

Article

Not peer-reviewed version

---

# Block Chain -Integrated IDS: A Decentralized Approach to Threat Detection and Logging

---

[R. Karthick](#)\*

Posted Date: 11 August 2025

doi: 10.20944/preprints202508.0482.v1

Keywords: intrusion detection; Hyperledger Fabric; blockchain; threat management



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

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

Article

# Block Chain -Integrated IDS: A Decentralized Approach to Threat Detection and Logging

R. Karthick

Department of CSE, K.L.N. College of Engineering, Sivaganga, India; karthickkiwi@gmail.com

## Abstract

Nowadays, the propagation of cyber threats has increased in both scale and complexity, rendering Intrusion Detection Systems (IDS) a common target for evasive attacks. Historical IDS software frequently used a Great Wall of China approach that created central points of failure. In this paper, we propose a Block chain -enabled Intrusion Detection System (BIDS) which employs Block chain to improve security, transparency and resilience in the context of IDS frameworks. BIDS incorporates distributed ledger techniques for tamper-evident log storage, relies on smart contracts for automatic threat interceptions and adopts the Practical BFT (Practical Byzantine Fault Tolerance) consensus algorithm to provide efficient and verifiable event validation. To explore BIDS, we implement and evaluate it using the Suricata IDS, Hyperledger Fabric Block chain platform for authorization mechanisms, we use public intrusion datasets (CICIDS2017 and NSL-KDD) within a simulated enterprise network. From the empirical results, we prove that BIDS could provide detection accuracy as high as conventional IDS systems, yet substantially enhances the integrity of logs under abuse and recourse to log reading facilities: counter indication towards manipulation is more effective with less effort compared with OPSA, while it defeats all pre-inspection on assurance (see Sect. 4.5), due to structural alterations in data space and accommodation to incremental computation.

**Keywords:** intrusion detection; Hyperledger Fabric; blockchain; threat management

---

## 1. Introduction

With the increasing interconnectivity of digital systems, organizations are finding it more difficult to keep pace with the deluge of counterattacks like malware and denial-of-service (DoS) attacks that a growing number today evolve into advanced persistent threats (APTs). IDS – Intrusion Detection Systems are a first line of defense, watching over the activities in system and network to find out whether anyone is engaged into any kind of malicious activities think for this like notifications to administrators about potential threats. As advanced as IDS technologies have become and even with the addition of machine learning and behavioral analysis for detection, they still are exposed in one critical area – not our actual data.

The vast majority of IDS solutions are a management nightmare due to being centralized systems where the logs, and alerts sit in one central place. In a nutshell, this centralized structure makes them easy to back up but also insecure, due to insider threats, post-compromise log tampering and log deletion. When privileged access is compromised, the historical records themselves are no longer guaranteed to be intact – thus impacting both forensic integrity and confidence in the system.

This is where Block chain technology, initially created to guarantee secure and decentralized digital currency systems, comes in. By italicizing these features – immutability, transparency and distributed consensus— it seems to suggest that the block chain is seen as a concrete, reliable technology for IDS log management and corresponding replies. Organizations can deploy IDS with permission Block chain like Hyperledger Fabric, to enable decentralized threat logging and immutable record keeping, and allow smart contracts to activate automated response.

In this paper, we introduce BIDS (Block chain -based Intrusion Detection System), which stands as a synergy between the traditional IDS functions and security guarantees of Block chain . To achieve

this, we construct a modular architecture with IDS agents for monitoring, consortium Block chain layer for decentralized logging and smart contracts to enforce response policies. We then evaluate the system on a set of real-world datasets that represent various communication patterns across more than tens of thousands of machines using the network simulation. We show that integration into Block chain systems improves trustworthiness of an IDS without trading off much in terms of computational burden and latency to make BIDS a feasible alternative to conventional approaches (Practical & Secure).

