Journal of Business and Management Studies (JBMS)
ISSN: 2709-0876
DOI: 10.32996/jbms
Journal Homepage: www.al-kindipublisher.com/index.php/jbms

JBMS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Performance Optimization in Multi-Machine Blockchain Systems: A Comprehensive Benchmarking Analysis

Mohotasim Billah[1]✉, Sadia Sharmeen Shatyi[2], GM Alamin Sadnan[3], Kazi Nehal Hasnain[4], Joynal Abed[5], Maksuda Begum[6] and Kazi Sharmin Sultana[7]

[1]*Master of Science in Computer Science, Washington University of Virginia(WUV)*
[2]*Master of Architecture, Louisiana State University*
[3]*Cybersecurity Analyst & Patient Care Technician, Farmingdale State College*
[4]*Master of Science in Information Technology (MSIT), Westcliff University, Irvine, CA*
[5]*Master of Architecture, Miami University, Oxford, Ohio.*
[6]*Master of Business Administration, Trine University.*
[7]*MBA in Business Analytics, Gannon University, Erie, PA*
Corresponding Author: Mohotasim Billah, Email: mb23301@wuv.edu

| ABSTRACT

The increase in decentralized applications has put blockchain technology in a very crucial position in the industries and various sectors such as finance, healthcare, and logistics in the United States. The need to optimize performance in blockchain systems and especially those operating on more than one machine grows along with the demand for secure and distributed ledgers. The main objective of the study was to develop a predictive dynamic benchmarking framework that allows optimizing the performance of the multi-machine blockchain systems. Through machine learning algorithms, which include random forests, gradient boosting, and support vector machines. The dataset used in this study is made up of high-resolution performance logs of a simulated multi-machine blockchain setting after having run 30 days of logs across both a permissioned (Hyperledger Fabric) and an accessible (Ethereum) environment. The main measurements observed are the block propagation times which is how long it takes a new block to be broadcasted to every single node of the network, along with the transaction confirmation latencies which will be the time difference between the moment a transaction is submitted and that same transaction is finally confirmed on the distributed ledger. The selection of three supervised machine learning models to be deployed was based on the capacity to work with high dimensionality, nonlinear as well as possible imbalanced performance classification problems. Several evaluation measures were calculated to have an overall picture of the classification ability of each model. Accuracy measures the level of overall accuracy of predictions, whereas precision, recall, and F1-score offer information about performance on minor classes needed to locate rare but serious blockchain performance reductions. The Confusion matrix analysis was employed in identifying the particular types of misclassifications. The Random Forest model outperformed the other two models, attaining the highest accuracy, with near-identical precision, recall, and F1 score values, indicating consistent and reliable predictions across all classes. Gradient Boosting performs closely and achieves a strong balance across other metrics, suggesting it is nearly as effective and particularly useful for more nuanced prediction tasks or imbalanced datasets. The application of optimized multi-machine blockchain performance is particularly applicable in many of the large sectors in the United States, in which fast, secure, and scalable digital infrastructure is in demand. Infrastructure-wise, it is possible to use the results of machine learning-powered benchmarking to dictate more optimal hardware and software settings of blockchain nodes set up in the U.S.-based data centers and cloud environments. From a public policy perspective, the research implications for performance optimization are largely compatible with a series of current federal efforts aimed at enhancing the digital trust infrastructure. In the future, it is possible to research the integration of Machine Learning-based benchmarking systems into real-time optimization engines, whereby we would be able to tune the behavior of nodes on the fly, given the latest telemetry of performance.

| KEYWORDS

Blockchain Optimization, Machine Learning, Benchmarking Analysis, Performance Prediction, Distributed Ledger, Consensus Protocols, Resource Efficiency, System Scalability

| ARTICLE INFORMATION

ACCEPTED: 01 November 2024 **PUBLISHED:** 17 November 2024 **DOI:** 10.32996/jbms.2024.6.6.18

I. Introduction

Background

The blockchain technology is quickly becoming an innovative infrastructure in the U.S. that reinvents industry like finance, health record management, and the transparency of the government through the decentralized and tamper-proof storage of data (Alladi et al., 2020; Aloqaily et al., 2020). The backbone of this transformation is the multi-machine blockchain deployments in which nodes are spread over physical or cloud-based solution environments. Nonetheless, these deployments also acquire complex operational inefficiencies. Problems like latency in communication between the nodes in a distributed system, unbalanced loads on the transactions, and resource contention are major factors that reduce the performance of a system (Govindasamy& Antonidoss, 2022; Ajayi & Saadawi, 2023). As an example, a report published by Deloitte in 2023 stated that 61 percent of blockchain companies headquartered in the U.S. claimed latency and throughput on the network to be their major issues when implementing enterprise blockchain products.

