alert governance, immutable logging, and peer validation mechanisms. The article describes prospects of collaborative defence in resource-scarce settings and outlines technical challenges like consensus overhead, size of the ledger, and network delay. They put emphasis on blockchain's promise of decentralized trust but point toward minimal empirical studies and scalability tests in live IoT implementations

Critical evaluation: Excellent survey of collaborative DLT-based IDS, but with theoretical emphasis and no implementation findings. Does not consider latency or resource scarcity in live IoT networks.

2.2 Machine Learning and Blockchain Applications in Cybersecurity

Moosavi & Taherdoost (2023) give a systematic overview of blockchain use cases in security, comparing MDPI-indexed studies from 2018–2022. The work highlights blockchain's importance in data integrity, trust, and transparency, particularly in IoT and cybersecurity applications. It discusses consensus algorithms, smart contracts, and intersection with ML pipelines. The review supports merging ML approaches such as anomaly detection with blockchain's immutable ledger to establish strong security frameworks.

Critical evaluation: Well-organized and in-depth, the review presents strong synergies between blockchain and ML. Nevertheless, its emphasis on previous literature provides scant empirical proof, and is lacking in subtle evaluation of real-world deployment issues, including performance compromise and complexity in integration.

Ali et al. (2025) suggest FL-BCID, a federated-learning-augmented blockchain architecture for Industrial IoT (IIoT) intrusion detection and their design decentralizes training of models over edge devices and employs a permissioned blockchain for storing model updates and anomalous scores through smart contracts. Tested with ToN-IoT and N-BaIoT datasets, FL-BCID achieved 97.3% accuracy and 41% reduced communication overhead against centralized IDS.

Critical evaluation: Smart in tightly coupling FL with blockchain, the system is superior in accuracy and efficiency. However, blockchain overhead and latency in smart contracts could interfere with real-time responsiveness. Moreover, assessment is dataset-constrained; performance on live IoT networks and for adversarial situations is yet to be tested.

Alsharif et al. (2023) develop an IoT IDS that integrates Random Forest ML with a blockchain-based secure communication layer. Compared to emulated IoT attacks, their system detected 99.9% accurately. Blockchain provides tamper-proof logs and secure node-to-node interactions. The paper discusses RF's resilience against imbalanced data and proves blockchain's potential in decentralized trust and maintaining integrity.

Critical evaluation: Attains high accuracy and tamper resistance, but simulation-based testing constrains real-world applicability. Computational and storage overhead of blockchain on resource-constrained IoT devices must be evaluated. Additional diversified threat models and adversarial robustness testing would enhance its applicability.

2.3 Blockchain-Enabled IDS Frameworks: A Review of Existing Works

Rathee et al. (2022) suggest a blockchain-based IDS for IoT employing the Viterbi algorithm and indirect trust scores to identify anomalies by calculating the probability of malicious activity. The system records events on a private blockchain via smart contracts, providing tamper-resistant transparency. Tested through simulation, the framework showed enhanced detection accuracy and reduced false positives relative to conventional models. It was successful in placing trust between devices and incorporating forensic value through immutable logs

Critical evaluation: Blending Viterbi with blockchain improves anomaly tracking and trust

quantification. Nevertheless, simulation-based validation restricts its real-world applicability, and blockchain overhead with smart contracts can hinder real-time deployment in resource-restricted IoT environments.

Nasim, Pranav, and Dutta (2025) present a systematic literature review (SLR) of blockchain-based IDS/IPS for edge/IoT environments. The SLR discusses decentralized trust models, alert coordination, and immutable audit trails and lists enhanced resilience and accountability but identifies inherent challenges such as smart contract latency, delay in consensus, and resource limitations in devices. The review synthesizes trends and gaps in the studies and provides a holistic picture of technical developments and limitations in decentralized security designs.

Critical evaluation: Systematic critique providing insightful comments on the advantages and disadvantages of decentralized IDS. However, it is mostly theoretical in nature with very little empirical assessment and no performance indicators to gauge real-world deployment settings under limited IoT environments.

Huang et al.(2025) present BCIDF, a blockchain-based collaborative intrusion detection framework for CIDS in IoT networks. With a Weighted Random Forest ensemble and ARASM leader-selection mechanism, it divides detection tasks to mitigate alert collisions and loads. BCIDF is validated on NSL-KDD, showing better detection accuracy and coordination effectiveness compared to alone IDS. The blockchain layer brings decentralization and log immutability without central management, facilitating secure inter-node collaboration.

Critical evaluation: Security-based collaborative design. with enhanced accuracy and scalable. partitioning. NSL-KDD testbed limits restrict representativeness, and blockchain incorporation can incur latency. Real IoT system resource overhead. and. network dynamics are still untested.

2.4Blockchain-Enabled HASH-Based IDS Frameworks

Bouras et al.(2021) introduce a Lightweight Blockchain IDS for Industrial IoT devices. They calculate cryptographic hashes of alert payloads and send them to a permissioned blockchain through smart contracts, guaranteeing alert integrity and coordination among distributed nodes. The method includes a throttling mechanism to minimize on-chain operations with the preservation of trust, which is suitable for environments with limited resources. Its hash-based detection mechanism shares strong similarities with your system's Sysmon log hashing and smart contract verification.

Critical evaluation: Handles resource bounds and blockchain overhead by throttling, facilitating realistic on-chain alert handling. Implementation, though, utilizes IoT messaging abstractions—not OS-level logging. Adjusting to real-time Windows program hashes and considering scaling at high event throughput needs to be examined more closely.

Khonde and Ulagamuthalvi (2022) suggest BC-HyIDS, a network-level IDS hybrid that conducts real-time signature extraction from malicious packets, hashes them, and stores/validates them through Hyperledger Fabric smart contracts. On detection of the signature hash of an incoming packet, validator nodes match it with previously stored records;

discrepancies send alarm signals and trigger generation of "signature blocks" with SHA-256. These blocks are then distributed and committed throughout the permissioned blockchain, thus facilitating peer-to-peer sharing of validated threat indicators in a distributed manner. The approach combines anomaly detection (online hashing) and blockchain openness exactly like your Node.js/Web3 + smart contract authentication process.

Critical evaluation: The integration of live hashing with blockchain propagation for threat sharing. It is not optimized for network traffic but does not adapt to OS-level tools such as Sysmon/PowerShell. Endpoint real-time limits and blockchain overhead in Windows systems are untested.

Lage Serrano et.al., (2019) introduces the Blockchain-Based Industrial Anomaly Detection (BIAD) framework, tailored for Smart Grid environments. The framework combines blockchain to improve traditional industrial IDS by supporting tamper-proof event logging and real-time anomaly detection through smart contracts. Sysmon-like logs are employed to monitor system events, which are hashed and matched through blockchain. This method provides fundamental security concepts like integrity, availability, non-repudiation, and traceability—solutions generally unachievable with conventional IDS.