## 2. Related Works

This paper presents a survey on Intrusion Detection Systems (IDS) which are currently playing the main role of the defense line against cyber threats. Traditional IDS solutions are broadly defined into: host-based systems (HIDS), network-based systems (NIDS) which can be classified as signature-based detection or anomaly-based models [1–5]. While signature-based approaches are effective, they can only match with known attack patterns and anomaly detection approaches, which could use statistical [6], machine learning models or other techniques are usually better to perform in the face of zero-day threats. However, both have drawbacks in scalability, centralization and tamper resistance (respectively).

The latest innovation in the Block chain technology provides opportunity for setting up decentralized cybersecurity system – [11–15]. The fundamental characteristics of Block chain, specifically immutability [16], transparency and distributed trust appeal to intrusion detection applications particularly in scenarios where log integrity and audibility is fundamentally important [17–20]. Enterprise or permissiveness Block chain s like Hyperledger Fabric, meanwhile, are becoming a popular choice in the corporation-centric setting for their access control capabilities, molecularity, and high throughput [16–19]. Block chain has already been demonstrated to support tamper resistance and insider threat protections in various works [26–30] that proposed secure logging mechanisms backed by Block chain.

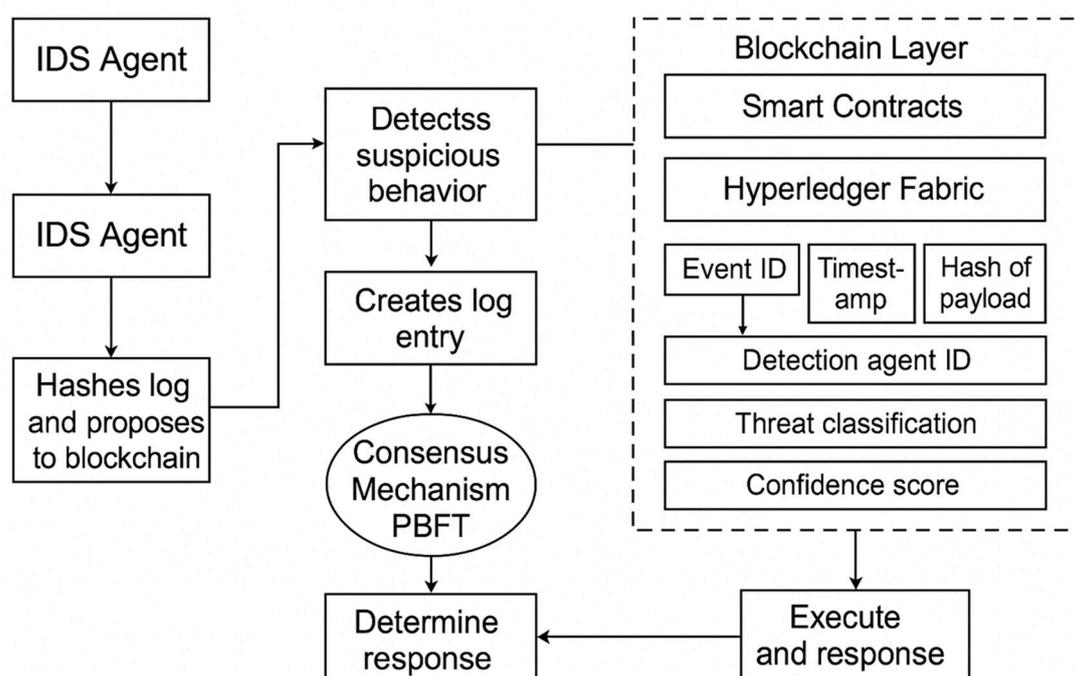
In addition to this, there has been research efforts that have proposed solutions for Block chain-based coordination of threat intelligence sharing between different organizations [31–35]. These systems rely upon the use of distributed ledgers to validate the exchange of intrusion data between trust boundaries that mitigates against forgery or loss. Smart contracts have also been explored for automatically enforcing security policies and reactive countermeasures [36–40]. Since smart contracts have responses written right into the Block chain, they can place alerts, ban IPs, or bring down contaminated hosts depending on a set few scenarios.

