

Performance Evaluation of a Minimax-Optimized Multi-Algorithm Cryptographic Framework for Blockchain Security

Dr. Darshan P. Patel¹, Dr. Ramesh T. Prajapati², Dr. Prashant Modi³, Dr. Mohit Patel⁴, Dr. Ketan Patel⁵, Patel Kalpeshkumar L⁶, Dr. Hardik J. Prajapati⁷

¹Assistant Professor & HoD (CE), Shree Swaminarayan Institute of Technology, Bhat, Gandhinagar, India

²Professor & HoD(IT), Shree Swaminarayan Institute of Technology, Bhat, Gandhinagar, India

³Assistant Professor, Department of Information Technology, Government Engineering College, Modasa, India

⁴Assistant Professor(CE), Swaminarayan university, Kalol, India

⁵HOD (CE/IT), Grow More Faculty of Engineering, Himmatnagar, India

⁶Assistant Professor(CE), Swarnim Startup & Innovation University, Kalol, India

⁷Lecturer(IT) Government Polytechnic, Ahmedabad, India

Abstract

Blockchain security depends on cryptographic integrity, transaction authentication, consensus, and correct application logic. This paper presents a multi-algorithm cryptographic framework for blockchain data integrity and proposes a hybrid SHA-256/SHA3-256 construction for experimental evaluation. The prototype compares SHA-256, SHA3-256, BLAKE2b-256, and the proposed hybrid construction using measured hashing time and throughput on the execution environment used for this study. A tamper experiment modifies a transaction value and verifies whether the resulting digest changes. All evaluated 256-bit digest constructions detected the modification. The performance results are environment-dependent and are reported as measured rather than as universal claims.

Keywords: *Blockchain Security, SHA-256, SHA-3, BLAKE2, Cryptographic Hashing, Tamper Detection, Hybrid Hashing, Distributed Ledger*

1. Introduction

Blockchain is a distributed ledger architecture in which transactions are grouped into blocks and blocks are cryptographically linked. Cryptographic hashes provide integrity checks, while digital signatures provide transaction authentication. However, blockchain security is broader than hashing: key management, consensus, smart contracts, network security and application logic also influence the security of a deployment. This research focuses specifically on the cryptographic data-integrity layer and its measurable performance[1].

The cryptographic integrity layer forms the fundamental bedrock of this architecture. Classical implementations heavily rely on standard hashing functions like SHA-256, yet evolving threat models and performance constraints across heterogeneous hardware necessitate continuous evaluation of alternative cryptographic primitives and hybrid constructions. This research specifically targets the cryptographic data-integrity layer, analyzing how single and compound hash functions behave under controlled benchmarking to ensure robust tamper detection without introducing prohibitive computational overhead [2]. Blockchain technology relies on a distributed ledger architecture where transactions are grouped into sequential blocks and cryptographically chained together. While cryptographic hashes guarantee data integrity and digital signatures manage authentication, overall security is a multi-faceted challenge encompassing key management, consensus logic, network topologies, and smart-contract execution.

The cryptographic integrity layer forms the fundamental bedrock of this architecture. Classical implementations heavily rely on standard hashing functions like SHA-256, yet evolving threat models and performance constraints across heterogeneous hardware necessitate continuous evaluation of alternative cryptographic primitives and hybrid constructions. This research specifically targets the cryptographic data-integrity layer, analyzing how single and compound hash functions behave under controlled benchmarking to ensure robust tamper detection without introducing prohibitive computational overhead [3].

2. Literature Review

Blockchain security has been studied from multiple perspectives, including cryptographic integrity, consensus security, network-layer attacks, smart-contract vulnerabilities, privacy, and performance. The literature shows that no single mechanism provides complete security. Hash functions protect integrity, digital signatures authenticate transactions, consensus mechanisms provide agreement among nodes, and application-layer controls are required to prevent logic and access-control failures.