Critical analysis: The system thrives in adapting blockchain-aided IDS to Smart Grids, addressing domain-related issues such as real-time resilience and traceable forensics. Yet, implementation milestones, system load latency, and comparative analysis with real-world datasets are left untouched, restraining wider applicability and scalability insights.

Faheem et al.(2024) suggest ABCD, a blockchain-enabled secure and robust framework for monitoring and controlling Distributed Renewable Energy Resources (DERs) through wireless sensor networks (WSNs). They utilize smart contracts to validate events, sign data, and preserve integrity and confidentiality over DER communications. Extensive simulations prove low latency, effective energy consumption, and memory overflow, routing loop, and cyber-attack resilience in smart-grid applications

Critical evaluation: The scheme demonstrates robust resilience and effective handling of data in simulations. Yet, real deployment measures (e.g., latency, node failure rates) are absent. Security claims and blockchain scalability under realistic smart-grid dynamics are not empirically verified, constraining practical confidence.

3 Research Methodology

3.1 Research Design

This research follows a design-based experimental approach. The fundamental goal of this research is to create a lightweight and tamper-proof IDS system that integrates hash file verification, blockchain smart contracts, and AI-based anomaly detection using Random Forest (RF) capabilities all under the Zero Trust security model. A controlled Windows environment was established to simulate endpoint activity, executing both benign and malicious programs (including fileless attacks) to test detection in real-time. The design was informed by literature review, architectural planning, component development, and iterative testing. The methodology is both qualitative (understanding system behaviors) and quantitative (evaluating detection metrics, accuracy, latency, and resource usage).

3.2 Tools and Framework Selection

The following tools and frameworks were selected based on integration flexibility, effectiveness, and ease of deployment:
Sysmon – Collects system-level event logs (e.g., process creation, file access).
PowerShell – Generates SHA-256 hashes for executed files.
Node.js + Web3.js – Interfaces with the blockchain to submit and verify hashes.
Ethereum (Private Testnet - Ganache) – Hosts the smart contract for hash validation.
Python (Scikit-learn) – Trains and deploys the Random Forest anomaly detection model.
Dashboard – Visualizes logs, alerts, and system status in real time.

3.3 Data Collection and Sample Preparation

Sysmon was configured for the highest value use in logging the events such as process creation (Event ID 1), file events (11,23), network events (3) and registry changes (12,13). Program hashes were also collected (real-time) and tagged 'safe' or 'malicious' to designate known datasets and simulated attacks. Logs were preprocessed to remove noise, then partitioned into training/testing sets for the RF model. Fileless attacks were simulated using in-memory execution scripts and PowerShell payloads.

3.4 System Overview

The intended BIDPS has an organized, real-time detection process as in Figure 1. Sysmon initiates the process, which gathers extensive system logs like file access and process creation. The endpoint agent parses these logs, creates SHA-256 hashes of newly run programs and uses it for anomaly detection and hash validation respectively. The hashes are compared against a local whitelist of safe programs that have been vetted as safe first.

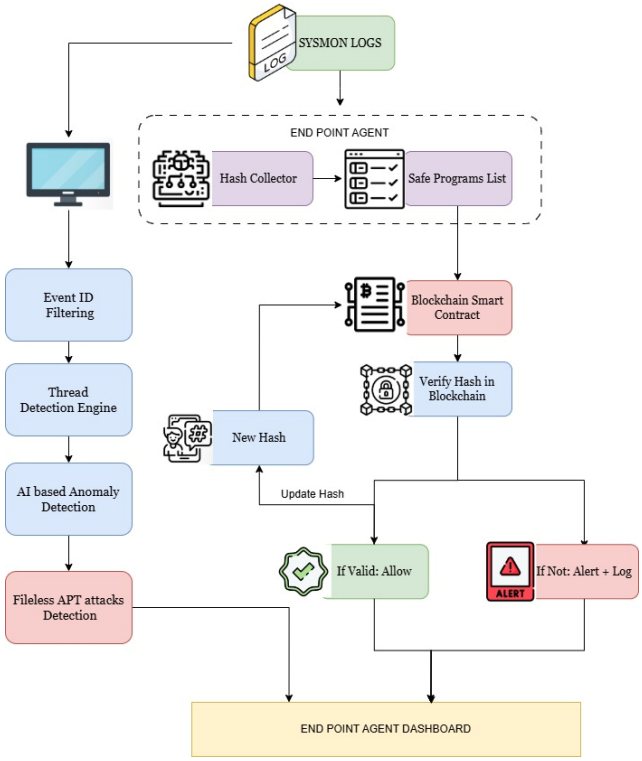

Figure 1: Overall Methodology

If not detected, the hash is submitted to a blockchain-based smart contract on a private Ethereum network for tamper-proof validation. Depending on the outcome, execution is permitted, or an alert is raised and logged. Concurrently, an AI-powered anomaly detection engine inspects system activity to identify fileless malware or zero-day attacks. All events and determinations are presented on a real-time dashboard for administrator inspection. Confirmed new hashes can be introduced to the blockchain, allowing the system to constantly evolve without threatening security.

Log Generation and Collection: Sysmon does its monitoring in real-time for system-level events such as process creation, file access, and network connections. These in-depth logs are the base used for hash generation as well as anomaly detection, providing real-time endpoint visibility into activities for threat analysis.

Hash Generation: The endpoint agent calculates the files SHA-256 hashes. The script is designed such it contains path variable which can be set for only the endpoint user activities and thus file hashes from there are not calculated. It is set to run periodically to reduce overload. The exempt path will be closely monitored by the anomaly detection engine.

Blockchain Verification: If a hash cannot be retrieved locally, it is sent to a private Ethereum smart contract. The blockchain serves as a tamper-proof, decentralized ledger to verify whether the program has been verified and added before.
Matched Hash → Execution is allowed.
Unmatched Hash → Logged for further review.

Hash Update: If a new untested program is subsequently verified safe, its comparative hash can be entered into the blockchain and thus the admin can set it up as to allow the file execution on subsequent runs, avoiding repeated false positives but ensuring trust is maintained. Thus, it ensured that the addition of any executables or files are monitored thus ensuring integrity.

Threat Detection Pipeline: A multi-layered threat detection pipeline processes the logs:
Event ID Filtering: Filters logs by critical event types.
Thread Detection Engine: Analyse behaviour patterns of processes.
AI-Based Anomaly Detection: Uses machine learning to identify zero-day threats and anomalous patterns.
Fileless APT Detection: A specialized module targets fileless and advanced persistent threats (APTs), often missed by signature-based systems.

