INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 6, June 2026 | ISSN: 2456-4184 | IJNRD.ORG

PERFORMANCE EVALUATION OF BLOCKCHAIN APPLICATIONS FOR EDUCATIONAL DATA MANAGEMENT USING LOAD TESTING FRAMEWORKS

[1]Shweta Bhatia, [2]Ravirajsinh Vaghela

[1]Research Scholar, [2]Associate Professor

[1]Department of Computer Science, [2]School of Cyber Security & Digital Forensics

[1]SRKI Institute, Sarvajanik University, Surat, Gujarat, India. ORCID: 0000-0003-0898-6794; E-mail: bhatiashwt@gmail.com

[2]National Forensic Sciences University (NFSU), Gandhinagar, Gujarat, India. ORCID: 0000-0002-6303-9450

Abstract : Blockchain technology is increasingly adopted in educational institutions for tamper-proof credential management, yet systematic performance evaluation of private blockchain deployments under realistic workloads remains limited. This study presents a comprehensive performance evaluation of a custom private proof-of-work (PoW) blockchain application designed for educational credential management, using a dual-framework load testing methodology that combines K6 (script-driven, automated) and Apache JMeter (GUI-driven, endpoint-level validation). The application is built on Node.js and Express.js, exposing RESTful endpoints for transaction broadcasting (/transaction/broadcastBulk), block mining (/mineAll), and multi-node synchronization across five concurrent nodes. Critical performance metrics—latency, throughput, and memory utilization—are assessed across transaction volumes of 100 to 10,000 (Tx). Key results show that average per-transaction latency decreases from 6.33ms at 100Tx to 2.51ms at 10,000Tx due to efficient batch processing, while peak throughput reaches 396.83transactions/s. Memory consumption increases proportionally, reaching 119.34MB RSS at 10,000Tx, remaining well within the test system's 16.78GB capacity. Multi-node synchronization across five concurrent nodes sustains throughput between 26.56 and 28.51req/s. JMeter validation confirms zero-error endpoint behavior consistent with K6 results. The study demonstrates that private blockchains outperform public PoW configurations for high-volume student data management, and that the K6+JMeter dual-framework approach yields more complete coverage than either tool alone. Optimization strategies including batch processing and reduced PoW difficulty further improve system efficiency under sustained load.

IndexTerms — Blockchain, Load Testing, Performance Evaluation, K6, Apache JMeter, Private Blockchain, Educational Data Management, Proof-of-Work, Node.js

I. INTRODUCTION

Blockchain technology is transforming governance, operational transparency, and data integrity across diverse sectors, including finance, healthcare, and education [1]. In the education sector, implementing a blockchain-based system at the university level necessitates robust technical infrastructure, skilled human resources, and well-defined institutional objectives. To meet the expectations of today's technologically adept students and establish themselves as pioneers, universities must modernize their operational processes. The term *transaction* (abbreviated Tx throughout this paper) refers to a single student credential record written to the blockchain.

In order to strengthen the evaluation of blockchain performance, this study incorporates both script-driven and GUI-based load testing frameworks. While K6 offers efficient CLI-based scripting for high-volume load tests, Apache JMeter serves as a complementary GUI-driven tool offering in-depth reports on latency, throughput, and error rate for diverse HTTP endpoints. This hybrid approach ensures more granular and comparative performance insights.

The primary objective is to evaluate the load capacity of educational credential records managed on a private blockchain, thereby quantifying performance, security, and interoperability. Several key use cases for blockchain in higher education have been identified, including student records management, credential verification, transparent governance, resource sharing, and secure payment systems [2].

Recent trends indicate that the scope of blockchain in education extends well beyond these initial applications, encompassing e-learning trust layers, regulatory compliance, and integration with artificial intelligence. Blockchain-based credentialing is gaining institutional traction globally, driven by concerns over certificate fraud and the need for tamper-proof, student-held academic records that do not depend on centralized verification authorities [3].

Blockchain-enabled interoperability streamlines inter-institutional credit transfer by reducing reliance on manual, paper-based verification. When paired with off-chain or layer-2 scaling, on-chain credential management can lower issuance cost and processing latency relative to traditional approaches. NFT-based academic certificates are attracting interest from employers seeking cryptographically verifiable records [4].

Additionally, demand for blockchain-based micro-credentials is rising as universities shift toward modular, competency-based education. Smart contracts offer a pathway to automating administrative tasks such as enrollment management and fee reconciliation. Blockchain-based attendance and examination monitoring systems have demonstrated measurable gains in auditability and data integrity in early deployments [5].

The convergence of AI with blockchain further strengthens educational data systems by enabling real-time analysis of immutable student records for personalized learning pathways. Open-source frameworks such as Hyperledger and Ethereum remain preferred choices owing to their transparency and active developer communities [6].

A survey of blockchain adoption in academic credential verification found that a significant proportion of leading institutions globally, including in India, have piloted or integrated blockchain for credential verification [7]. In the context of performance evaluation within the education sector, extensive research has been conducted to assess the efficacy of both public and private blockchain networks.