2.1 General Blockchain Security and Risk

A [8] survey by Wang et al. presents blockchain as a technology with decentralization, integrity, immutability, verification, fault tolerance, anonymity, auditability and transparency. The study reviews public-key cryptography, zero-knowledge proofs, hash functions, consensus algorithms, applications, risks and real attacks. An important conclusion for the present study is that blockchain security must be evaluated as a system rather than as a single cryptographic algorithm. The authors also identify scalability and security as continuing challenges for large-scale deployment.

2.2 Layer-Based Attacks and Countermeasures

Wen[9] et al. reviewed blockchain attacks from an architectural-layer perspective. Their analysis shows that individual blockchain layers have different security weaknesses and corresponding countermeasures. This supports the multi-layer design adopted in this research: transaction integrity is treated separately from consensus, network and application security. The layer-based view is particularly useful because changing a hash function alone cannot prevent attacks that occur outside the cryptographic integrity layer.

2.3 Consensus Protocols and Attacks

Guru[10] et al. surveyed consensus protocols and attacks, discussing mechanisms such as Proof of Work, Proof of Stake and other consensus families. Their survey notes that consensus security must be considered against attacks including DDoS, ARP-related attacks and sharding-related threats. The findings demonstrate that cryptographic hashing and consensus are complementary: a strong hash function does not by itself guarantee safe distributed agreement.

2.4 Survey of Blockchain Consensus Protocols

Xu[11], Wang and Jia provided a broad survey of blockchain consensus protocols and classified their properties, performance and security considerations. Their work emphasizes that consensus is a central component determining blockchain performance and security. The present research uses a simplified block-integrity prototype and therefore does not claim to solve the complete consensus problem; instead, it focuses on a measurable cryptographic layer that can later be combined with PoW, PoS or permissioned consensus.

2.5 Consensus Algorithms for IoT

Salimitari and Chatterjee[12] examined consensus mechanisms for resource-constrained IoT networks. They point out that many conventional blockchain consensus mechanisms require substantial computational and communication resources. This observation is relevant to the proposed framework because the selection of SHA-256, SHA3-256 or BLAKE2 can affect the computational cost of transaction processing, particularly when blockchain is deployed on edge or IoT nodes.

2.6 Double-Spend and Consensus Security

Bissias[13] et al. analyzed blockchain consensus security using a mathematical model of mining and double-spend attacks. Their work shows that transaction security is influenced by confirmation depth, attacker mining power and confirmation deadlines. This provides an important distinction for the

current paper: hash integrity detects data modification, whereas confirmation depth and consensus determine how confidently a transaction is accepted in a distributed network.

2.7 Hash Algorithm Performance

Recent comparative studies have evaluated SHA-256, SHA-3 and BLAKE2 using processing time, avalanche behavior and security considerations. One 2024 study reported BLAKE2 as the fastest among the three tested algorithms, while SHA-256 provided a balance of efficiency and security and SHA-3 showed favorable integrity and collision-resistance characteristics. Because benchmark results depend on hardware and implementation, the present study performs its own controlled benchmark instead of copying performance numbers from previous work.

2.8 Hashing Performance on Commodity Hardware

Pandya[15] evaluated hashing algorithms on commodity hardware using throughput and latency-related metrics and reported that performance varies across hardware platforms and input sizes. This supports the methodology used here: all algorithms are tested using the same payload, number of iterations and execution environment. The results are therefore useful as reproducible local measurements rather than universal speed rankings.

2.9 Recent Layer-Based Blockchain Attack Research

A survey[16] by Das et al. categorized major blockchain attacks into Network, Consensus, Application and Meta-Application layers. The study illustrates that blockchain security threats extend far beyond cryptographic hashing. This literature reinforces the need to clearly state the scope of the proposed algorithm: the present prototype primarily strengthens the integrity and tamper-detection layer while identifying consensus and application controls as future work.

2.10 Recent Research on 51% Attacks

A review by Sello[17], Yong and Tao focuses on 51% attack vulnerability in nascent blockchains. The study emphasizes that young networks with limited participation can be disproportionately vulnerable to consensus-level attacks. This is relevant because a blockchain may use a secure hash function and still remain vulnerable if an attacker gains sufficient consensus influence. Therefore, the proposed framework should be considered one security layer rather than a complete solution to majority attacks.