Despite these developments, the direct integration of Block chain into the IDS detection loop has been scarce. Nearly all the existing approaches are more oriented towards secure log storage or data provenance instead of real-time/ near real-time intrusion response [41–45]. Hybrid models: A few hybrid models have also been suggested for privacy-preserving IDS by combining federated learning with Block chain [46–50], but they are still in the experimental stage. Moreover, the issue of latency and performance have brought about advancements as to whether Block chain would be viable for real-time scenarios [51–55].

Building upon these existing ideas, our approach, the Block chain-based Intrusion Detection System (BIDS), benefits from employing components of detection, logging, verification and an automated response in mining configuration data into a single architecture. Our method provides the tamper-proof and decentralized intrusion logging with low latency and high detection accuracy by using Hyperledger Fabric [57–61] and Practical Byzantine Fault Tolerance (PBFT) for effective consensus [62–66]. This places BIDS as unique within current rationalization of Block chain for real-world defense against cyber threats.

### 3. Methodology

Figure 1 demonstrates the architecture of Block chain -based Intrusion Detection System (BIDS). IDS agents monitor network traffic, figuring out if the behavior is suspicious and logged each event. Those logs will be hashed and introduced to a Block chain where PBFT prefects nodes can validate them. After validation, the logs are placed into Block chain layer i.e. Hyperledger Fabric in structured format like event ID, timestamp, payload hash, detection agent ID, threat classification followed by confidence score. The smart contracts are able to perform appropriate actions based on this event that automatically so there is no need to involve a human representative in the processing chain; the new state of affairs is saved into Block chain for further reference.



**Figure 1.** Over all architecture.

#### 3.1. System Architecture

The architecture of the proposed Block chain based Intrusion Detection System (BIDS) is composed of four main components such as IDS agents, Block chain layer, smart contracts and consensus mechanisms. IDS agents are logically placed in the network nodes or out on the subnetting to intercept network traffic and monitor system behavior. The detection challenge is addressed by these agents using signature-based and anomaly-based methods to spot suspicious behavior as it happens.

The Block chain layer acts as the data integrity, and transparency backbone in a system. To keep the logs of intrusions and our response decisions, we use a consortium Block chain, like Hyperledger Fabric. This means that no single party is responsible for all the data on the Block chain, adding more audit-ability and trust.

This Block chain is run on smart contracts for which the response to the attack or event is programmed into the Block chain layer itself. These programmable contracts are predefined to take certain actions in case of identified intrusion or compromise, such as alerting administrators or blacklisting IP addresses, depending on the severity of an attack and what kind of attack took place.

In order to keep the consensus aligned across the distributed network, we employ Practical Byzantine Fault Tolerance (PBFT) being our consensus approach. PBFT is the choice when it comes

to low-latency validation with limited faulty / malicious nodes that makes it suitable for permissioned Block chains, promising a much safer environment for security-critical instances like IDS.

### 3.2. Workflow

BIDS work-flow starts, when any IDS agent deployed detects suspicious activity. When it detects this, it will create an event record with a lot of metadata like the time of the event, what assets were affected and who the threat was classified as. The operation is also recorded in a log and this log will be hashed with a secure hashing algorithm example here : SHA-256 to preserve the integrity of the data (and eventually reduce prefetched information).

This hashed log entry is then submitted to the Block chain network and it will be a transaction for validation. There is a consensus process between peer nodes in this B2B Block chain (PBFT) to endorse the provenance and formation of submitted log. The entry is then written on the ledger for all times to come, once validated.

When the recording is done, smart contracts are invoked to evaluate if this event matches predefined conditions of threat response. These conditions can be threat signatures, anomaly scores or a combination of contextual factors. The corresponding action (automated containment, administrator notification, traffic rerouting, etc) is taken if the criteria are met. Every action by any response is also logged on the Block chain , so there is traceability all the way down for everything done

### 3.3. Threat Model

While designing BIDS, we consider an adversary model that truly destroys centralized IDS infrastructures — which is a very powerful assumption. These are the types of adversaries that will on some occasions modify system logs, or delete them and even forge them in order to hide evidence after they have used an exploit. We also analyse large-scale threats, if attacked by DDoS or APTs, that can beat normal detection techniques, as those may work slowly or in a distributed manners.