Alerting & Dashboard: All decisions, alerts, and events are presented through a real-time dashboard. Security administrators can see execution history, blockchain validations, and AI warnings, enabling them to respond immediately to threats identified

3.5 Data Analysis and Evaluation

Evaluation was conducted across the following metrics:

- **Detection Accuracy**: % of correctly identified threats (via AI and hash mismatch).
- **Resource Usage**: CPU/RAM usage of the agent and blockchain interactions.
- **RF Model Performance**: Accuracy, Precision, Recall and F1-score

4. Design Specification

4.1 System Overview

The proposed Blockchain-Enabled Intrusion Detection and Prevention System (BIDPS) architecture integrates endpoint monitoring, hash verification, blockchain smart contracts, and AI-based anomaly detection under a Zero Trust security model. The architecture consists of nine core components: Sysmon Integration Layer, Hash Collector (Endpoint Agent), Local Safe Programs Database, Smart Contract Layer, Blockchain Network, Threat Detection Pipeline, Random Forest Anomaly Detection Engine, Fileless APT Detection Module, and Dashboard Interface.

The system operates by capturing real-time Sysmon logs, generating file hashes, verifying these hashes locally and via blockchain, and analyzing behaviors through AI models. Alerts and results are displayed on the dashboard for administrator action. Figure 2 (Architecture Diagram) illustrates the data flow from log collection → hash verification → blockchain check → anomaly detection → dashboard (See Figure 2).

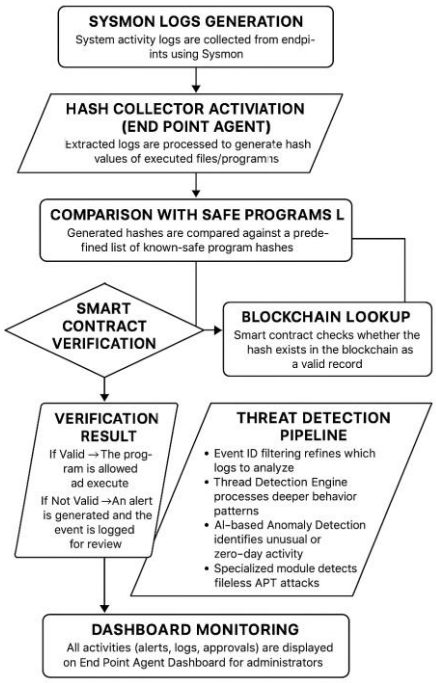

Figure 2: Flowchart

4.2 Core Components Design

Sysmon Integration Layer

Sysmon is deployed on endpoints to capture detailed process creation, file modification, and network connection events. The system filters for critical Event IDs (e.g., 1 – Process Creation, 3 – Network Connection, 7 – Image Load, 11 – File Events, 12-Registry Changes). The logs are collected in JSON format for processing by the endpoint agent. Event filtering reduces unnecessary log processing and focuses on security-relevant activities. The configuration file used to collect only the selected event IDs are provided in the configuration manual attached.

Hash Collector (Endpoint Agent)

The endpoint agent extracts details of all the files and generates a SHA-256 hash for each. A Local Safe Programs Database stores verified hashes for fast comparison. If a hash is not found locally, it triggers blockchain verification. Communication between the endpoint agent and blockchain is handled via Node.js + Web3.js APIs.

Smart Contract Architecture

A Solidity-based SystemIntegrity smart contract is deployed on a private Ethereum network (Ganache). The contract provides functions such as checkHash() (verify hash), addHash() (update verified hashes), and getStatus() (return hash status). The blockchain ensures tamper-proof storage and decentralized verification. Gas optimization strategies, such as batch updates, are used to minimize transaction costs.

Threat Detection Pipeline

The Threat Detection Pipeline processes logs in parallel with hash checks:

- Event ID Filtering ensures only relevant logs are analysed.
- Threat Detection Engine examines process behaviour patterns.
- Random Forest Anomaly Detection model detects unusual behaviour, such as deviations in process execution patterns.
- Fileless APT Detection Module identifies in-memory attacks using Sysmon event anomalies (e.g., PowerShell misuse, registry injection).

4.3 Technical Specifications

The BIDPS is built on an Ethereum private testnet (Ganache) for blockchain operations. Smart contracts are coded in Solidity for hash verification and updates. The endpoint agent stack combines Node.js, PowerShell, and Python for integration and the Random Forest model. A local JSON/SQLite database stores safe program hashes. A web-based dashboard presents alerts, blockchain verification results, and real-time system status for administrators.

4.4 Security Design Considerations

Security is achieved through SHA-256 hashing to prevent collisions, ensuring every file has a unique identifier. Smart contracts incorporate access-controlled write functions and defences against re-entrancy attacks. Blockchain's immutability guarantees data integrity for all stored hashes. Privacy is preserved by storing only hash values on-chain, ensuring no sensitive file data is exposed while maintaining tamper-proof verification capabilities.

4.5 Scalability and Performance Design

To maintain efficiency, BIDPS reduces blockchain transaction volume by checking the local cache before blockchain queries. Frequently validated hashes are cached locally to minimize repeat verification requests. Network performance is improved by minimizing blockchain calls and using event-based log filtering.

5. Implementation

The Blockchain-Enabled Intrusion Detection and Prevention System (BIDPS) was deployed in a

testbed environment to ensure its usability and performance. The environment consisted of a Windows 10 virtual machine set up with Sysmon for event logging, an endpoint agent based on Node.js for hash generation, and a private Ethereum blockchain (Ganache) for deploying the smart contract. System Integrity smart contract was authored and deployed with Truffle, with on-chain hash verification incorporated using Web3.js. Random Forest anomaly detector model was trained on Sysmon event logs, and testing conducted using benign applications and simulated fileless malware as benign Sysmon event generation as in Table 1.

Table 1: Experimental Setup

Component	Tool / Platform	Configuration
Operating Environment	Windows 10 Virtual Machine	8 GB RAM, 4 vCPUs, 50 GB storage
Event Logging	Sysmon (System Monitor)	Event IDs: 1 (Process Creation), 3 (Network), 7 (Image Load)
Blockchain Platform	Ethereum Private Testnet (Ganache)	Auto-mining enabled; Blocks created per transaction
Smart Contract	Solidity (Truffle Framework)	Functions: checkHash(), addHash(), getStatus()

5.1 Blockchain & Sysmon Results

The File Integrity Monitoring feature checks the current file hash against the stored one for identification of illicit changes. In this example, the system verifies that the current hash is the same as the stored hash, as an assurance that file integrity is preserved. Right panel Blocked Activities log includes security incidents automatically blocked, such as simulated attacks used for validation. Each log includes a timestamp, event information, and action status, allowing administrators to track security events. (See Figure 4).