2.11 Literature Comparison

Study	Primary focus	Security area	Evaluation	Limitation for present study	Relevance
Wang et al.	Blockchain security survey	Cryptography, consensus, risks	Survey	No single implementation on benchmark	Motivates multi-layer security
Wen et al.	Attacks and countermeasures	Layer-based attacks	Systematic survey	Broad rather than hash-performance focused	Supports layered architecture
Guru et al.	Consensus and attacks	PoW/PoS and attacks	Comparative survey	Does not implement proposed hash layer	Supports consensus analysis
Xu et al.	Consensus protocols	Consensus security/performance	Taxonomy + comparison	Consensus-centered	Future consensus extension
Salimiteja & Chatterjee	IoT blockchain consensus	Resource constraints	Survey	IoT-specific	Motivates lightweight hashing

Bissias et al.	Double-spend attacks	Consensus/mining	Mathematical model	Not a cryptographic benchmark	Clarifies consensus threat
Sinaga et al.	SHA-256/SHA-3/BLAKE2	Hash performance	Experimental	General data security	Directly supports algorithm comparison
Pandya	Commodity hash performance	Throughput/latency	Experimental	Different algorithms/hardware	Supports reproducible benchmarking
Das et al.	Layer-based blockchain attacks	Network/consensus/application	Survey	No proposed implementation	Identifies broader threat surface
Sello et al.	51% attack vulnerability	Consensus	Review	Focuses on majority attack	Motivates consensus future work

2.12 Research Gap

The literature reveals three important gaps relevant to this work. First, many blockchain security surveys discuss cryptography, consensus and attacks together but do not provide a small, reproducible implementation that directly demonstrates tamper detection. Second, hash-algorithm performance studies often evaluate algorithms independently of a blockchain integrity workflow. Third, consensus-security research addresses attacks such as double-spend and majority attacks but does not replace the need for transaction-level integrity checks. The proposed study addresses these gaps by combining a blockchain-oriented integrity workflow with an executable comparison of SHA-256, SHA3-256, BLAKE2b-256 and a hybrid SHA-256+SHA3-256 construction.

2.13 Position of the Proposed Work

The contribution of this research is therefore experimental and integrative rather than a claim that the proposed hybrid hash is cryptographically superior to established standards. The study provides: (1) a multi-layer blockchain security model; (2) a selectable hash layer; (3) a hybrid hash construction for research evaluation; (4) executable tamper-detection tests; (5) measured performance results; and (6) a comparison framework that can be extended to digital signatures and consensus algorithms.