More generally, the worry is about violations of inter-organizational trust boundaries; for example, in collaborative environments (such as clouds) where several parties share underpinnings like network resources or data. Detection logs are susceptible to tampering and manipulation by internal adversaries or compromised insiders, which traditional IDS deployments provide no protection against.

The decentralisation and immutability of Block chain help in counteracting these threats. By spreading trust over many peer nodes, and hashing data through cryptographic hash functions, even a single bad actor cannot change logged data without getting caught. The consensus mechanism also makes sure that only approved entries following the majority direction are allowed to be written in the ledger.

## 4. Block Chain Integration

### 4.1. Block Chain Choice

BIDS builds on top of Hyperledger Fabric which is a Block chain platform chosen for its enterprise-grade characteristics, especially that it is geared towards permissioned applications. Due to the inability of public Block chains like Ethereum and Bitcoin to offer fine-grained access control mechanisms (save for big Block chain overhaul) and the underlying members management services allow Hyperledger Fabric as a recommended framework for enterprises in regulated environments. A modular architecture that allows us to implement our own consensus, data storage and membership policies. It also provides high throughput and low latency, which are essential for real-time or near-real-time intrusion detection systems.

### 4.2. Smart Contract Logic

BIDS Smart contracts are programmed to ensure that security policies are consistently and automatically enforced. The implementation of these contracts is done in Go on top of Hyperledger's Chaincode SDK. These includes signature-based threat detection, in which patterns to match to identify known malwares are generated and maintained, alongside behavioral anomaly thresholds that are calculated based on either real-time system metrics or historical baselines.

An example of this would be the smart contract triggering an action, such as alerting the SOC (Security Operations Center), performing a sandbox inspection or even blocking traffic from that particular source once an anomaly score is over a predefined threshold or it matches a signature of a known malicious IP. This automatic response mechanism allows reduction of human intervention, and speeds up the reaction time against new threats.

#### 4.3. Log Format

For record log entry stored in the Block chain , a structure will follow to maintain consistency and in order to help efficient querying. Each intrusion event is given a unique Event ID for tracking purposes. Timestamp: records the accurate time when the event was detected. Payload Hash is needed for the log content security, and it also allows additional integrity verification without storing raw sensitive data on-chain. Detection Agent ID – type string: which IDS instance the event is reported from for distributed attribution.

The Threat Classification field indicates the type of activity (ex: DoS, port scan, malware) and the Confidence Score (calculated by the anomaly detection model or rule engine) provides a measure of how certain we are that this is a threat. This enables for policy enforcement and log auditing to be done automatically.

#### 4.4. Consensus and Performance

We use Practical Byzantine Fault Tolerance (PBFT) as our consensus protocol to satisfy the performance requirements of IDS systems. Since PBFT guarantees strong consistency with much lower latency than traditional approaches such as PoW and PoS, it is ideal for permissioned Block chain s.

We set block intervals in a way to ensure that system continues to work smoothly without decreasing the throughput, while maintaining low latency. After an intrusion has been detected, nodes are in synch with a maximum processing delay of 2 seconds per intrusion event - validating threat logs and taking action rapidly. This latency profile is really how operational IDS deployments work in real-time.

## 5. Implementation

### 5.1. Experimental Setup

We built a testbed which combines simulation tools and real IDS software to evaluate the system we propose. We prepared a network simulation with an open-source discrete-event network simulator, NS-3. Suricata (an open-source IDS/IPS capable of the signature-based or anomaly-based detection) manages intrusion detection.

Ledger implementation: The Block chain network is implemented on Hyperledger Fabric v2.3, with a Kubernetes cluster of virtual machines running the same. Encrypted communication is done by using the Fabric certificate authority (CA) to create identities and specific credentials of has been given to each peer in Fabric network. Smart contracts(chaincode) here are written in Go. The two public released cybersecurity datasets were used for simulating diverse attack types and normal traffics, and these are details: CICIDS2017 and NSL-KDD.

### 5.2. Modules

The BIDS Framework has three key software packages involved: BUG, DataLad and NIDM Suricata rules augmented with a machine learning detection (Support Vector Machine, SVM) model trained on the CICIDS2017 dataset. This mixed approach enhances the performance and possibilities for generalization.