Al-Refaie & Hawadi (2025) found that consensus protocols in multi-node architecture, such as Proof-of-Work (PoW) or Practical Byzantine Fault Tolerance (PBFT), also contribute to the performance lag. The requirement of inter-node agreement ensures that the messages have to be exchanged several times, and this is a weakness of the network that affects real-time operation. There also arise inconsistencies in communication patterns, particularly in a situation where the nodes are distributed in geographically distributed regions, which leads to differential propagation delays. This means that it is hard to evaluate the system efficiency based on old-fashioned standards that consider a specific parameter one by one, such as transaction rate or block size. More versatile performance assessment models are required, which would consider interdependencies of these elements, therefore making it possible to fully comprehend blockchain behavior under pressure (Feng et al., 2024).

Furthermore, the distinctiveness of the workloads that blockchain platforms handle, such as micropayments, smart contract execution, requires a versatile but strong benchmarking method. Industries in the United States, whose regulatory environment is rapidly changing, and digital currency penetration is rapidly increasing, should focus on scalable and performance-oriented blockchain solutions (Ge et al., 2020). NIST (2006, para 1) points out the significance of system-level benchmarking of the distributed settings, particularly the setting where critical infrastructures are being utilized. Thus, the issue of performance shortcomings in multi-machine blockchain systems is a challenge to the technological competitiveness of the United States and its cybersecurity, rather than a technical challenge (Hafeez et al., 2023; Chouksey et al., 2023).

Problem Statement

Gupta et al. (2021) highlighted that despite advancements in blockchain technology, benchmarking tools have not evolved at the same pace, often resulting in insufficient performance evaluations in real-world, multi-machine setups. Most existing benchmarking frameworks—such as Block-bench or Caliper—primarily analyze system throughput or latency in isolated environments, failing to capture the interactions between consensus protocols, communication layers, and dynamic workloads. Ge et al. (2020) added that these tools typically assume uniform conditions that rarely hold in large-scale deployments where nodes experience variable latencies, resource bottlenecks, and unpredictable traffic. As a result, optimization strategies derived from such benchmarks are often inadequate when applied to distributed, multi-machine systems used in actual business or governmental scenarios.

Moreover, a lack of predictive modeling in current benchmarking methodologies further limits their utility. Without the ability to anticipate how performance might degrade or improve under varying workloads, administrators are left to rely on trial-and-error or overprovision hardware resources (Hakeem & Kim, 2025). This inefficiency is particularly costly in the U.S. enterprise landscape, where blockchain platforms underpin mission-critical applications such as identity verification, interbank transfers, and medical data integrity. For example, JPMorgan Chase's Quorum blockchain platform processes hundreds of transactions per second and requires performance predictability to meet regulatory and service-level agreements. Without accurate benchmarks that reflect system-wide interactions, organizations are hindered in their ability to preemptively scale, optimize, or troubleshoot their blockchain systems (Elghaish et al., 2023).

The current state of benchmarking also fails to incorporate adaptive, learning-based approaches that could enhance predictive accuracy (Han et al., 2022). This is a significant gap, considering the widespread availability of performance monitoring data in modern distributed systems. The integration of machine learning for performance classification and anomaly detection has

proven successful in cloud infrastructure management, yet it remains underutilized in the context of blockchain optimization (Ge et al., 2020). This research aims to bridge that gap by employing machine learning to develop a comprehensive benchmarking model that not only evaluates current system performance but also predicts future states based on historical and real-time metrics.

Objective

The primary goal of this study is to create a predictive and dynamic benchmarking framework that enables the performance optimization of multi-machine blockchain systems. By leveraging machine learning algorithms—such as random forests, gradient boosting, and support vector machines—this research seeks to classify performance patterns, detect inefficiencies, and forecast system behavior under various configurations. Using real-world data from multiple blockchain platforms and testbeds deployed in simulated cloud environments, the proposed model will offer a nuanced understanding of how system variables interact over time, informing smarter resource allocation and configuration decisions.

This paper further aims to bridge the gap between benchmarking theory and practical system management. By analyzing core performance metrics such as transaction per second (TPS), block propagation time, and CPU/memory utilization, the study will generate a performance taxonomy that distinguishes between healthy, degraded, and critical states in a blockchain system. These classifications will be validated against known performance baselines and used to train predictive models capable of suggesting optimization strategies before issues arise. In doing so, we create a continuous feedback loop that integrates monitoring, benchmarking, and predictive optimization—a first in the realm of multi-node blockchain performance research.

From a broader perspective, this research also strives to align with national priorities in blockchain standardization and infrastructure modernization. The U.S. Department of Commerce has advocated for greater adoption of blockchain technology across federal agencies, but emphasizes the need for scalable and dependable systems. By delivering an intelligent benchmarking solution, this study supports not only the immediate technical needs of system administrators but also the strategic vision of a blockchain-enabled digital economy. Through rigorous experimentation and data-driven analysis, the proposed model aims to become a cornerstone for future research and deployment standards.