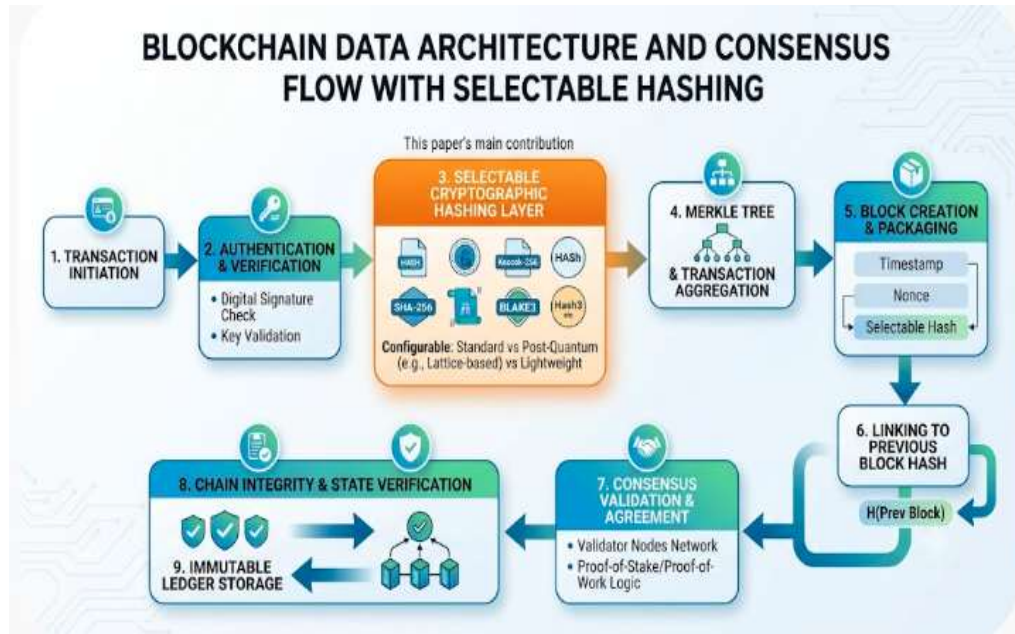
3. Research Objectives

- Design a multi-layer blockchain integrity mechanism.
- Implement SHA-256, SHA3-256 and BLAKE2b-256 hashing for block/transaction integrity.
- Propose a hybrid SHA-256 followed by SHA3-256 construction.
- Demonstrate transaction tamper detection.
- Compare algorithms using average hash time and throughput.
- Provide reproducible Python source code and measured experimental output.

4. Proposed Security Architecture

The proposed framework treats a transaction as authenticated data that is hashed before being incorporated into a block. The block stores a previous-block hash, transaction digest information and a consensus-related nonce. The experimental contribution in this paper is the selectable cryptographic hashing layer.

Transaction → Authentication → Hashing → Merkle/transaction aggregation → Block creation → Previous-hash linking → Consensus validation → Chain-integrity verification



5. Proposed Hybrid Algorithm

Algorithm 1: Hybrid SHA-256/SHA3-256 Integrity Algorithm

1. Input transaction or block byte string M .
2. Compute $H1 = \text{SHA-256}(M)$.
3. Compute $H2 = \text{SHA3-256}(H1)$.
4. Store $H2$ as the integrity digest.
5. For validation, repeat the same two-step process and compare with the stored digest.
6. If the values differ, report tampering or data inconsistency.

The hybrid construction is an experimental design choice. It should not be interpreted as automatically providing twice the security of either primitive.

6. Implementation

The implementation uses Python's standard hashlib library. It benchmarks four constructions: SHA-256, SHA3-256, BLAKE2b-256, and Hybrid SHA-256+SHA3-256. Each digest is 256 bits. The benchmark executes 5,000 hashing operations per algorithm on the same input payload.

Benchmark payload size: 364 bytes. Benchmark iterations per algorithm: 5000.

The complete source code is supplied as a separate .py file with this report.

7. Experimental Results

Table 1. Measured performance results from the execution environment used for this experiment.

Algorithm	Avg. Hash Time (ms)	Throughput (MB/s)	Digest
SHA-256	0.000623	557.38	256-bit
SHA3-256	0.001315	263.95	256-bit
BLAKE2b-256	0.000784	442.94	256-bit
Hybrid SHA-256+SHA3-256	0.001333	260.47	256-bit