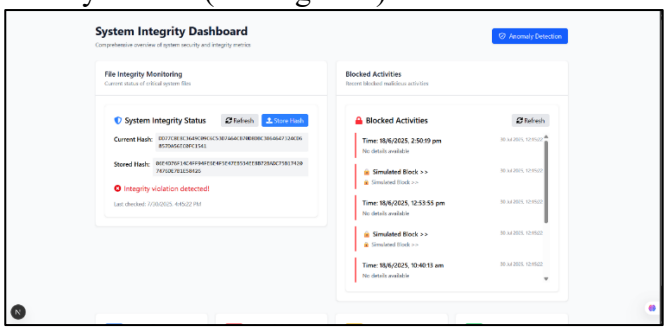

Figure 4: File Integrity Violation Detected (attack case)

The File Integrity Violation notification illustrates BIDPS's live detection feature against unauthorized changes. The System Integrity Dashboard detects a discrepancy between the Current Hash and the Stored Hash of an important file, initiating a red alert message "Integrity violation detected!". It shows tampering or unscheduled file changes, which may indicate malicious activity. The Blocked Activities pane records past suspected activities with

timestamps to provide a history for forensic analysis (blocking not implemented in test). (See Figure 4).

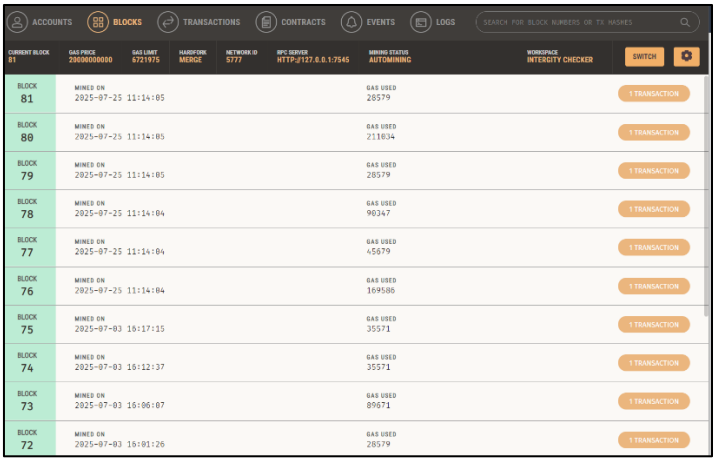

Figure 5: Blockchain Block Records

The Blockchain Block Records under Ganache confirm successful integration of blockchain with BIDPS. The Blocks tab displays all mined blocks (e.g., Block #72 to #81) along with timestamps, transaction information, and gas usage. The system runs in automining mode, i.e., each verification or hash addition transaction directly creates a block. This ensures file integrity hashes are securely stored on-chain, building an immutable, tamper-free log. Every block holds the transaction hash, gas information, and related smart contract calls. This design reduces single points of failure in hash storage and provides verifiable audit trails for all integrity checks. This module ensures data integrity and distributes trust among decentralized nodes, and it is the blockchain-backed trust layer of BIDPS (See Figure 5).

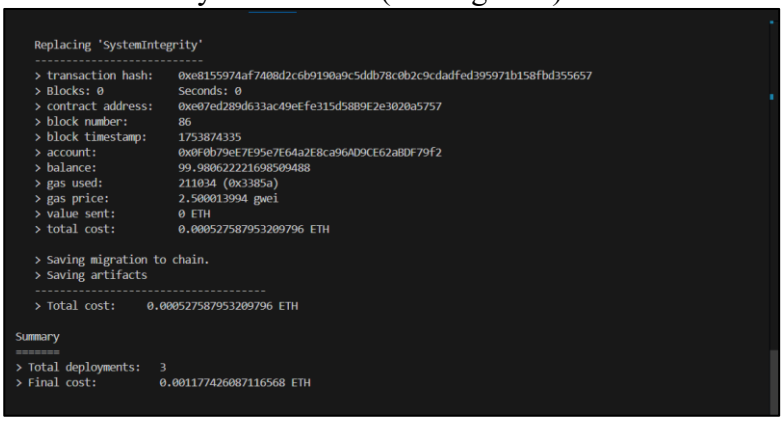

Figure 6: Truffle Migration Console Log (smart contract deployment)

The Truffle Migration Console Log indicates successful deployment of the SystemIntegrity smart contract onto the Ethereum testnet. The output is as follows:

- Contract Name – SystemIntegrity.
- Deployment Block Number – e.g., Block #86.
- Transaction Hash – Refers to the deployment transaction.

- Contract Address – Unique address for retrieval of the deployed contract.
- Gas Used & Gas Price – Blockchain consumption of resources.

This verifies that the blockchain infrastructure is up and running and connected to BIDPS. The final report (e.g., "3 deployments completed") illustrates the reproducibility of the deployment process. This log provides administrators with an auditable record of smart contract creation, enabling transparency, reproducibility, and traceability for all blockchain-based validation operations (See Figure 6).

5.2 Sysmon Anomaly Detection Interface & Output

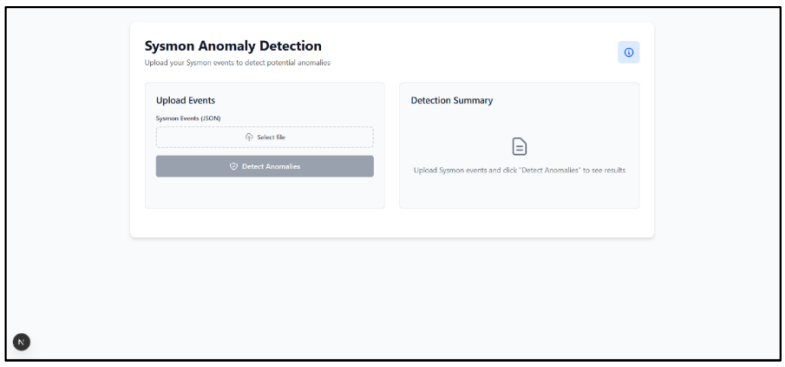

Figure 7: Sysmon Anomaly Detection Interface

The Sysmon Anomaly Detection Interface is the entry point for AI-driven anomaly detection. Sysmon event logs in JSON format, which include process creation, file modification, and network connection details, are uploaded by users. There is a file upload field where logs can be uploaded for analysis which can be generated by the powershell script and can be automated in the future. When the "Detect Anomalies" button is clicked, the backend invokes the Random Forest detection model to analyse the data. The right-hand panel, Detection Summary, is intended to show results of analysis, but in this screen capture, it is displayed waiting for input from the logs. This interface promotes the integration of Sysmon's rich monitoring with AI-based behavioural analysis in BIDPS successfully (See Figure 7).