Research Significance

Hafeez et al. (2023) indicated that the significance of optimizing performance in multi-machine blockchain systems cannot be overstated, especially when considering the vital roles these systems are beginning to play across U.S. industries. In sectors such as finance, where high-frequency trading and real-time settlement are essential, even a slight latency can result in millions of dollars in losses or increased systemic risk. According to a 2024 report by the Federal Reserve, over 50% of banks in the United States are actively testing or deploying blockchain technologies to enhance transaction security and processing efficiency (Feng et al, 2024). However, many of these deployments face scalability issues, particularly when operating across geographically dispersed machines. Optimizing performance through intelligent benchmarking allows financial institutions to handle higher transaction volumes without compromising system integrity or speed, making them more resilient in fast-paced markets (Elghaish et al., 2023).

In logistics and supply chain management, the U.S. Department of Transportation has identified blockchain as a pivotal tool for tracking goods, verifying origins, and ensuring compliance with federal regulations (Hawashin et al., 2024). However, supply chains are inherently distributed and often involve dozens, if not hundreds, of nodes spread across different networks. This complexity introduces challenges related to node synchronization, data consistency, and latency. Efficient benchmarking tools and predictive models help address these challenges by simulating real-world conditions and recommending configurations that enhance system responsiveness. By optimizing how data is shared and validated across machines, companies can reduce delivery delays, minimize losses, and meet federal logistics standards with improved transparency and accountability (Jaffar & Acikgoz, 2023). According to IBM, logistics companies using blockchain can reduce documentation errors by 75% and operational costs by up to 30%—figures that can only be realized with well-optimized systems (Jakir et al., 2023).

According to Jasim & Hadi (2023), performance-enhanced blockchain systems can benefit government services in an enormous way. There are applications like land registry, identity management, and dispersing of the public funds for which both high security and real-time data processing are required. The U.S. General Services Administration (GSA), an agency that works to digitize processes in federal acquisition, is testing the use of blockchain to streamline the work across many interdependent systems. These initiatives are neither well optimized nor performant; thus, they can fail because of the slow consensus algorithms, overloaded nodes, or weak scalability (Hossain et al., 2024). With the use of machine learning enhanced benchmarking models, agencies will be able to optimize the computational resources, predict aspects of performance bottlenecks, as well as scale the services easily. Kai et al. (2024) also assumed that the outcome is not only cost reduction and higher community confidence but also substantial conformity to the executive order on the responsible development of digital assets signed by President Biden. In this way, this study is of critical importance because it allows blockchain systems to respond to the challenging requirements of the U.S. operations in both the public and private sectors.

II. Literature Review

Multi-Machine Blockchain Architecture

An analysis done by Soudan et al. (2025) indicates that multi-machine blockchain systems architecture is the apparent direction of developing distributed ledger technology (DLT) that is strategically intended to boost a system to be fault-tolerant, augment parallelism of transactions, and achieve expanded geographic coverage. Such systems are characterized by distributed ledgers being implemented, in various cases on different machines and even in different data centers or clouds to implement resilience towards single points of failure. These nodes synchronize them with consensus protocols and communication patterns in the form of gossip to ensure that the blockchain is in a coherent state (Rana et al., 2023). This architecture may commonly be used with the help of container orchestration systems such as Kubernetes or Docker Swarm, which enables a scalable deployment that fits within cloud infrastructure. The Hyperledger Fabric framework, spearheaded by the Linux Foundation and backed by multinational companies like IBM, Intel, has made the idea of modular architecture a default, where ordering nodes, peer nodes, as well as certificate authority nodes, are dedicated to separate machines. The advantage of this structure is the increased level of customization and security, as well as the complexity of coordination and performance optimization (Wang et al.,2023).

The deployment of blockchain technologies in the enterprise and government systems in the United States gained momentum and led to the rise of demand in scalable and multi-node infrastructures (Yang et al., 2022). A 2023 report of the National Institute of Standards and Technology (NIST) further found that more than 60 percent of the blockchain deployments considered in federal pilot programs ran in multi-machine configurations to provide availability, redundancy, as well as federal IT modernization compliance (Yi et al.,2024). These arrangements can include hybrid cloud infrastructures, in which nodes can be on-premise and others in the cloud, such as AWS GovCloud or Microsoft Azure Government. Although the distributed characteristic of such systems enables decentralized control and allows them to remain operational regardless of machine failure, it also necessitates the use of complex synchronization, failover recovery, and communication mediums across different machines to guarantee the integrity of the ledger, irrespective of the conditions in the system (Rahman et al., 2023).