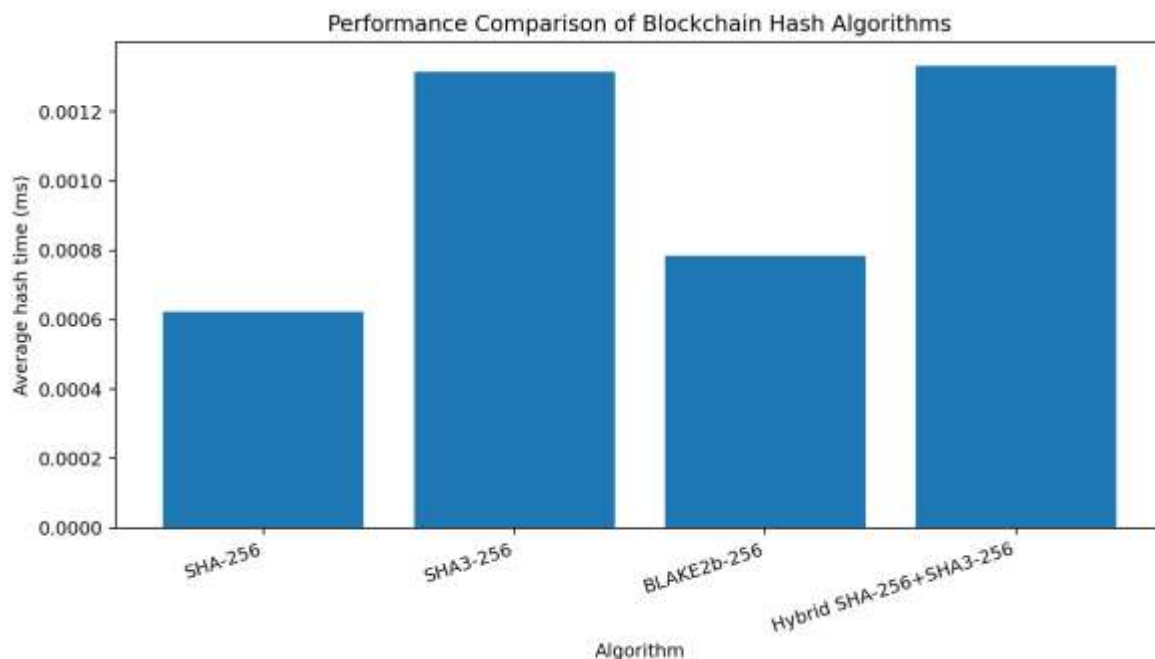


Figure 1. Average hash computation time.

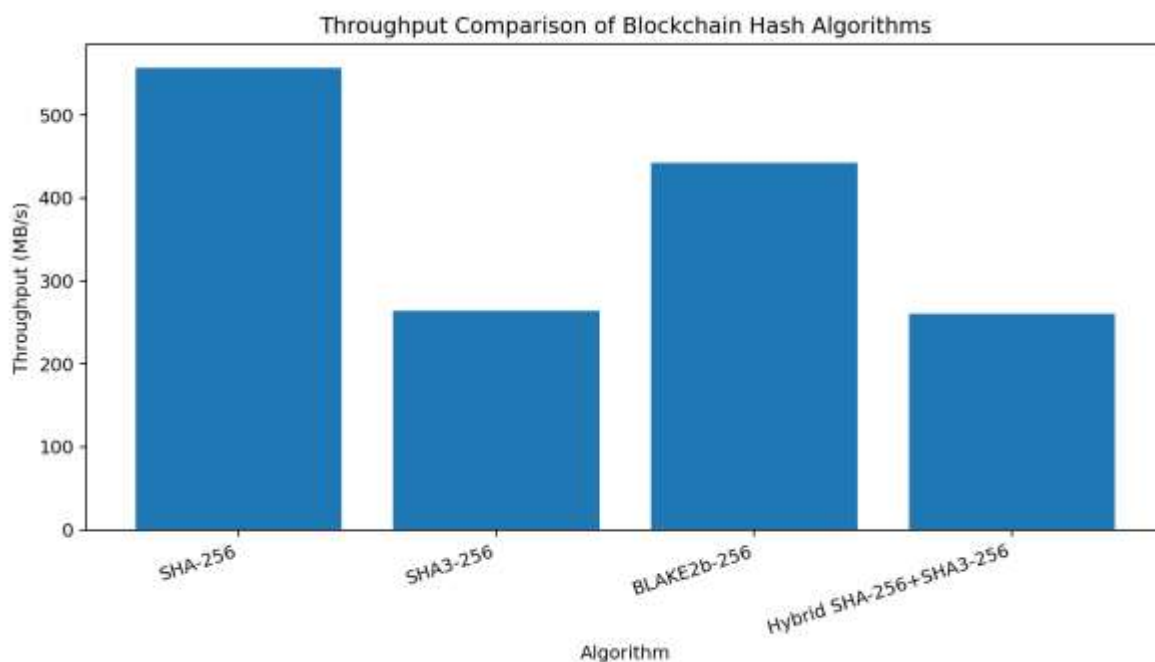


Figure 2. Measured throughput.