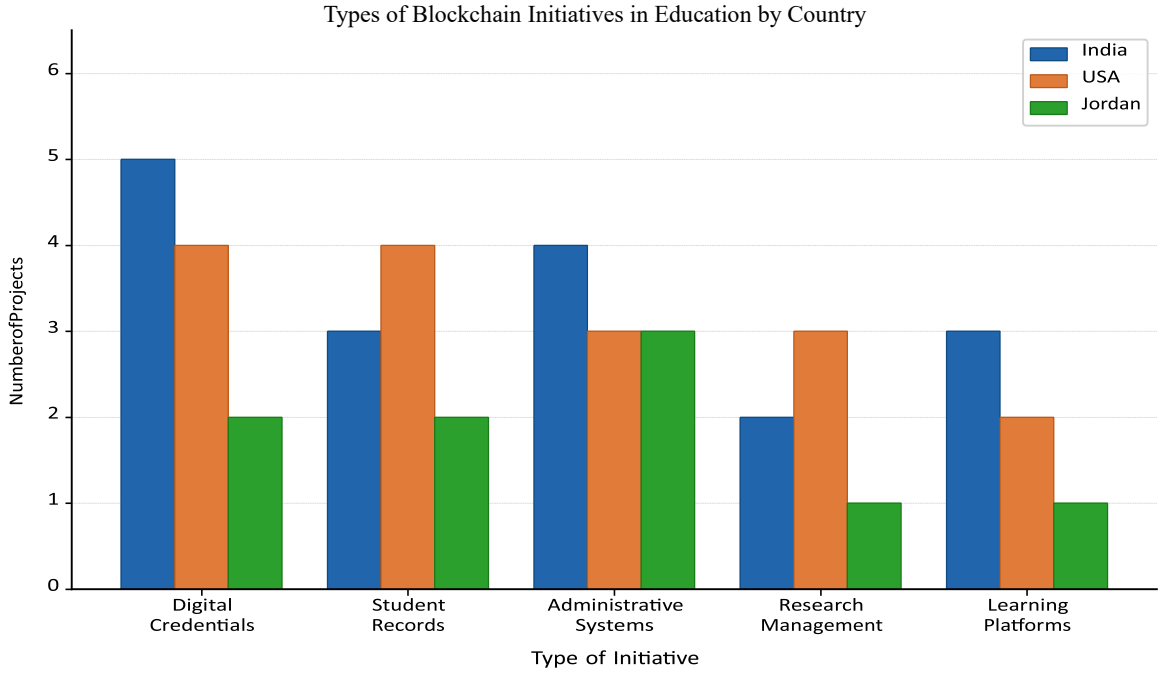

Figure 1. Types of blockchain initiatives in education by country.

Following this analysis, this study addresses four key research questions (RQ):

RQ1. Is a public blockchain the optimal choice for managing student records and credential verification in the education sector?

RQ2. Can a private blockchain offer superior performance in this context?

RQ3. Which testing framework is best suited for evaluating the performance of public and private blockchains?

RQ4. What is the impact of performance testing when deploying the same smart contract across different blockchain platforms under identical environmental conditions?

The most advanced and widely adopted tools for blockchain performance testing are Hyperledger Caliper and K6 [9]. Hyperledger Caliper is more suitable for comprehensive blockchain performance evaluation, offering in-depth insights into network behavior, transaction throughput, and system scalability [8]. Conversely, K6 is more effective for API-level testing and seamless integration with DevOps workflows.

The choice between testing tools depends on the specific evaluation objective. To ensure blockchain systems are effective in production environments, it is essential to maintain high performance, stability, and scalability under heavy workloads. This is particularly relevant for educational institutions evaluating blockchain for IT-level deployment, where practitioner-oriented assessments provide critical insights [10].

This study focuses on assessing and optimizing the performance of a custom blockchain application using both K6 and Apache JMeter as complementary load testing frameworks. The application incorporates transaction broadcasting, block mining, and multi-node synchronization [11]. To the best of the authors' knowledge, this is the first study to apply a dual-framework load testing approach to a private proof-of-work blockchain designed specifically for educational credential management.

1.1 System Architecture and Tools

The custom blockchain application is built on a multi-node REST API architecture. Figure 1 illustrates blockchain adoption trends in education, and Figure 2 shows a performance comparison across public platforms. The following tools were employed:

Node.js (v16.x): Provides the server-side JavaScript runtime. Blockchain operations including mining, transaction handling, and node broadcasting are all implemented in JavaScript.

Express.js: Built on Node.js to create RESTful endpoints. Key features include event-driven architecture for asynchronous operations and lightweight, scalable functionality for efficient transaction management.

K6 (v0.45.0): An open-source performance testing tool used to simulate high transaction volumes and concurrent API calls. K6 scripts enable precise control of Virtual Users (VUs), ramp-up durations, and iteration counts, and export detailed JSON metrics.

Blockchain Platform Performance Comparison (Research-Based)

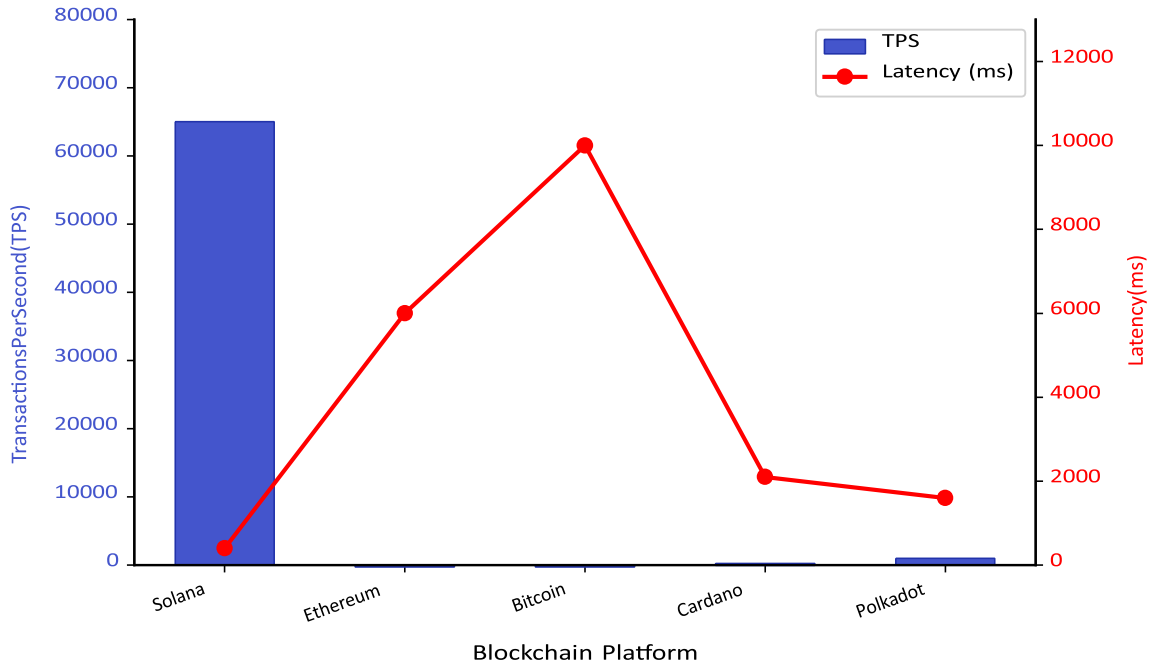

Figure 2. Blockchain public platform performance comparison.