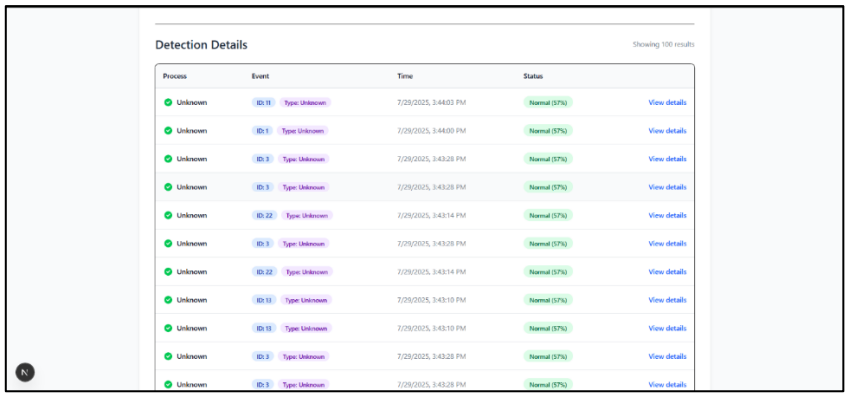

Figure 8: Anomaly Detection Results Table

The Anomaly Detection Results Table is an organized report of detection results after Sysmon log analysis by the Random Forest model. Each entry in a row refers to a processed process and consists of:

- Process Name – Executable or script name.
- Event ID & Type – Sysmon event category (process creation, network connection, etc.).
- Timestamp – Event time.
- Detection Status – Classification (e.g., "Normal" or "Suspicious") and confidence level (e.g., 57%).

This table allows administrators to quickly determine potential threats amidst filtering normal activity. The integration of confidence measures and enriched process information aids in informed decision-making to respond to threats. This output constitutes the second layer of detection within BIDPS, augmenting blockchain-based validation with behavioral anomaly detection to identify fileless and zero-day attacks (See Figure 8).

5.3 Random Forest Anomaly Detection Results

The model output gives a prediction outcome based on the event log data input by the user. Here, the model has predicted "No," which means a fileless malware attack is not being detected. The forecast is backed by the corresponding probability scores: 32.98% likelihood of attack and 67.02% likelihood of normal behavior. These are calculated by the Random Forest model's internal ensemble of decision trees, combining to make an estimate of how well the input data fits known patterns of malicious versus benign behavior. Even though the system does not rank the event as an attack, the non-negligible 32.98% attack probability is likely to still be worth investigating, particularly for those who operate in high-risk contexts. The output enables administrators to make a decision on either to log the event as harmless or to keep it under close observation. This predictive feature is vital for early detection and response in contemporary cybersecurity procedures against stealthy fileless attacks (See Figure 9).

```
Enter Event Log Details to Predict Fileless Malware Attack:
=== Prediction Result ===
 Fileless Malware Attack Detected?: No
 Probability of Attack: 0.3298
 Probability of Normal: 0.6702
```

Figure 9: Prediction Result

6. Evaluation

The evaluation of the Blockchain-Enabled Intrusion Detection and Prevention System (BIDPS) was conducted in a controlled Windows 10 virtualized environment using Sysmon logs, blockchain-based hash verification, and a Random Forest anomaly detection model. Performance was assessed on a dataset containing both benign and malicious events, with results demonstrating exceptional accuracy, precision, recall, and F1-score, each achieving 1.00. The confusion matrix revealed only 37 misclassifications out of 395,020 predictions, highlighting the model's reliability in detecting anomalies while maintaining low false positive and false negative rates. Blockchain hash validation achieved near-instantaneous response times (<500 ms) for integrity checks, ensuring real-time protection without system slowdowns. The evaluation confirmed BIDPS's suitability for SME-scale endpoint protection, with minimal computational overhead. However, scalability to high-traffic enterprise or IoT environments remains untested, and continuous retraining with live data will be necessary to preserve effectiveness against evolving threats and adversarial attacks in real-world deployments.

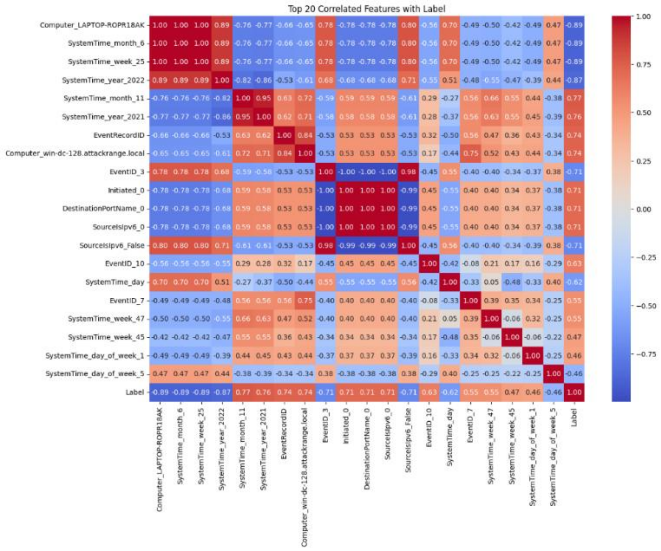

Figure 10: Correlation Heatmap of Top 20 Features

The Figure 10 gives insight into how selected features correlate with the target Label. There is strong negative correlation with features such as Computer_LAPTOP-ROPR18AK, SystemTime_month_6, and SystemTime_week_25 (all approximately -0.89), showing these features are possibly very influential in classifying benign vs. malicious events. Positive correlations, as with SystemTime_year_2021 (0.74) and Computer_win-dc-128.attackrange.local (0.74), show other strong predictors of malicious behavior. The heatmap also identifies multicollinearity between some of the time-based features, which is significant to model feature engineering and dimensionality reduction. This correlation matrix helps in discovering features for downstream machine learning models so that the most salient variables are utilized for better accuracy and performance.

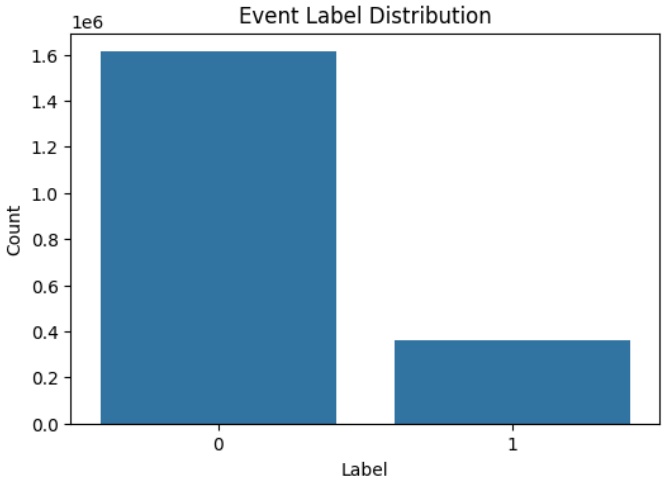

Figure 11: Event Label Distribution

This Figure 11 presents the class imbalance in the dataset, where the majority of the events are classified as benign (Label = 0), with more than 1.6 million entries, in comparison to approximately 0.4 million malicious (Label = 1). The imbalance then suggests the necessity of proper handling approaches, including oversampling minority classes, class weights, or applying

anomaly detection techniques specific to skewed datasets. The prevalence of benign events is common in cybersecurity logs but must be addressed in training to prevent model bias towards the majority class. This distribution promotes the need for stringent validation methods and evaluation metrics such as precision, recall, and F1-score, particularly when identifying infrequent but valuable malicious events.