Furthermore, various machines also require multi-machine architecture to deal with the diversity of hardware and network attributes, most notably in case of blockchain systems that are distributed between several groups or jurisdictions (Zhou et al.,2024). As an example, the financial institutions in the U.S. need to operate under different security standards (e.g., PCI-DSS on payment data), thus affecting the design of node-to-node trust and encryption protocols. Such fragmentation makes architecture design complicated because developers have to consider the tradeoff between system interoperability and security, and compliance policies (Yang et al., 2022). With more complex and bigger systems, there is an obvious need to have automated and intelligent ways to measure and tune the performance of the system, considering the sheer amount of hardware and workload dynamics across distributed nodes (Hasan et al., 2025)

Performance Challenges

Inherent limitations to the performance of multi-machine blockchain systems are critical barriers caused by latency inefficiencies, throughput limits, overhead of consensus protocols, and resource contention. Inter-node communication is a bottleneck in multi-node configurations when the number of nodes rises (Mehta et al., 2020). In general, network latency, especially in geographically decentralized deployments, can extremely delay the propagation of blocks and the finalization of transactions. The U.S. Department of Homeland Security (DHS) revealed that blockchain latency problems used in customs processing and border security might result in a delay in the application at its peak of more than 40 percent of a given application (Soudan et al., 2025). These delays are further increased with the consensus algorithms such as Proof-of-Work (PoW) or Practical Byzantine Fault Tolerance (PBFT), which involve several rounds of communication and cryptographic verifications to reach agreement and therefore attach quantifiable delay to each transaction (Li et al.,2024).

Throughput also emerges as an urgent issue, especially in specialized applications in the enterprise or in the public sector, where large numbers of transactions need to be processed within short periods. According to a 2022 study by MIT Digital Currency Initiative, Bitcoin has an average of 7 transactions per second (TPS) and Ethereum has about 30 TPS, although 5 to 10 TPS is more typical; the permissioned blockchain systems such as Hyperledger Fabric can achieve up to 3,000 TPS in the ideal, tightly controlled setting (Kumar et al., 20225). Nevertheless, real deployments in the wild perform below such figures and are limited by bottlenecks of resources and non-optimal setups. As noted by Kai et al., (2024), in the U.S. healthcare sector, especially, blockchain networks that have been piloted to enable sharing of data on patients between institutions have documented loss of up to 50 percent of throughput levels when the systems are undergoing maintenance or software updates and this contravenes reliability and services delivery.

Competition for resources also adds to the problem of performance in a multi-machine blockchain. Performance can be reduced when many nodes compete over a finite pool of CPU cycles, memory bandwidth, or access to latent I/Os, particularly in the case of cloud-based virtualization (Kim et al., 2018). Many systems have missed service-level expectations, failing to launch the necessary virtual machines and engage effective load balancing mechanisms (Ghribi et al.,2020), and non-deterministic behavior in smart contracts execution can even result in highly unpredictable performance results when the workload is not distributed evenly. The problems presented provide the argument that intelligent benchmarking and tuning models are required to be able to adapt system parameters on-the-fly (Jasim and Hadi, 2023).

Machine Learning in Distributed Systems Performance Tuning

Chouksey et al. (2023) believed that machine learning (ML) has been demonstrated to have good potential towards improving the performance of distributed computing systems through proactive management of resources, detecting anomalies, and keeping a balance on dynamic resources. Recently, ML models have been employed in cloud computing and data centers to predict traffic surge, detect performance degradation, and resource scheduling (Ajayi & Saadawi, 2023). Indicatively, the DeepMind project developed by Google is well-known for having decreased the amount of energy required at data centers by 40 percent by optimizing ML-based cooling systems. Within the U.S., the Department of Energy (DOE) and the Defense Advanced Research Projects Agency (DARPA) are major federal agencies involved in AI/ML research to enhance the efficiency of the underlying systems supporting these missions, which include distributed ledger technologies (El-Refaie & Hawadi, 2025).

In this case of blockchain, nevertheless, the implication of ML remains at an early stage. The majority of previous usages have centered on security and fraud prevention, including the use of supervised learning techniques with smart contracts, like finding abnormal transactions or even warning about smart contract security issues (Govindasamy & Antonidoss, 2022). That aside, a rising stack of evidence is showing that ML has the potential for use in the performance tuning of distributed systems. As examples, there has been the use of reinforcement learning methods to dynamically optimize parameters such as block size and mining difficulty over testbed environments to achieve higher transaction throughput and no manual reconfiguration. In the same way, support vector machines and decision forests have been utilized in cloud systems to forecast the pattern of usage of the CPU and identify resource starvation and load imbalance between virtual machines (Govindasamy & Antonidoss, 2022).