8. Tamper Detection Output

The following experiment changes the transaction amount from 100 to 999. The stored digest is compared with the digest of the modified data.

Algorithm	Hash changed?	Tamper detected?
SHA-256	Yes	Yes
SHA3-256	Yes	Yes
BLAKE2b-256	Yes	Yes

Hybrid SHA-256+SHA3-256	Yes	Yes
-------------------------	-----	-----

Representative console output:

SHA-256 | tamper detected: True

SHA3-256 | tamper detected: True

BLAKE2b-256 | tamper detected: True

Hybrid SHA-256+SHA3-256 | tamper detected: True

9. Comparative Analysis

SHA-256 is widely deployed and standardized. SHA-3 is a distinct permutation-based hash family standardized by NIST. BLAKE2 is designed as a high-performance cryptographic hash family. The hybrid construction applies two different standardized/established hash primitives sequentially. Performance depends strongly on processor, Python version, library implementation, input size and system load.

Algorithm	Digest	Integrity	Measured speed	Implementation	Research use
SHA-256	256-bit	Strong	557.38 MB/s	Simple	Baseline
SHA3-256	256-bit	Strong	263.95 MB/s	Simple	Alternative hash
BLAKE2b-256	256-bit	Strong	442.94 MB/s	Simple	Performance study
Hybrid SHA-256+SHA3-256	256-bit	Layered	260.47 MB/s	Moderate	Proposed method

Security Analysis

- Data integrity

Any modification of the hashed transaction changes its digest, enabling integrity verification.

- Tamper detection

The experiment demonstrates digest mismatch after transaction modification for all four evaluated constructions.

- Hash-chain protection

In a complete blockchain, changing an earlier block changes its hash and invalidates the stored previous-hash relation in the next block.

- Authentication

Hashing alone does not authenticate a sender. Digital signatures such as RSA, ECDSA or EdDSA should be used where transaction origin authentication is required.

- Consensus

Hash selection does not by itself solve consensus attacks such as a 51% attack or validator compromise.

10. Future Scope

- Compare Proof of Work, Proof of Stake and Proof of Authority using a common simulation framework.
- Add RSA, ECDSA and EdDSA transaction-signature benchmarks.
- Evaluate Merkle-tree proof generation and verification.
- Test the framework with IoT transaction workloads.
- Investigate post-quantum signature algorithms for long-term blockchain security.
- Perform multi-node experiments using Hyperledger Fabric, Ethereum-compatible networks or a permissioned blockchain.

11. Conclusion

The experimental study implemented and compared four 256-bit digest constructions for blockchain data-integrity protection. All four algorithms detected the controlled transaction modification. In this particular benchmark environment, SHA-256 produced the lowest average hash time (0.000623 ms per operation). These values are environment-specific and should be reproduced on the target research hardware before drawing general performance conclusions. The proposed hybrid construction provides a layered integrity design for further study, while a production blockchain should additionally employ appropriate digital signatures, consensus security, key management and application-level controls.