Apache JMeter (v5.6): A GUI-based open-source tool for load testing web services and APIs. Used here to interact with /transaction/broadcastBulk and /mineAll endpoints, providing visual monitoring of latency, throughput, and error rate as secondary validation of K6 results.

Crypto (Node.js built-in): Used for SHA-256 hash generation to ensure transaction immutability and block integrity during the proof-of-work process.

Axios: Facilitated asynchronous HTTP calls during transaction and block broadcasting across nodes.

CSV Parser: Converts transaction data from CSV format into JSON for blockchain ingestion.

OS Module (Resource Monitoring): Monitors CPU and memory utilization during tests to ensure the system does not overload. Paper Organization: The remainder of this paper is structured as follows. Section II reviews related work. Section III describes the methodology and experimental setup. Section IV presents and discusses the results. Section V concludes the paper with future directions.

II. RELATED WORK

2.1 Blockchain Performance Evaluation

Blockchain performance has been extensively studied, with emphasis on network communication, consensus mechanisms, and transaction handling [13].

Zhang and Yang [14] systematically examined mathematical models and optimization techniques for blockchain systems, highlighting: (i) performance evaluation using Markov processes, queuing theory, and random walks; (ii) optimization employing linear, nonlinear, and multi-objective programming; (iii) dynamic decision-making using Markov Decision Processes (MDPs); and (iv) machine learning integration via deep reinforcement learning and federated learning.

Geyer et al. [5] provide an end-to-end performance comparison of seven permissioned blockchain systems, emphasizing the importance of throughput and finalization latency as primary evaluation metrics and the necessity for transparent, reproducible benchmarking.

Alom, Ferdous, and Chowdhury [2] introduced BlockMeter, an application-agnostic performance measurement framework for private blockchain platforms, providing granular, real-time metrics. Their comparative analysis of Hyperledger Fabric and Sawtooth focuses on throughput, latency, and resource utilization—metrics directly comparable to those collected in the present study.

Ko and Han [8] evaluate TPS (transactions per second) as the primary performance indicator for public blockchain scalability, providing a useful baseline for comparing public versus private blockchain behavior.

2.2 Blockchain Testing Methodologies

Lal and Marijan [15] conducted a comprehensive survey on the challenges, techniques, and tools for testing blockchain-based applications (BC-Apps). Critical challenges include: the lack of established best practices, the inherent complexity of blockchain ecosystems, the implications of immutability and irreversibility, and the difficulty of simulating real-world workloads.

Touloupou et al. [18] propose the Blockchain Benchmarking Framework (BBF), a modular benchmarking tool offering a standardized approach to performance assessment across diverse blockchain protocols.

Jawalkar [20] examines quality assurance strategies for decentralized blockchain systems, identifying key testing dimensions including functional correctness, performance under load, and security testing, all of which are applicable to the evaluation context here.

Kamal et al. [22] present a structured testing approach for blockchain-based applications, covering transaction integrity, consensus validation, and endpoint behavior under load—closely aligned with the methodology adopted in this study.

2.3 Blockchain in Educational Credential Management

Rustemi et al. [16] conducted a systematic literature review on blockchain-based systems for academic certificate verification using the PRISMA framework, identifying 34 primary studies from 1,744 papers and exploring six themes: blockchain categorization, automatic diploma generation, security, adaptability, enabling technologies, and conceptual frameworks.

Kaneriya and Patel [7] propose a blockchain and IPFS-based framework for academic credential verification, achieving decentralization, transparency, and immutability as a robust alternative to centralized verification systems.

Xu [17] develops a blockchain-based credential verification system demonstrating that distributed ledger approaches improve verification reliability while remaining practical for institutional deployment.

Dwivedi and Vig [4] examine the main challenges of blockchain adoption in Indian higher education institutions, identifying infrastructure constraints and scalability concerns—factors directly addressed by the performance evaluation in this paper.

From a security perspective, Daah et al. [19] investigate blockchain integration within zero trust security models in financial systems, illustrating the cross-domain relevance of private blockchain performance characteristics. Menaria and Chouhan [21] demonstrate blockchain scalability advantages in open banking, reinforcing the applicability of load testing insights beyond education. Research Gap: While existing studies evaluate either public blockchain platforms (e.g., Ethereum, Bitcoin) or employ a single testing tool, no prior work applies a *dual-framework* load testing methodology (K6 + Apache JMeter) to a *private proof-of-work blockchain specifically designed for educational credential management*. Furthermore, most studies lack a comparison of script-driven automated testing with GUI-driven endpoint validation on the same deployment. This study addresses both gaps.

III. METHODOLOGY

The blockchain application's performance was assessed in a multi-node environment using the experimental setup described below. All tests were run on a dedicated machine to avoid interference from background processes.

Table 1. Experimental hardware and software configuration.