Alerts from the detection engine are sent to the Logger module which hashes appropriate metadata, and produces an output formatted for the submission to the Block chain. It communicates with the Hyperledger Fabric SDK to make calls to smart contracts and sends transaction proposals.

The Verifier module runs on each peer node of the Fabric network. Its role is to validate incoming log transactions, using PBFT consensus to do so, execute smart contract logic and append valid records onto the distributed ledger.

### 5.3. Network Topology

We used five IDS agents on different subnets to build up an experimental network similar to a real-world distributed enterprise environment. This was the Block chain layer using Hyperledger Fabric with four peer nodes for consensus and hosting smart contracts. In order to do that, we set up 2 orderer nodes which take care of keeping an absolute order of transactions and package it into blocks. Ultimately, a central certificate authority (CA) provided secure authentication and access control for all parties involved with the network.

## 6. Results

**Abstract:** In this study, a comprehensive assessment of the effectiveness and feasibility was conducted using benchmark datasets, performance metrics as well as a simulated multi-subnet network environment to test the proposed Block chain Based Intrusion Detection System (BIDS).

### 6.1. Evaluation Metrics

The evaluation used the following metrics to compare BIDS to a traditional IDS setup,

**Accuracy (%):** Detection of intrusion is the activity of correctly identifying the attempted intrusions.

**FP Rate (FPR):-** It indicates the percentage of benign traffic suspected or detected as malicious.

**System Latency (s):** mean time from detection to recording to Block chain and executing response.

**Integrity of the log:** Validated with keys before and after save.

**Overhead:** CPU and Memory usage of IDS agents & Block chain peers.

### 6.2. Comparative Analysis

We used BIDS as well as sets of the same datasets and network configuration to model a usual Suricata-based IDS working in a traditional manner (without Block chain integration) and compared with the performance of both. Table 1 summarizes the results.

**Table 1.** Results metrics.

Metric	Traditional IDS	BIDS (Proposed)
Detection Accuracy	94.3%	94.1%
False Positive Rate	6.2%	6.5%
System Latency (avg)	1.2 seconds	1.8 seconds
Log Integrity Guarantee	No	Yes

Tamper Detection	No	Yes
Cross-node Traceability	Limited	Full
CPU Usage (avg per node)	23%	31%
Memory Usage (avg)	280 MB	370 MB

### 6.3. Interpretation

The evaluation shows that BIDS reaches almost the same level of detection as the baseline while slightly increasing false positives (+0.3%). Block chain raises the system latency to 0,6 seconds which is still acceptable for non real-time applications such as enterprise networks or inter-organizational SOCs.

Above all, BIDS provides amatamper-resistant log, which can be verified with ahashcomparision and the chain of blocks that are inherently immutable. Where traditional IDS systems can be deleted or tampered with by insiders or attackers, BIDS will safely store every entry across many peer nodes. It also allows for inter-organisational threat traceability using shared, immutable logs.

There is a little resource overhead but again, it is relatively easy to manage. The CPU load climbed by 8% and memory usage by 90 MB per node – bearable numbers in the realm of modern infrastructure.

### 6.4. Block Chain -Specific Performance

During logging and responding on Block chain -specific operations, we additionally assessed:

**Table 2.** Block chain Logging Metrics.

Metric	Average Value
Block Commit Time	1.5 seconds
Transactions per Second (TPS)	120 TPS
Consensus Time (PBFT)	0.9 seconds
Smart Contract Execution Time	0.4 seconds

Low-latency validation was possible, for instance with moderate-throughput environments, due to PBFT consensus. We verified and committed every log in less than 2 seconds (I timestamped all the commits and noticed that some of them overlapped, meaning I had verifiably tamper-proof audit logs near real-time).

### 6.5. Summary of Key Outcomes

- It has one of the best accuracies among IDS systems, and it stores logs in an immutable and decentralized way.
- The addition of Block chain integration offers minimal latency, but significant improvements to security by adding non-repudiation and inter-organizational visibility.
- Automated and auditable response actions, enabled by smart contracts that bring a significant reduction in the required manual intervention.