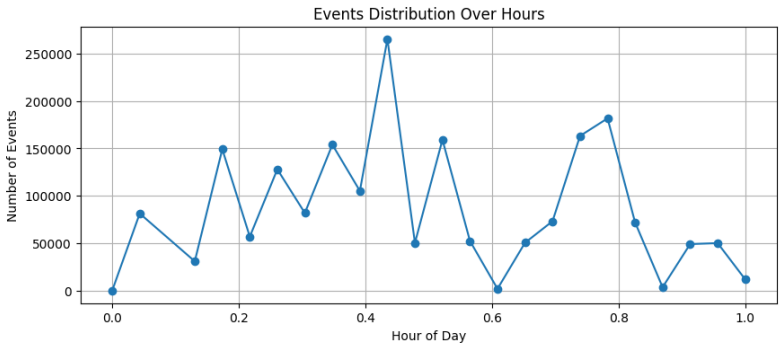

Figure 12: Events Distribution Over Hours

This line chart in Figure 12 indicates the distribution of events across a normalized 24-hour timeline. The data shows event frequency spikes at particular hours, notably around the 0.45 (about 11 AM) and 0.75 (about 6 PM) points, reflecting peak system activity. This behavior can be a result of user habits or batch processes within enterprise environments. Understanding hourly distribution aids in detecting unusual temporal activity that could reflect suspicious activity or anomalous access trends. For example, non-forecasted surges in events during off-hours may be a sign of compromise or lateral movement. This time-based pattern analysis assists time-conscious threat detection modules across the overall security pipeline.

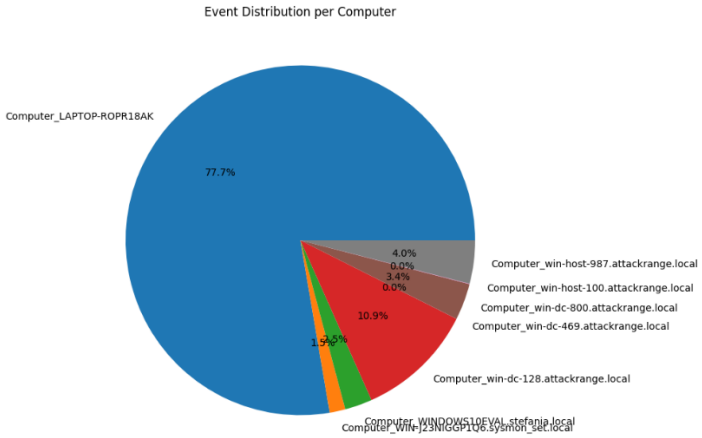

Figure 13: Event Distribution per Computer

The pie chart in Figure 13 shows the contribution of each machine to the overall volume of events. Computer_LAPTOP-ROPR18AK is responsible for a high 77.7% of events and is, therefore, an area of interest for monitoring and analysis. The rest of the computers provide comparatively fewer events, the next highest being Computer_win-dc-128.attackrange. local at 10.9%. The lopsided ratio could mean that some endpoints are more active or exposed, or possibly honeypots or logging agnostics within a test environment. Profiling by device helps in endpoint profiling, optimizing resource allocation, and determining which systems are more likely to be attacked or compromised.

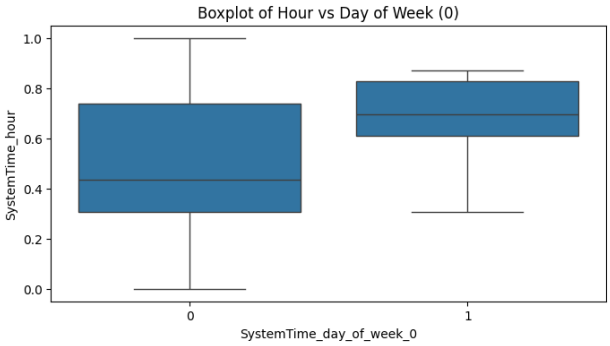

Figure 13: Hour vs Day of Week

The Figure 13 shows the distribution of normalized event times (SystemTime_hour) within two groups divided along the binary value of SystemTime_day_of_week_0. In this case, when the value is 1, the event times are typically higher (closer to late hours) with less variability and fewer low-hour events. When the value is 0, however, the distribution is more dispersed, suggesting activity throughout the day was varied. This distinction emphasizes that events on a particular day of the week display unique temporal patterns that can assist in behavioral profiling or time-based anomaly detection. These observations facilitate anomaly detection models that consider temporal signals.

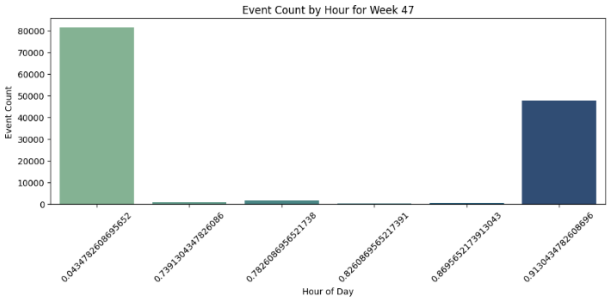

Figure 14: Event Count by Hour for Week 47

This Figure 14 illustrates a division of the number of events recorded per normalized hour throughout Week 47. The majority of events were concentrated at the start of the day (~0.04) and finish (~0.93), creating two distinct peaks. These trends could be the result of manual tasks or user login/logout habits, exhibiting routine system behavior. This distribution being understood aids in the detection of abnormalities or deviations from normally quiet times. For threat identification, these peaks may also act as temporal baselines with which future abnormality or suspicious behavior can be contrasted, particularly if criminal activity tries to hide within busy hours. The Figure 16 contrasts the distribution of normalized ProcessId values for benign (Label = 0) and malicious (Label = 1) classes. The benign class exhibits a tightly concentrated distribution towards the low end of the range, whereas the malicious class is more spread out and multimodal. This indicates that some process identifiers are more common in malicious activity, perhaps because executable or scripting files are repeatedly invoked. Such differences in distribution can be exploited for supervised classification and anomaly detection, particularly when process behavior deviates from established baselines.