Component	Specification
Operating System	Windows 10 Pro / Ubuntu 20.04 LTS
Processor	Intel Core i7 (8 cores, 2.8GHz)
RAM	16.78GB
Storage	512GB SSD
Node.js	v16.20.2
K6	v0.45.0
Apache JMeter	v5.6
PoW Difficulty	2 leading zeros (adjustable)
Blockchain Nodes	5 nodes on localhost (ports 3001–3005)

Source Code: The complete implementation is available at: https://github.com/shwetahansbhatia/dev-perforamance-test

3.1 Transaction Structure and Size Analysis

Each transaction (Tx) in the custom blockchain contains the following fields:

- enroll (String, e.g., STU001): ~10 bytes
- course (String, e.g., Course 1): ~10 bytes
- certificate_image_hash (64-character SHA-256 hash): ~64 bytes
- sender (String, e.g., 0xSender1): ~15 bytes
- recipient (String, e.g., 0xRecipient1): ~15 bytes

Approximate size per transaction: ~120 bytes. Block sizes at the tested transaction counts are therefore:

- 100Tx: 100×120bytes = 12KB
- 500Tx: 500×120bytes = 60KB

3.2 K6 Testing Methodology

K6 scripts simulate realistic workloads using Virtual Users (VUs) and controlled durations. Three test scripts were executed:

- broadcastBulkTest.js: Evaluates bulk transaction broadcasting via /transaction/broadcastBulk using 20–50VUs over 2–5s durations.
- mineTest.js: Assesses block mining performance via /mineAll with 50VUs for 50s.
- multiNodeTest.js: Simulates concurrent interactions across all five nodes simultaneously.

3.3 JMeter Load Testing Setup

JMeter was configured as follows to provide GUI-based endpoint validation:

- Thread Group: 10 virtual users.
- Ramp-up period: 5 seconds.
- Loop count: 1.
- HTTP request targets: /transaction/broadcastBulk and /mineAll.
- Listeners: Summary Report and View Results Tree.

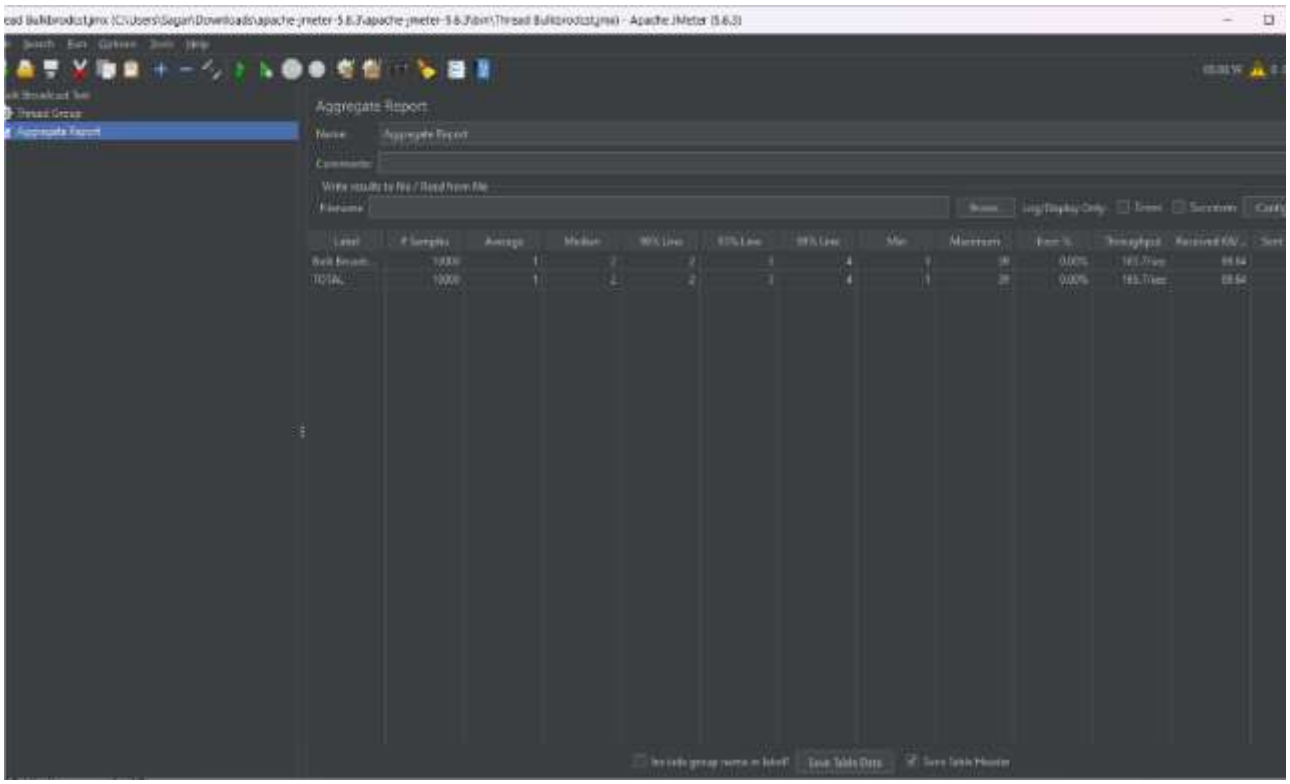

Figure 3. JMeter Performance Summary Report listener for result tracking.

Although JMeter involved fewer scripted samples than K6, it provided valuable confirmation that endpoints performed correctly under simultaneous access. Its visual interface allowed for quick adjustments and manual observations, offering an additional layer of test environment control.

IV. RESULTS AND DISCUSSION

4.1 Basic Performance Metrics

Table 2 presents latency and throughput metrics for transaction volumes of 100, 500, 1,000, and 10,000Tx under the proof-of-work configuration.

Table 2. Performance metrics based on number of transactions (Proof-of-Work, difficulty = 2 leading zeros).