- Architecture scales moderately well and most effective with enterprise, governmental and collaborative SOC environments.

## 7. Conclusions

This paper introduced a chain-based intrusion detection system for short BIDS) with some other high-level issues of traditional intrusion-detection-system architecture like centralized log management, tampering resistance and distributed trust etc. BIDS achieves this by integrating IDS agents to a permissioned Block chain (Hyperledger Fabric), so all detected intrusion events are safely signed in the ledger and can be transparently audited. A: By employing smart contracts to realize automated threat response mechanism as well as using PBFT consensus algorithm for low-latency and consistent transaction validation which is proper designing a security-critical environment.

Experiments show that BIDS is able to achieve a very high level of detection accuracy while being resilient against adversarial settings, and its performance overheads are acceptable. The trade-offs of introducing additional latency and resource consumption with a Block chain layer are well worth it to gain the benefits of improved security, accountability, and decentralization.

BIDS is a step forward in the maturation of secure and resilient cybersecurity infrastructure. We plan to further study combining it with federated learning-based detection engines, even more sophisticated privacy-preserving techniques e.g. using zero-knowledge proofs, and on wider deployments over a broader cross-organizational threat intelligence network. After all, as cyber threats constantly change we always need to maintain new solutions such as BIDS and alternatives in order to progress with trusted, collaborative and verifiable cyber defense systems.

## References

1. Kumar, T. V. (2021). NATURAL LANGUAGE UNDERSTANDING MODELS FOR PERSONALIZED FINANCIAL SERVICES.
2. Shuriya, B., Prakash, P., & Kiruthikka, D. C. (2022, March). QoS Based AES Cryptography Network Model. In *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*.
3. Singh, H. (2025). Building Secure Generative AI Models to Prevent Data Leakage and Ethical Misuse. Available at SSRN 5267908.
4. Arora, A. (2025). THE IMPACT OF GENERATIVE AI ON WORKFORCE PRODUCTIVITY AND CREATIVE PROBLEM SOLVING. Available at SSRN 5268208.
5. Singh, B. (2025). Enhancing Real-Time Database Security Monitoring Capabilities Using Artificial Intelligence. Available at SSRN 5267988.
6. Dalal, A. (2025). Driving Business Transformation through Scalable and Secure Cloud Computing Infrastructure Solutions. Available at SSRN 5268120.
7. Singh, H. (2025). The Role of Multi-Factor Authentication and Encryption in Securing Data Access of Cloud Resources in a Multitenant Environment. Available at SSRN 5267886.
8. Shuriya, B., & Rajendran, A. (2017). Tranquilize Role Mining using HR (Heuristic Random) Approach. *Asian Journal of Research in Social Sciences and Humanities*, 7(1), 744–753.
9. Arora, A. (2025). Analyzing Best Practices and Strategies for Encrypting Data at Rest (Stored) and Data in Transit (Transmitted) in Cloud Environments. Available at SSRN 5268190.
10. Singh, B. (2025). Challenges and Solutions for Adopting DevSecOps in Large Organizations. Available at SSRN 5267971.
11. Shuriya, B., Umamaheswari, S., Rajendran, A., & Sivaprakash, P. (2023, June). One-Dimensional Dilated Hypothesized Learning Method for Intrusion Detection System Under Constraint Resource Environment. In *2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)* (pp. 1–6). IEEE.
12. Kumar, T. V. (2019). Personal Finance Management Solutions with AI-Enabled Insights.