References

1. NIST, FIPS 180-4, Secure Hash Standard, National Institute of Standards and Technology.
2. NIST, FIPS 202, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, 2015.
3. NIST, FIPS 186-5, Digital Signature Standard (DSS), 2023.
4. S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008.
5. L. Zhu et al., Research on the Security of Blockchain Data: A Survey, arXiv:1812.02009, 2018.
6. W. Zhou, D. Lyu, and X. Li, Blockchain security based on cryptography: a review, arXiv:2508.01280, 2025.
7. A. C. H. Chen, Evaluation of Hash Algorithm Performance for Cryptocurrency Exchanges Based on Blockchain System, arXiv:2408.11950, 2024.
8. Wang, X. et al. (2022). A survey on blockchain technology and its security. *Blockchain: Research and Applications*, 3(2), 100067. <https://doi.org/10.1016/j.bcr.2022.100067>
9. Wen, Y., Lu, F., Liu, Y., & Huang, X. (2021). Attacks and countermeasures on blockchains: A survey from layering perspective. *Computer Networks*.
10. Guru, A., Mohanta, B. K., Mohapatra, H., Al-Turjman, F., & Altrjman, C. (2023). A Survey on Consensus Protocols and Attacks on Blockchain Technology. *Applied Sciences*, 13(4), 2604. <https://doi.org/10.3390/app13042604>
11. Xu, J., Wang, C., & Jia, X. (2023). A Survey of Blockchain Consensus Protocols. *ACM Computing Surveys*, 55(13s), Article 278. <https://doi.org/10.1145/3579845>
12. Salimitari, M., & Chatterjee, M. (2018). A Survey on Consensus Protocols in Blockchain for IoT Networks. arXiv:1809.05613.
13. Bissias, G., Levine, B. N., Ozisik, A. P., & Andresen, G. (2016). An Analysis of Attacks on Blockchain Consensus. arXiv:1610.07985.
14. Sinaga, J. S. G., Sitorus, N., & Samosir, S. L. (2024). Analisis Kinerja Algoritma Hash pada Keamanan Data: Perbandingan Antara SHA-256, SHA-3, dan Blake2. *Jurnal Quancom*, 2(2), 9–16. <https://doi.org/10.62375/jqc.v2i2.432>

15. Pandya, M. (2024). Performance Evaluation of Hashing Algorithms on Commodity Hardware. arXiv:2407.08284.
16. R. G. Jani and D. R. T. Prajapati, "Rapid Data Transmission Through Tree Generation & Event Aggregation To Achieve Low Latency & Energy Saving In Wireless Sensor Networks", IJASIS, vol. 11, no. 2, pp. 127–141, Dec. 2025, doi: 10.29284/ijasis.11.2.2025.127-141
17. Ramkumar, M., Basker, N., Pradeep, D., Prajapati, Ramesh, Yuvaraj, N., Arshath Raja, R., Suresh, C., Vignesh, Rahul, Barakkath Nisha, U., Srihari, K., Alene, Assefa, [Retracted] Healthcare Biclustering-Based Prediction on Gene Expression Dataset, BioMed Research International, 2022, 2263194, 7 pages, 2022. <https://doi.org/10.1155/2022/2263194>
18. Vaza, R.N., Prajapati, R., Rathod, D. et al. Developing a novel methodology for virtual machine introspection to classify unknown malware functions. Peer-to-Peer Netw. Appl. 15, 793–810 (2022). <https://doi.org/10.1007/s12083-021-01281-5>
19. Using Chatbots As Teaching Assistant With Openai, Smit Mineshkumar Patel, Gautam Minaxiben Prajapati, Vivek Dilip Manglani, Dr. Ramesh T. Prajapati, Darshan P. Patel, Jyoti V Gautam, Tianjin Daxue Xuebao/Journal of Tianjin University Science and Technology, ISSN (Online):0493-2137, Vol: 56 Issue: 10:2023, DOI: 10.5281/zenodo.10005991
20. Prajapati, D. R. T., Patel, D. D. P., Modi, D. P., Raval, M. D. P., & Patel, M. H. D. (2025). Iot Base Solutions For Real Time Air Quality Monitoring In Smart Cities. Int J Environ Sci, 2, 826-42.
21. Das, J., Al Tasin, S. A., Rabbi, M. F., & Ferdous, M. S. (2025). A Survey of Attacks on Blockchain Systems Using a Layer-based Approach. Computer Networks, 265, 111274.
22. Sello, B., Yong, J., & Tao, X. (2026). 51% attack vulnerability of nascent blockchains: a comprehensive review. Complex & Intelligent Systems, 12, Article 120.
23. NIST. FIPS 180-4. Secure Hash Standard (SHS). National Institute of Standards and Technology.
24. NIST. FIPS 202 (2015). SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions.
25. NIST. FIPS 186-5 (2023). Digital Signature Standard (DSS).