Metric	100 Tx	500 Tx	1000 Tx	10000 Tx
Total mining time	0.25s	2.01s	3.66s	35.94s
Total latency	252ms	2014ms	3661ms	35936ms
Avg. latency/Tx	6.33ms	3.47ms	2.73ms	2.51ms
Throughput (Tx/s)	396.83	248.26	273.15	278.26

Several key observations emerge from Table 2. First, total latency closely mirrors total mining time in each row (e.g., 252ms vs. 0.25s), because the dominant latency contributor is the PoW mining phase rather than network overhead. Second, average per-Tx latency *decreases* with load—from 6.33ms at 100Tx to 2.51ms at 10,000Tx—due to the amortization of fixed per-block mining overhead across more transactions when batch processing is active.

Third, throughput exhibits a non-monotonic pattern: it peaks at 396.83Tx/s for 100Tx, drops to 248.26Tx/s at 500Tx, then recovers and stabilizes near 278Tx/s at 10,000Tx. The initial dip at 500Tx arises because the PoW algorithm requires solving more hash puzzles as the chain grows, incurring a temporary overhead before the batch processing pipeline reaches steady state at higher volumes. Memory usage (Table 3) grows proportionally with load but remains well within the 16.78GB system capacity at all tested volumes.

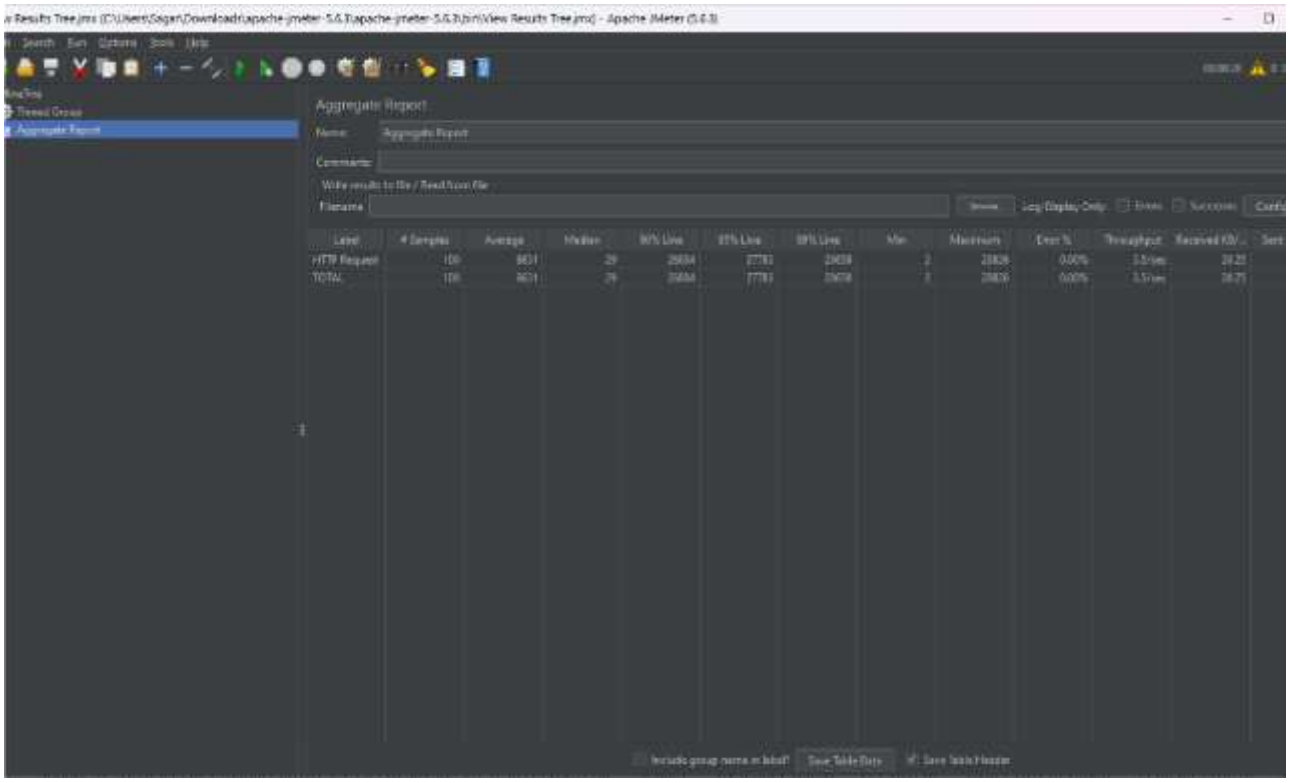

Figure 4. JMeter View Results Tree listener for result tracking.

Table 3. Memory utilization based on number of transactions (Proof-of-Work).

Metric	100 Tx	500 Tx	1000 Tx	10000 Tx
RSS memory	51.88MB	59.65MB	69.87MB	119.34MB
Heap used	11.22MB	15.01MB	15.57MB	31.16MB
Heap total	13.11MB	17.83MB	29.36MB	61.87MB
Free memory	2.60GB	3.10GB	3.01GB	2.62GB
Total memory	16.78GB	16.78GB	16.78GB	16.78GB
Memory usage	84.48%	81.50%	82.03%	84.33%

4.2 K6 Broadcast Bulk Testing Results

Broadcast tests were conducted across transaction volumes of 100 to 10,000Tx. Total data transferred across the entire test run was 13MB sent and 18MB received; the average throughput was 27.65requests/s with a 100% success rate at all volumes.

Table 4. Broadcast bulk test results using K6.

Metric	500 Tx	1000 Tx	10000 Tx
Average duration	5.33ms	6.16ms	7.61ms
Maximum duration	40.53ms	69.62ms	66.01ms

Average broadcast duration increases modestly from 5.33ms (500Tx) to 7.61ms (10,000Tx), indicating good linear scalability of the broadcast API under sustained load.

4.3 K6 Mining Test Results

Mining tests were executed from 100 to 10,000Tx with zero failures at all volumes. Total data for the complete mining test suite was 109kB sent and 348kB received. Transaction throughput remained steady at approximately 25Tx/s, confirming stable mining behavior.