More recent scholarly work has first started to look at how these Machine Learning approaches can be scaled to multi-machine blockchain systems. At Stanford University, researchers produced a model that forecasts gas fee spikes in Ethereum based on time-series forecasting, and a study at the University of California used clustering algorithms to cluster transaction types based on resources utilized (Feng et al.,2024). These initiatives, as good as they are, have so far been more academic or simulations. There is not much that has been implemented operationally in large-scale networks in production blockchain networks. It is worth noting, however, that these seminal works pose a prospect of ML models not only being observational of distributed blockchain systems but actively enabling their overall optimization as a whole, when being supplemented by real-time data ingestion and feedback loops (Hafeez et al.,2023).

Gaps in Research

Recent research has shown that there is a lack of understanding of how predictive benchmarking techniques can be applied to real-world multi-machine blockchain systems, despite the increased work on blockchains and ML. The majority of benchmarking available tools, including Hyperledger Caliper or Blockbench, are designed to perform testing in a set amount of time without the opportunity to be dynamic and adaptive (Feng et al.,2024). They normally evaluate performance by using a measure of one aspect, such as throughput or latency, without considering any relationships of system parameters and the dynamic measurement relating node interactions in a multi-machine system architecture (Hafeez et al., 2023). Consequently, such benchmarks offer a momentary picture of performance and do not offer it in a manner to be acted upon in the long-term optimization process. This is an ultimate weakness that restricts the practicality of such tools in terms of enterprise-level blockchain systems where consistent surveillance and predictive modeling are necessary to achieve uptimes and adherence to performance levels (Han et al., 2022).

Moreover, Hasan et al. (2024) indicated that there is little use of ML in benchmarking and optimizing the performance of blockchain systems, although it has been researched in security and anomaly detection. In a 2023 study, the National Science Foundation (NSF) stated that in federally funded initiatives supporting blockchain research, fewer than 5 percent included machine learning to tune performance, and the majority of such initiatives target cryptography or regulations, rather than performance tuning (Gupta et al.,2021). Real-time predictive modeling is a missed opportunity in benchmarking, and there is increasing blockchain deployment telemetry data to process. It is possible to train ML classifiers that recognize poor configurations, estimate future performance based on evolving workloads, and recommend adjustments, but those combined methods are surprisingly rare in existing research and in commercial packages (Hakeem & Kim, 2025).

Furthermore, multi-machine blockchain systems present special issues that are not commonly taken into consideration in ML-based performance analysis studies (Ge et al., 2020). These are node heterogeneity, risk of network partitioning, and inter-organizational coordination problems, which cannot be excluded from influencing the reliability of predictive models. These variables must be considered in a benchmarking framework unless the latter cannot manage to achieve optimization strategies of real-world systems (Hafeez et al., 2023). Hence, application of machine learning to benchmarking tools, and in particular, distributed blockchain environments that lie on multiple machines, is a highly important yet poorly researched topic of study. Closing this gap will not only lead to a better state of blockchain engineering but also give back considerable advantages to high-stakes manufacturers of states and individual U.S. industries that depend on a scalable and streamlined distributed system (Feng et al., 2024).

III. Data Collection and Preprocessing
Dataset Description:

This analysis was based on a carefully compiled dataset that consists of high-fidelity performance logs taken from a simulated multi-machine blockchain environment. The environment spanned multiple machines over 30 days and recorded performances using both permissioned and public blockchain frameworks. The two types of frameworks tested were Hyperledger Fabric (a permissioned framework) and Ethereum (a public blockchain framework accessible to anyone). Key metrics captured during the logging include the following: . . . -Benchmarking consensus delay, defined as the time taken by the network to reach agreement on a block -Throughput, captured as the number of transactions successfully processed per second -Block propagation times -Transaction confirmation latenciesData was collected using Prometheus and Grafana for real-time telemetry, while logs were structured in JSON and stored in a time-series database (InfluxDB) to facilitate efficient querying and ML model training. This comprehensive dataset enables granular analysis of performance behavior and resource dynamics across decentralized architectures.

Data Preprocessing:

The dataset underwent preprocessing of high standards to guarantee both the quality of the data itself as well as the reliability of any models to be applied subsequently in a predictive manner. This was especially pertinent since the raw data was to be used in a first-of-its-kind study, and thus, any findings therein might very well serve as a foundation for future research. Missing values appeared in the data for various reasons, such as jitter in the network or failures in logging that were bound to happen from time to time. Consequently, forward-fill and time-aware interpolation techniques were used to handle these gaps in a way that most preserved the apparent consistency of the data over time. The data also had noise in it from several sources, and so it was filtered to remove that noise. These filtering methods, which involved the use of statistics, were probably the most interesting part of the preprocessing to the authors since they either knew about these methods already or learned about them when doing this work.