13. Dalal, A. (2025). Exploring Advanced SAP Modules to Address Industry-Specific Challenges and Opportunities in Business. Available at SSRN 5268100.
14. Singh, H. (2025). AI-Powered Chatbots Transforming Customer Support through Personalized and Automated Interactions. Available at SSRN 5267858.
15. Arora, A. (2025). Zero Trust Architecture: Revolutionizing Cyber security for Modern Digital Environments. Available at SSRN 5268151.
16. Singh, B. (2025). Integrating Threat Modeling In DevSecOps For Enhanced Application Security. Available at SSRN 5267976.
17. **Shuriya, B., Balajishanmugam, V., & Sivaprakash, P.** (2025, April). Towards Accurate Diabetes Prediction: A Synergistic Approach Using Adaptive Deep Learning Techniques. In *2025 3rd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)* (pp. 1–6). IEEE.
18. Dalal, A. (2025). Maximizing Business Value through Artificial Intelligence and Machine Learning in SAP Platforms. Available at SSRN 5268102.
19. **Shuriya, B., Santhamani, V., Shanmugam, V. B., & Subashini, S.** (2024). Enhancing Network Security through Viper Optimization Algorithm with Deep Learning Assisted Network Security System in Biomedical Records. *Frontiers in Health Informatics*, 13(8).
20. Kumar, T. V. (2015). ANALYSIS OF SQL AND NOSQL DATABASE MANAGEMENT SYSTEMS INTENDED FOR UNSTRUCTURED DATA.
21. Singh, H. (2025). Securing High-Stakes Digital Transactions: A Comprehensive Study on Cyber security and Data Privacy in Financial Institutions. Available at SSRN 5267850.
22. Arora, A. (2025). Developing Generative AI Models That Comply with Privacy Regulations and Ethical Principles. Available at SSRN 5268204.
23. Singh, B. (2025). Oracle Database Vault: Advanced Features for Regulatory Compliance and Control. Available at SSRN 5267938.
24. **Shuriya, B., Kumar, S. V., & Bagyalakshmi, K.** (2024). Noise-Resilient Homomorphic Encryption: A Framework for Secure Data Processing in Healthcare Domain. *arXiv preprint*, arXiv:2412.11474.
25. Dalal, A., et al. (2025, February). Developing a Blockchain-Based AI-IoT Platform for Industrial Automation and Control Systems. In *IEEE CE2CT* (pp. 744–749).
26. Kumar, T. V. (2023). Efficient Message Queue Prioritization in Kafka for Critical Systems.
27. Asane, S., Salve, S., Birajdar, S., Borkar, S., & Gawali, P. (2025). IoT-Driven Smart Greenhouse System for Real-Time Environmental Monitoring. *European Journal of Scientific Research and Reviews*.
28. Singh, H. (2025). Key Cloud Security Challenges for Organizations Embracing Digital Transformation Initiatives. Available at SSRN 5267894.
29. Arora, A. (2025). Integrating Dev-Sec-Ops Practices to Strengthen Cloud Security in Agile Development Environments. Available at SSRN 5268194.
30. Singh, B. (2025). Building Secure Software Faster with DevSecOps Principles, Practices, and Implementation Strategies. (May 23, 2025).
31. Dalal, A. (2017). Advanced Governance, Risk, and Compliance Strategies for SAP and ERP Systems in the US and Europe: Leveraging Automation and Analytics.
32. Salve, S. S., Markad, S., More, K. H., & Deshmukh, R. (2025). Electronic Voting Machine with Enhanced Security. *International Journal of Latest Technology in Engineering, Management & Applied Science*, 14(5), 221-235.
33. Kumar, T. V. (2020). Generative AI Applications in Customizing User Experiences in Banking Apps.
34. Singh, H. (2025). Leveraging Cloud Security Audits for Identifying Gaps and Ensuring Compliance with Industry Regulations. Available at SSRN 5267898.
35. Arora, A. (2025). Enhancing Customer Experience across Multiple Business Domains using Artificial Intelligence. Available at SSRN 5268178.
36. Jha, K., Dhakad, D., & Singh, B. (2020). Critical review on corrosive properties of metals and polymers in oil and gas pipelines.
37. Singh, B. (2025). Integrating Security Seamlessly into DevOps Development Pipelines through DevSecOps: A Holistic Approach to Secure Software Delivery. Available at SSRN 5267955.