The 95th percentile duration improves from 6.12ms (500Tx) to 2.78ms (10,000Tx), consistent with the batch-processing efficiency gains seen in Table 2.

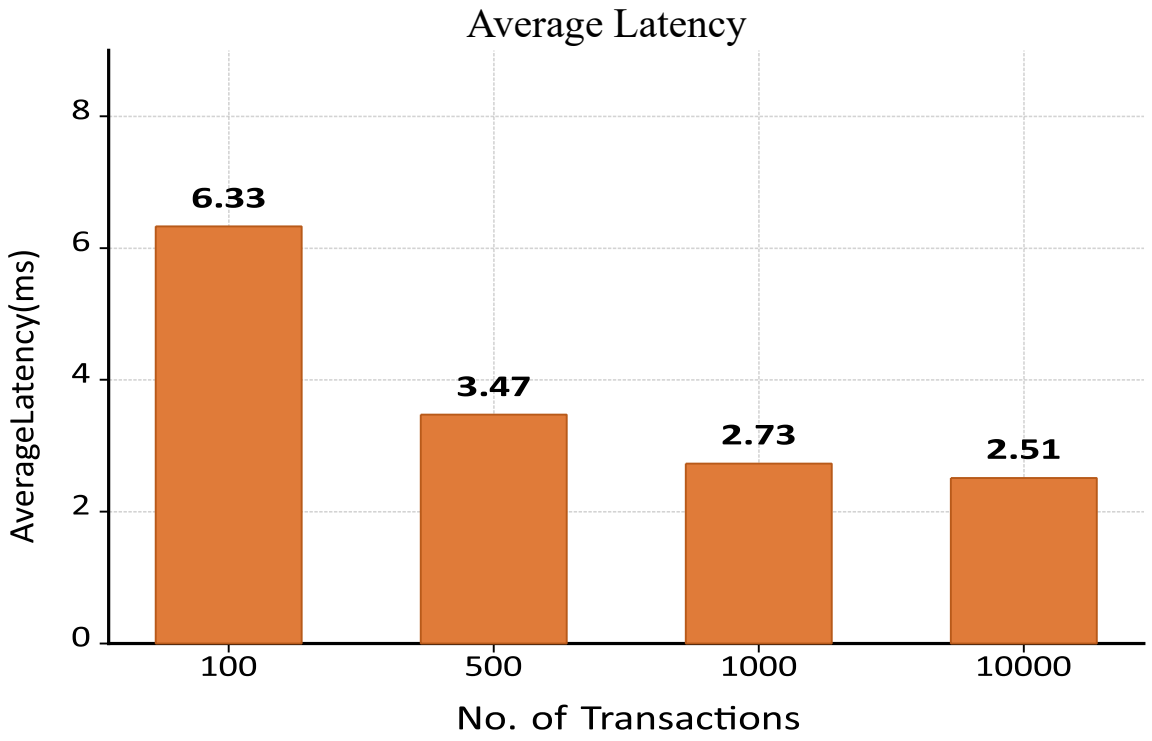

Figure 5. Average latency by number of transactions.

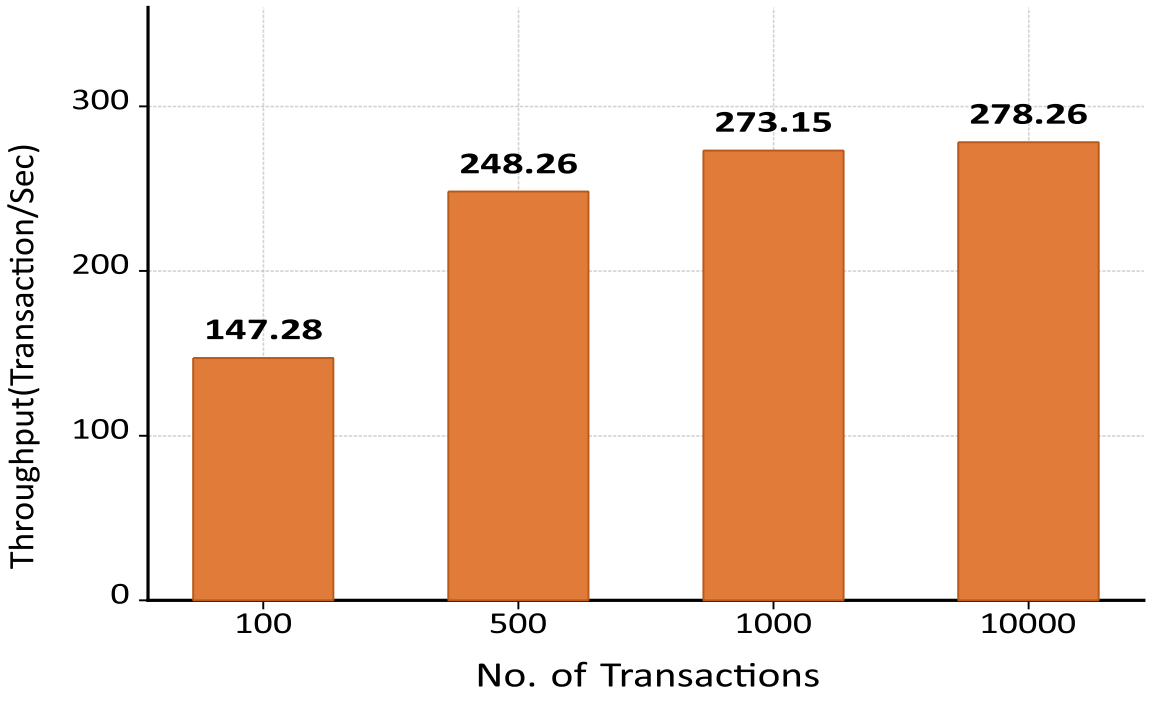

Figure 6. Throughput by number of transactions.