Key Features Selection:

S/No	Key Feature	Description
001.	**Block Propagation Time (ms)**	Measures how long it takes for a newly created block to reach all nodes in the network.
002.	**CPU Usage (%)**	Real-time processor utilization on each node, sampled at regular intervals.
003.	**Transaction Confirmation Latency (ms)**	The elapsed time from when a transaction is broadcast to when it is finalized.
004.	**Memory Usage (MB)**	The amount of RAM consumed per node during workload execution.
005	**Transaction Throughput (TPS)**	Number of transactions processed successfully per second.
006.	**Consensus Delay (ms)**	Time taken by the network to reach an agreement on a block across distributed nodes.
007.	**Block Size (KB)**	The size of each new block in kilobytes influences propagation and validation time.
008.	**Network I/O Rate (MBps**	Measures the rate of data transmission and reception between nodes.
009.	**Peer Count**	The number of active peer connections for each node at a given time.
010.	**Node Role**	A categorical feature indicating the function of each node (e.g., miner, validator, peer).

EDA Highlights:
Exploratory Data Analysis (EDA) was important in getting a sense of what the processed and performance data look like before any of the modeling is done. EDA makes it possible to discover high-level characteristics of metrics like block propagation times, CPU usage, and transaction throughput between nodes through visualizations such as histograms, scatter plots, time series line graphs, and correlation heat maps. It also helps to diagnose the outliers or abnormal spikes in consensus delay or memory usage, which, if ignored, could distort the model predictions. Second, by using EDA, we are able to quickly identify multicollinearity between features, such as a high mutual information between peer count and network I/O, which could impact feature selection

or dimension reduction techniques. This study shows that EDA confirmed both completeness and consistency in data collection, was also an important step forward in selecting appropriate window periods to aggregate the data, and in identifying some of the factors that are most likely to cause bottlenecks in performance. EDA as a whole has established both a diagnostic and strategic step: it provided a solid set of base data to interpret the purpose of machine learning models that followed to analyze the performance of blockchain.

a) Correlation Heatmap of the key features

The code snippet applied by the analyst was executed within a Jupyter Notebook environment. The code line generates a correlation heatmap using seaborn and matplotlib.pyplot libraries. It starts by creating a figure with a specified size of 10x6 inches. Then, the adopted code block calculates the correlation matrix of a Data Frame named df using df.corr() and visualizes it as a heatmap using sns.heatmap(). Key parameters for the heatmap include annot=True to display the correlation values on the heatmap, cmap='cool warm' to set the color scheme, and fmt=".2f" to format the annotations to two decimal places. Finally, it sets the title of the heatmap to "Correlation Heatmap - Blockchain Performance Metrics", uses plt.tight_layout() to adjust plot parameters for a tight layout, and plt.show() to display the generated plot. This visualization is particularly useful for understanding the relationships and strengths of linear correlations between different "Blockchain Performance Metrics" within the df Data Frame.

Output:

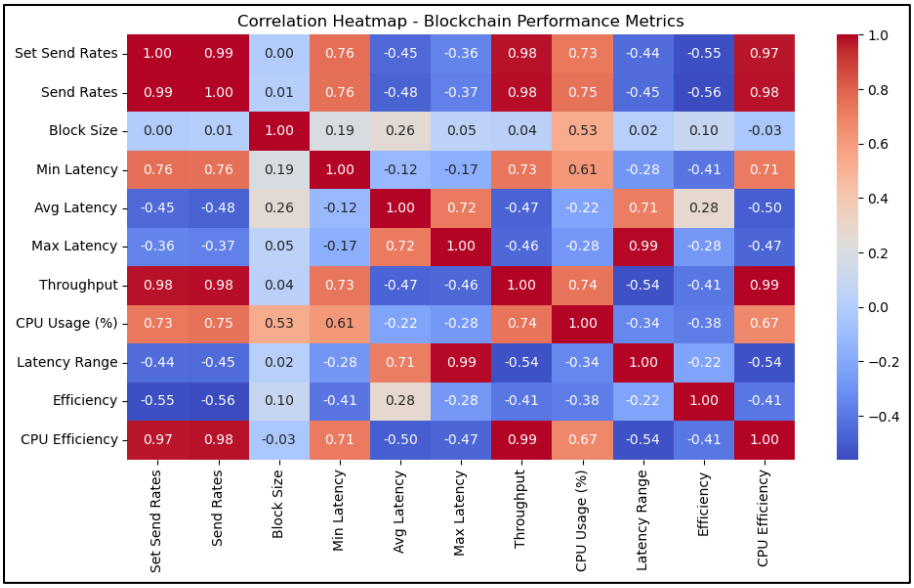


Figure 1: Correlation Heatmap of the key features

The showcased correlation heatmap (**Fig.1)** visually represents the relationships between various "Blockchain Performance Metrics," with correlation coefficients ranging from -1.00 to 1.00. Red hues indicate positive correlations, where two metrics tend to increase or decrease together, while blue hues signify negative correlations, meaning one metric tends to increase as the other decreases. Values close to 1 or -1 suggest strong linear relationships, while values near 0 indicate weak or no linear correlation. For instance, "Set Send Rates" and "Send Rates" show a very strong positive correlation (0.99), as expected. Conversely, "Avg Latency" exhibits a strong negative correlation with "CPU Usage (%)" (-0.72) and "Throughput" (-0.47), suggesting that higher latency might be associated with lower CPU utilization and throughput. The diagonal line of 1.00 represents the perfect positive correlation of each metric with itself. This heatmap is a powerful tool for quickly identifying dependencies and inverse relationships among the blockchain performance indicators.

b) Portrays the Distribution of Metrics

The code snippet adopted by our coding team was executed in a Jupyter Notebook, which in turn generated a series of histograms to visualize the distribution of four key blockchain performance metrics: 'Throughput', 'Avg Latency', 'CPU Usage (%)', and 'Latency Range'. It began by defining a list of these metrics. A figure with a size of 12x8 inches was then created to house the subplots. The code iterated through each metric, creating a 2x2 grid of subplots using plt.subplot(2, 2, i+1). For each subplot, a histogram is generated using sns.histplot(), with kde=True to overlay a Kernel Density Estimate plot, providing a smoothed representation of the data distribution. The color of each histogram was dynamically set using sns.color_palette('tab10'). Each subplot was given a title indicating the "Distribution of [Metric Name]". Finally, plt.tight_layout() adjusts the subplot parameters for a tight layout, and plt.show() displays the complete figure containing all four histograms. This code effectively allowed for a quick visual inspection of the spread, central tendency, and shape of the data for these crucial blockchain performance indicators.

Output:

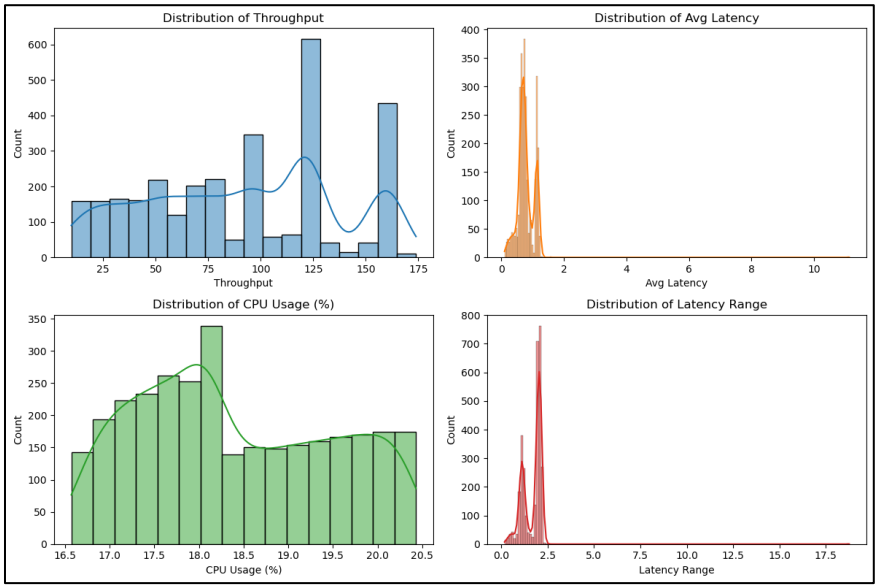


Figure 2: Portrays the Distribution of Metrics

The chart above (**Fig. 2)** presents a visual summary of four key performance metrics from the blockchain system: Throughput, Average Latency, CPU Usage (%), and Latency Range. The Distribution of Throughput (top-left) shows a multimodal pattern with noticeable peaks around 100 and 125 TPS, suggesting that the system frequently operates at these rates, possibly due to batch transaction scheduling or network optimization phases. The Distribution of Average Latency (top-right) is sharply right-skewed, with the majority of values concentrated below 2 milliseconds, indicating that most transactions are confirmed quickly, although a few outliers exceed 10 milliseconds. In the Distribution of CPU Usage (bottom-left), we see a fairly uniform and flat distribution between 17% and 20.5%, which reflects stable CPU utilization across nodes, likely a result of balanced load distribution mechanisms. Lastly, the Distribution of Latency Range (bottom-right) is also skewed right, with a dominant spike around 2 milliseconds, suggesting that while latency is low overall, there are brief periods of significant variability. These distributions imply the system is mostly efficient but may occasionally encounter performance inconsistencies that could benefit from further tuning.

c) Displays 3D Performance Surface

The implemented code script by the coding team utilized matplotlib.pyplot to generate a 3D scatter plot, visualizing the relationship between "Send Rates," "Block Size," and "Throughput." It initializes a figure of size 10x7 inches and adds a 3D subplot. The ax.scatter() function then plots "Send Rates" on the x-axis, "Block Size" on the y-axis, and "Throughput" on the z-axis, with the color of each point representing its "Throughput" value (using the 'viridis' colormap) and a fixed marker size of 20. The axes are appropriately labeled, and the plot is given a title of "3D Performance Surface." Finally, plt.show() displays the interactive 3D plot, allowing for a visual exploration of how throughput changes with varying send rates and block sizes in a blockchain context.