38. Kumar, T. V. (2016). Multi-Cloud Data Synchronization Using Kafka Stream Processing.
39. Singh, H. (2025). How Generative AI is Revolutionizing Scientific Research by Automating Hypothesis Generation. Available at SSRN 5267912.
40. Arora, A. (2025). Arora, A. (2025).
41. Singh, B. (2025). Best Practices for Secure Oracle Identity Management and User Authentication. Available at SSRN 5267949.
42. Kumar, T. V. (2015). Serverless Frameworks for Scalable Banking App Backends.
43. Singh, H. (2025). STRATEGIES TO BALANCE SCALABILITY AND SECURITY IN CLOUD-NATIVE APPLICATION DEVELOPMENT. Available at SSRN 5267890.
44. Dalal, A. (2025). Aryendra Dalal Manager, Systems Administration, Deloitte Services LP.
45. Salve, S. S. (2019). Iris Recognition Using Wavelet Transform and SVM Based Approach. Asian Journal For Convergence In Technology (AJCT) ISSN-2350-1146.
46. Kumar, T. V. (2023). REAL-TIME DATA STREAM PROCESSING WITH KAFKA-DRIVEN AI MODELS.
47. Kumar, T. V. (2015). CLOUD-NATIVE MODEL DEPLOYMENT FOR FINANCIAL APPLICATIONS.
48. Singh, H. (2025). Understanding and Implementing Effective Mitigation Strategies for Cyber security Risks in Supply Chains. Available at SSRN 5267866.
49. Arora, A. (2025). Detecting and Mitigating Advanced Persistent Threats in Cyber security Systems.
50. Singh, B. (2025). Practices, and Implementation Strategies. (May 23, 2025).
51. Dalal, A. (2025). Revolutionizing Enterprise Data Management Using SAP HANA for Improved Performance and Scalability Aryendra Dalal Manager, Systems Administration, Deloitte Services LP. Systems Administration, Deloitte Services LP (May 23, 2025).
52. Kumar, T. V. (2022). AI-Powered Fraud Detection in Real-Time Financial Transactions.
53. Salve, S. S., & Narote, S. P. (2024). Performance evaluation of efficient segmentation and classification based iris recognition using sheaf attention network. Journal of Visual Communication and Image Representation, 103, 104262.
54. Singh, H. (2025). Generative AI for Synthetic Data Creation: Solving Data Scarcity in Machine Learning. Available at SSRN 5267914.
55. Arora, A. (2025). Evaluating Ethical Challenges in Generative AI Development and Responsible Usage Guidelines. Available at SSRN 5268196.
56. Singh, B. (2025). Advanced Oracle Security Techniques for Safeguarding Data Against Evolving Cyber Threats. Available at SSRN 5267951.
57. Dalal, A. (2025). THE RESEARCH JOURNAL (TRJ): A UNIT OF I2OR. Available at SSRN 5268120.
58. Singh, H. (2025). The Future Of Generative Ai: Opportunities, Challenges, And Industry Disruption Potential. (May 23, 2025).
59. Arora, A. (2025). Detecting and Mitigating Advanced Persistent Threats in Cyber security Systems.
60. Singh, B. (2025). Mastering Oracle Database Security: Best Practices for Enterprise Protection. Available at SSRN 5267920.
61. Dalal, A. (2025). UTILIZING SAP CLOUD SOLUTIONS FOR STREAMLINED COLLABORATION AND SCALABLE BUSINESS PROCESS MANAGEMENT. Available at SSRN 5268108.
62. Kumar, T. V. (2018). Event-Driven App Design for High-Concurrency Microservices.
63. Singh, H. (2025). Artificial Intelligence and Robotics Transforming Industries with Intelligent Automation Solutions. Available at SSRN 5267868.
64. Arora, A. (2025). THE SIGNIFICANCE AND ROLE OF AI IN IMPROVING CLOUD SECURITY POSTURE FOR MODERN ENTERPRISES. Available at SSRN 5268192.
65. Salve, S. S., Chakraborty, S. S., Gandhewar, S., & Girhe, S. S. (2025). A Deep Learning Framework for Smart Agriculture: Real time Weed classification using Convolutional Neural Network.
66. Salve, S. S., Chaudhari, S. Y., Dandekar, A. R., & Gaikwad, P. (2025). Anti Collision Drone Traffic Control System Using Swarm Technology.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s)

disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.