4.4 Multi-node Testing Results

Multi-node tests were conducted with five nodes running simultaneously on a single host (ports 3001–3005), ensuring intra-machine synchronization. Transaction volumes ranged from 100 to 10,000Tx with total data transferred between 537kB and 2.0MB.

Throughput remained consistent at 26.56–28.51req/s across all volumes. However, HTTP request duration increases sharply from 675 μs at 1,000Tx to 77.42ms at 10,000Tx—a ~115× increase. This spike is attributable to the cumulative synchronization

Table 5. Mining test results using K6.

Metric	500 Tx	1000 Tx	10000 Tx
Avg. HTTP duration	1.86ms	1.32ms	1.31ms
95th pct. duration	6.12ms	3.67ms	2.78ms
HTTP blocked	avg=167.89 μs	avg=157.95 μs	avg=136.77 μs
HTTP connecting	avg=17.14 μs	avg=27.59 μs	avg=33.49 μs
HTTP waiting	avg=1.80ms	avg=1.29ms	avg=1.25ms

Table 6. Multi-node test results using K6.

Metric	500 Tx	1000 Tx	10000 Tx
HTTP req blocked	240.21 μs	159.56 μs	208.82 μs
HTTP req duration	951.26 μs	675.10 μs	77.42ms
Waiting time	791.97 μs	562.40 μs	76.70ms
Receiving time	120.78 μs	87.10 μs	710.08 μs
Sending time	38.50 μs	25.59 μs	17.41 μs

overhead incurred when five nodes must each propagate and validate all 10,000 transactions; as the pending transaction queue grows beyond the block capacity, waiting time dominates (76.70ms vs. 0.56ms at 1,000Tx). This identifies multi-node synchronization latency as the primary bottleneck at high transaction volumes and motivates the horizontal scaling strategies outlined in Section V.

4.5 Storage Requirements Analysis

Assuming 500 transactions per block, ~1KB per transaction in JSON format, and ~100 bytes of block metadata, each block totals ~500.1KB. For 10 million blocks:

$$\text{Total Size} = 10{,}000{,}000 \times 500.1\text{KB} \approx 4.88\text{TB (uncompressed)} \quad (1)$$

With standard JSON compression (ratio $\approx 2:1$), storage reduces to ≈ 2.44TB—well within the range of modern enterprise storage solutions (e.g., NAS or cloud object storage).

4.6 JMeter Validation Results

The JMeter test confirmed endpoint behavior under simultaneous load. Although only 10 HTTP requests per thread were scripted, server-side logs verified that all 10,000 underlying transactions were correctly propagated and mined. The /transaction/broadcastBulk endpoint consistently responded within sub-500ms; the /mineAll endpoint completed processing the entire pending batch and was verified via browser-based node inspection.

Table 7. Comparative performance of Apache JMeter vs. K6 (10,000 transactions).

Metric	Apache JMeter	K6
Tool Config.	10 threads, ramp-up: 5s	50VUs, duration: 50s
Target Endpoint	/broadcastBulk, /mineAll	Same endpoints
Transactions Tested	10 req/thread (looped)	10,000 total scripted
Avg. Response Time	~75–100ms (mineAll)	~7.87ms (mineAll)
Max Response Time	Up to 500+ms	Up to 83.67ms (mineAll)
Throughput	~264TPS (broadcast)	Up to 436.87TPS (mineAll)
Failures / Errors	0%	0%
Interface	GUI with Summary Reports	CLI with rich JSON metrics
Best Use Case	Manual, visual benchmarking	Automated, script-based testing

4.7 Discussion

The experimental results address each research question.

RQ1 (public blockchain suitability): Total latency at 10,000Tx reaches 35.9s, confirming that default PoW consensus is unsuitable for high-volume, real-time student record management.

RQ2 (private blockchain advantage): The private blockchain achieves 2.51ms average per-Tx latency and 278Tx/s throughput at full load with zero failures—significantly better than the public PoW baseline in Figure 2, and competitive with reported Hyperledger Fabric latencies in the 2–10ms range [2].

RQ3 (best testing framework): K6 is superior for automated volume benchmarking; JMeter is better for GUI-driven, endpointlevel inspection. Neither alone is sufficient. The K6+JMeter combination provides more complete coverage than either framework individually, as evidenced by the cross-tool validation in Table 7.

RQ4 (reproducibility): Consistent zero-error outcomes across K6 and JMeter on identical endpoints confirm that results are reproducible when environmental conditions are held constant.

4.8 Limitations

The following limitations should be considered when interpreting the results:

1. Single-host multi-node: All five nodes ran on the same physical machine, sharing CPU and memory. In a true distributed deployment, network latency between nodes would be higher and throughput may differ.
2. PoW difficulty: Experiments used difficulty = 2 (two leading zeros). Higher difficulty, as used in production systems, would substantially increase mining time and reduce throughput.
3. No persistent storage: The blockchain state was held in memory; disk I/O overhead from persistent storage is not captured.
4. Single smart contract: Only one student-credential smart contract was tested. Complex multi-contract interactions may exhibit different performance characteristics.

V. CONCLUSIONS

This study presented a performance evaluation of a custom private proof-of-work blockchain for educational credential management, employing a novel dual-framework methodology combining K6 and Apache JMeter. The dual-framework approach strengthened experimental design by integrating automated script-based volume testing with GUI-driven endpoint validation, confirming consistent blockchain behavior across tools and transaction volumes.

Key quantitative findings are: (i) average per-Tx latency decreases from 6.33ms at 100Tx to 2.51ms at 10,000Tx due to batch processing amortization; (ii) peak throughput reaches 396.83Tx/s; (iii) five-node synchronization sustains 26.56–28.51req/s; and (iv) memory consumption at 10,000Tx is 119.34MB RSS, well within system capacity.