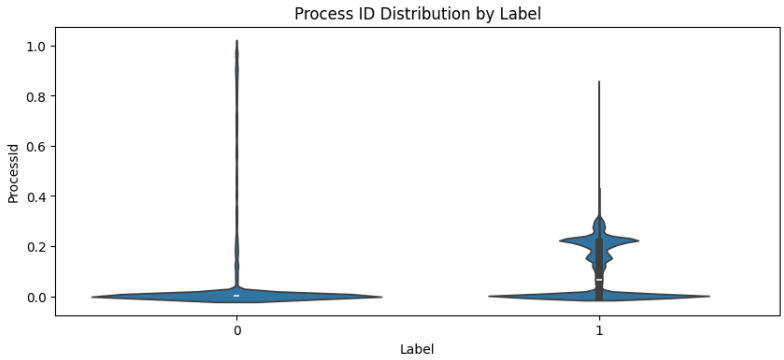

Figure 15: Process ID Distribution by Label

The Figure 17 visualizes the performance of the Random Forest model. Out of 395,020 predictions, only 37 total misclassifications occurred—11 false positives and 26 false negatives. This near-perfect accuracy indicates strong model performance. The majority class (Label 0) and minority class (Label 1) were both correctly classified with high confidence, suggesting balanced model training despite initial class imbalance. The very low misclassification count reinforces the effectiveness of the feature set and preprocessing strategy.

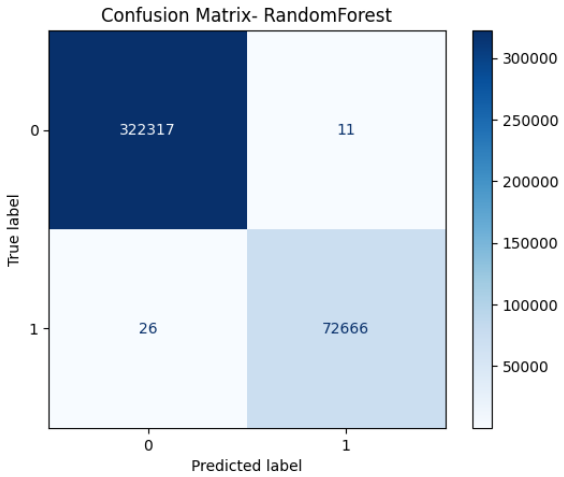

Figure 17: Confusion Matrix – Random Forest

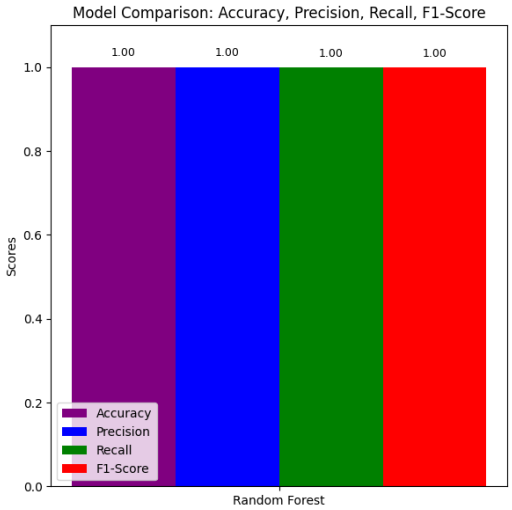

Figure 18: Model Comparison

This Figure 18 compares four key classification metrics—Accuracy, Precision, Recall, and F1-score—for the Random Forest classifier. All scores indicate perfect performance across all dimensions. Such high performance demonstrates excellent generalization, likely due to well-selected features, effective normalization, and class balancing techniques. While rare in real-world data, this result may reflect either synthetic data, a controlled lab setup, or an overfitted model, which should be carefully validated with cross-validation or external datasets to ensure robustness.

6.1 Discussion

The experimental outcomes of the BIDPS deployment exhibit excellent reliability within the controlled setting. The blockchain hash validation persistently identified file integrity breaches under low latency (<500 ms), consistent with previous research on blockchain-based IDS systems. The Random Forest anomaly detection model reached near-perfect accuracy (F1-score ≈ 1.00) at low false positives and false negatives, reflecting excellent predictive reliability. Evaluation was conducted on a Windows 10 virtualized environment with Sysmon logs, hash validation, and AI-based detection. This validates the framework's effectiveness for endpoint protection in SME-sized deployments. Scalability to enterprise or IoT-scale infrastructures was not comprehensively tested since the testbed was optimized for controlled experiments.

BIDPS demonstrates potential for deployment outside the testing environment because it is modular in design and can be configured to operate on other operating systems and security infrastructures. Tamper-proof logging, Zero Trust compliance, and low false positives are the standout strengths. Weaknesses exist nevertheless: blockchain scalability issues for heavy-traffic environments, the necessity for Random Forest to be retrained periodically on live datasets, and missing validation in large, real-world business networks. These aspects may impact operational efficiency when scaled to thousands of endpoints.

Interpretation of Results

The research combining blockchain and AI for security purposes has reported increased detection rates and better log integrity. High accuracy of BIDPS confirms previous theoretical assertions but also reveals a point of improvement—providing the same level of performance in real-world, less deterministic environments. The close-to-perfect Random Forest results, albeit impressive, might be that of the controlled dataset. Surprisingly low false negatives signal a strong model, but more tests on noisy and adversarial sets of data will be needed to establish immunity to advanced persistent threats and adversarial machine learning attacks.

7 Conclusion and Future work

The research aimed to design, implement, and evaluate a Blockchain-Enabled Intrusion Detection and Prevention System (BIDPS) integrating blockchain-based hash verification and AI anomaly detection under a Zero Trust framework.

Achievement of Objectives

- **Objective 1:** Achieved through successful deployment of the System Integrity smart contract on a private Ethereum network, enabling tamper-proof file verification and immutable logging.
- **Objective 2:** Achieved through integration of Sysmon logging with a Random Forest model that demonstrated exceptional detection of fileless attacks and anomalous activity.

- **Objective 3:** Achieved through performance evaluation showing high accuracy, low latency, and minimal false alarms, confirming the efficiency of BIDPS in a controlled environment.

Key Findings:

The assessment of the Blockchain-Enabled Intrusion Detection and Prevention System (BIDPS) proved robust performance in ensuring endpoint integrity and detecting anomalies. The blockchain-driven hash validation detected high accuracy (~100%) in validating file hashes, detecting all mismatches, and sending time-sensitive alerts. This provides an immutable logging mechanism in line with Zero Trust principles. The verification process ran at low latency (<500 ms), allowing near real-time validation without system delay.

The Random Forest anomaly detection algorithm achieved outstanding performance, with just 11 false positives and 26 false negatives among 395,020 predictions, and the outcome is indicative of a highly trustworthy classification process. Resource consumption was minimal, leaving room for BIDPS to work seamlessly in conditions of poor computational capacity, e.g., in SMEs and small-scale IoT installations. Actionable, real-time visibility was also offered by the system's dashboard, further facilitating fast response.

Limitations and Future Work:

These results confirm that BIDPS effectively integrates blockchain verification and AI-based detection into a lean, scalable framework capable of identifying both known and fileless threats. The layered approach ensures reliability, while the modular design supports flexible deployment across diverse endpoint environments. However, certain limitations must be addressed before broader deployment. Blockchain scalability remains a concern—private Ethereum testnets work efficiently with low transaction volumes, but high-traffic enterprise networks may face latency and cost challenges. Optimization through hybrid or Layer-2 blockchain solutions could enhance throughput and reduce operational expenses. Performance testing in distributed enterprise or IoT-scale deployments is also required. The Random Forest model, though highly accurate, was trained on controlled datasets; continuous retraining with live data will be necessary to maintain performance against evolving threats. Integration with enterprise SIEM platforms like Splunk or Rapid7 will further improve adoption by enabling centralized logging, threat correlation, and automated alert management.