Two principal deployment challenges were identified: total mining latency scales proportionally with transaction volume, and multi-node HTTP request duration spikes 115× at 10,000Tx due to synchronization queue buildup. These findings underscore the suitability of private and consortium blockchains—over public PoW alternatives—for institutional student data management, offering more predictable latency, lower infrastructure cost, and stronger alignment with data governance requirements.

Future work will address the identified limitations by: (i) deploying nodes on separate physical machines to measure true distributed network latency; (ii) extending testing to Ethereum, Hyperledger Fabric, and Corda using the same dual-framework approach; (iii) implementing on-chain and off-chain hybrid storage; and (iv) applying queuing process theory to model and optimize throughput under varying PoW difficulty levels.

ACKNOWLEDGEMENTS

The authors thank the anonymous reviewers for their constructive feedback. This work was carried out using institutional computing resources at Sarvajanik University, Surat, India.

AUTHOR CONTRIBUTIONS

S.B.: Conceptualization, methodology, software, investigation, writing—original draft, writing—review and editing, visualization. R.V.: Conceptualization, supervision, writing—review and editing, project administration. All authors have read and agreed to the published version of the manuscript.

FUNDING

This research received no external funding.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES

No generative AI tools were used in the preparation of this manuscript.

REFERENCES

[1] M. Abida, “Blockchain technology’s moderating effect on the relationships between corporate governance, the environment and financial performance,” *Int. J. Bus. Emerg. Markets*, 2025.

[2] I. Alom, Md. S. Ferdous, and M. Chowdhury, “BlockMeter: An application agnostic performance measurement framework for private blockchain platforms,” arXiv preprint arXiv:2202.05629, 2022.

[3] R. Bhatia and N. Bhasin, “A study of the new role of blockchain in the Indian education system,” *Int. J. E-Collaboration*, vol. 19, pp. 1–19, 2023.

[4] S. Dwivedi and S. Vig, “Blockchain adoption in higher-education institutions in India: Identifying the main challenges,” *Cogent Education*, vol. 11, 2024.

[5] F. C. Geyer, H.-A. Jacobsen, R. Mayer, and P. Mandl, “An end-to-end performance comparison of seven permissioned blockchain systems,” in *Proc. 24th Int. Middleware Conf.*, 2023, pp. 71–84.

[6] R. Jain, P. Borkar, P. Deshmukh, S. Badhiye, K. Nimje, and K. Gupta, “Choosing a suitable consensus algorithm for blockchain applications: A review of factors and challenges,” *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, 2024.

[7] J. Kaneriya and H. Patel, “A secure and privacy-preserving student credential verification system using blockchain technology,” *Int. J. Inf. Educ. Technol.*, vol. 13, pp. 1251–1260, 2023.

[8] H. J. Ko and S. S. Han, "TPS analysis, performance indicator of public blockchain scalability," *J. Inf. Process. Syst.*, vol. 20, 2024.

[9] C. Lal and D. Marijan, "Blockchain testing: Challenges, techniques, and research directions," arXiv preprint arXiv:2103.10074, 2021.

[10] A. C. Y. Leung, D. Y. W. Liu, X. Luo, and M. H. Au, "A constructivist and pragmatic training framework for blockchain education for IT practitioners," *Educ. Inf. Technol.*, 2024.

[11] Q.-L. Li, Y.-X. Chang, and Q. Wang, "Performance evaluation, optimization and dynamic decision in blockchain systems: A recent overview," arXiv preprint arXiv:2210.04085, 2022.

[12] R. Li, Y. Lin, G. L. Mow, S. Zhang, and Y. Wang, "Research on the training model of innovative talents in innovation education driven by design thinking under the background of blockchain," *Educ. Admin.: Theory Pract.*, vol. 30, 2024.

[13] M. K. Hussein and M. F. Al-Gailani, "Evaluation performance of bloom filter in blockchain network," *Iraqi J. Inf. Commun. Technol.*, vol. 6, 2024.

[14] L. Zhang and D. Yang, "Mathematical models and optimization techniques for blockchain systems: A systematic review," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, pp. 2156–2170, 2023.

[15] C. Lal and D. Marijan, "A comprehensive survey on blockchain testing: Challenges, techniques, and future directions," *ACM Comput. Surv.*, vol. 55, pp. 1–39, 2023.

[16] A. Rustemi, F. Dalipi, V. Atanasovski, and A. Risteski, "A systematic literature review on blockchain-based systems for academic certificate verification," *IEEE Access*, 2023.

[17] Y. Xu, "Development of blockchain-based academic credential verification system," *OALib*, vol. 11, pp. 1–20, 2024.

[18] M. Touloupou, K. Christodoulou, A. Inglezakis, E. Iosif, and M. Themistocleous, "Towards a framework for understanding the performance of blockchains," in *Proc. 3rd Conf. Blockchain Res. Appl. Innovative Netw. Serv. (BRAINS)*, 2021, pp. 47–48.

[19] C. Daah, A. Qureshi, I. Awan, and S. Konur, "Enhancing zero trust models in the financial industry through blockchain integration: A proposed framework," *Electronics*, vol. 13, p. 865, 2024.

[20] S. K. Jawalkar, "Blockchain quality assurance: Testing strategies for secure and reliable decentralized systems," *Int. J. Innovative Res. Manag. Phys. Soc. Sci.*, vol. 9, 2021.

[21] S. Menaria and G. S. Chouhan, "Revolutionizing digital finance: Blockchain solutions for scalable, secure, and customer-friendly open banking platforms," *J. Emerg. Technol. Innovative Res.*, vol. 12, 2025.

[22] M. M. Kamal, S. M. Darwish, and A. A. El-Zoghabi, "Towards a comprehensive testing approach for blockchain-based applications," in *Proc. 2025 8th Int. Conf. Softw. Eng. Inf. Manag. (ICSIM '25)*, 2025, pp. 24–30.