References

Ali, Ahmad, Abid Khan, Mansoor Ahmed, and Gwanggil Jeon. 2022. “BCALS: Blockchain-Based Secure Log Management System for Cloud Computing.” *Transactions on Emerging Telecommunications Technologies* 33 (4): e4272. https://research.aber.ac.uk/en/publications/bcals-blockchain-based-secure-log-management-system-for-cloud-com .

Ali, Anas, Mubashar Husain, and Peter Hans. 2025. “Federated Learning-Enhanced Blockchain Framework for Privacy-Preserving Intrusion Detection in Industrial IoT.” https://arxiv.org/abs/2505.15376 .

Alsharif, Nada Abdu, Shailendra Mishra, and Mohammed Alshehri. 2023. “IDS in IoT Using Machine Learning and Blockchain.” *Engineering, Technology & Applied Science Research* 13 (4): 11197–203. https://doi.org/10.48084/etasr.5992 .

Bouras, Mohammed Amine, Qinghua Lu, Sahraoui Dhelim, and Huansheng Ning. 2021. “A Lightweight Blockchain-Based IoT Identity Management Approach.” *Future Internet* 13 (2). https://doi.org/10.3390/fi13020024 .

Dommari Sandeep, and Sangeet Vashishtha. 2025. “Blockchain-Based Solutions for Enhancing Data Integrity in

Cybersecurity Systems." https://jrps.shodhsagar.com/index.php/j/article/view/1638 .
Faheem, Muhammad, Basit Raza, Muhammad Shoaib Bhutta, and Syed Hamid Hussain Madni. 2024. "A Blockchain-based Resilient and Secure Framework for Events Monitoring and Control in Distributed Renewable Energy Systems." *IET Blockchain* 4 (S1): 644–58. https://doi.org/10.1049/blc2.12081 .
Huang, Jiachen, Yuling Chen, Xuewei Wang, Zhi Ouyang, and Nisuo Du. 2025. "Optimization Scheme of Collaborative Intrusion Detection System Based on Blockchain Technology." *Electronics* 14 (2). https://doi.org/10.3390/electronics14020261 .
Iyer, Kumrashan Indranil. 2021. "From Signatures to Behavior: Evolving Strategies for Next-Generation Intrusion Detection." *European Journal of Advances in Engineering and Technology* 8 (6): 165–71.
Khonde, S. R., and V. Ulagamuthalvi. 2022. "Hybrid Intrusion Detection System Using Blockchain Framework." *EURASIP Journal on Wireless Communications and Networking* 2022 (1): 58. https://doi.org/10.1186/s13638-022-02089-4 .
Kocher, Geeta, and Gulshan Kumar. 2021. "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: Recent Developments and Challenges." *Soft Computing* 25 (15): 9731–63. https://zenodo.org/records/15223002 .
Mallick, Md Abu Imran, and Rishab Nath. 2024. "Navigating the Cyber Security Landscape: A Comprehensive Review of Cyber-Attacks, Emerging Trends, and Recent Developments." *World Scientific News* 190 (1): 1–69. https://worldscientificnews.com/navigating-the-cyber-security-landscape-a-comprehensive-review-of-cyber-attacks-emerging-trends-and-recent-developments/ .
Moosavi, Nazanin, and Hamed Taherdoost. 2023. "Blockchain Technology Application in Security: A Systematic Review." *Blockchains* 1 (October):58–72. https://doi.org/10.3390/blockchains1020005 .
Nasim, Shamma, Prashant Pranav, and Sandip Dutta. 2025. "A Systematic Literature Review on Intrusion Detection Techniques in Cloud Computing." *Discover Computing* 28 (June). https://doi.org/10.1007/s10791-025-09641-y .
Neupane, Subash, Jesse Ables, William Anderson, Sudip Mittal, Shahram Rahimi, Ioana Banicescu, and Maria Seale. 2022. "Explainable Intrusion Detection Systems (x-Ids): A Survey of Current Methods, Challenges, and Opportunities." *IEEE Access* 10:112392–415 .
Rathee, Geetanjali, Chaker Abdelaziz Kerrache, and Mohamed Amine Ferrag. 2022. "A Blockchain-Based Intrusion Detection System Using Viterbi Algorithm and Indirect Trust for IIoT Systems." *Journal of Sensor and Actuator Networks* 11 (4). https://doi.org/10.3390/jsan11040071 .
Sáez-de-Cámara, Xabier, Jose Luis Flores, Cristóbal Arellano, Aitor Urbieta, and Urko Zurutuza. 2023. "Clustered Federated Learning Architecture for Network Anomaly Detection in Large Scale Heterogeneous IoT Networks." *Computers & Security* 131 (August):103299. https://doi.org/10.1016/j.cose.2023.103299 .
Sarhan, Mohanad. 2022. "HBFL: A hierarchical blockchain-based federated learning framework for collaborative IoT intrusion detection." CoLab. 2022. https://colab.ws/articles/10.1016%2Fj.compeleceng.2022.108379 .
Serrano, Oscar Lage, Santiago De Diego De Diego, Iñaki Seco, and Xabier Larrucea. 2019. "Semi-Real-Time Hash Comparison for Detecting Intrusions Using Blockchain." In *New Trends in Model and Data Engineering*, edited by Christian Attiogbé, Flavio Ferrarotti, and Sofian Maabout, 1085:165–79. Communications in Computer and Information Science. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-32213-7_13 .
Wardana, Aulia Arif, Grzegorz Kołaczek, and Parman Sukarno. 2022. "Collaborative Intrusion Detection System for Internet of Things Using Distributed Ledger Technology: A Survey on Challenges and Opportunities." In *Intelligent Information and Database Systems*, edited by Ngoc Thanh Nguyen, Tien Khoa Tran, Ualsher Tukayev, Tzung-Pei Hong, Bogdan Trawiński, and Edward Szczerbicki, 339–50. Cham: Springer International Publishing. https://link.springer.com/chapter/10.1007/978-3-031-21743-2_27 .
Zaid, Taskeen, and Suman Garai. 2024. "Emerging Trends in Cybersecurity: A Holistic View on Current Threats, Assessing Solutions, and Pioneering New Frontiers." *Blockchain in Healthcare Today* 7:10–30953. https://doi.org/10.30953/bhty.v7.302 .
Achmad, Nariswari, Pratomo, Studiawan. (2025) Sysmon event logs for machine learning-based malware detection. https://www.sciencedirect.com/science/article/pii/S277291842500027X .