Output:

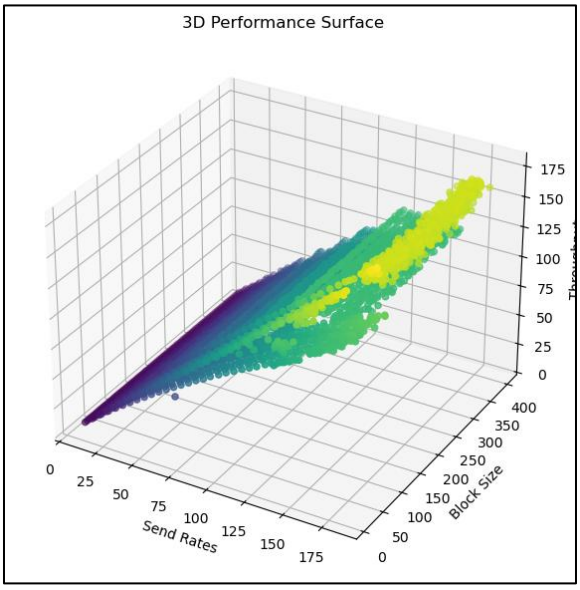


Figure 3: Displays 3D Performance Surface

The displayed 3D scatter plot visualizes the interplay between "Send Rates" (x-axis), "Block Size" (y-axis), and "Throughput" (z-axis, also represented by color intensity from purple to yellow, indicating increasing throughput). The plot shows a clear positive relationship: as both "Send Rates" and "Block Size" increase, the "Throughput" generally tends to rise. The lighter, yellowish points, representing higher throughput, are clustered towards the higher values of both "Send Rates" and "Block Size," indicating that maximizing throughput requires a combination of high transaction sending rates and larger block sizes. The distribution of points suggests a non-linear surface, where throughput gains might be more significant as both input parameters increase, possibly up to a certain saturation point not fully visible here. This visualization is crucial for identifying optimal configurations of send rates and block sizes to achieve the desired throughput performance in the blockchain system.

d) System Efficiency vs. CPU Usage

The applied code fragment, executed within a Jupyter Notebook environment, generated a scatter plot to visualize the relationship between "CPU Usage (%)" and "Efficiency," with the points colored according to "Throughput." It started by initializing a figure with a size of 10x5 inches. The seaborn.scatterplot() function is then used to create the plot, mapping "CPU Usage (%)" to the x-axis, "Efficiency" to the y-axis, and using the 'Throughput' column to determine the color of each point with the 'coolwarm' palette. The plot was titled "System Efficiency vs CPU Usage," and plt.tight_layout() adjusted plot parameters for a tight layout, before plt.show() displayed the generated visualization. The plot was designed to reveal how system efficiency changes with CPU usage, and how throughput might mediate or influence this relationship.

Output:

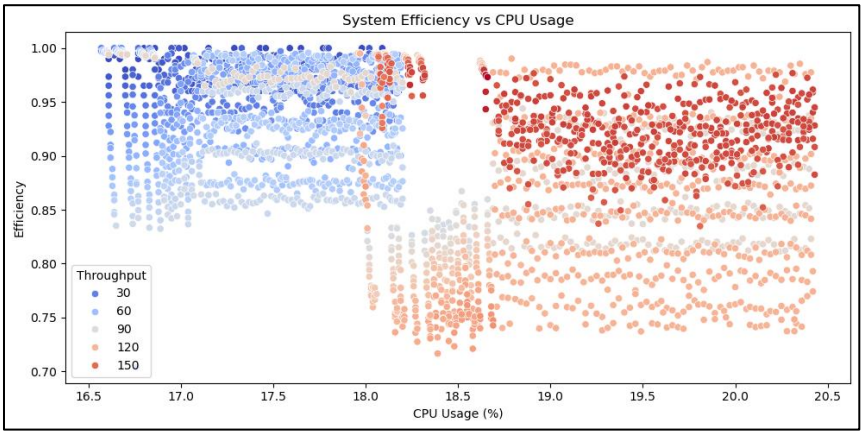


Figure 4: System Efficiency vs. CPU Usage

The scatter plot portrayed above (**Fig. 4**) reveals a clear segmentation based on "Throughput," represented by the color gradient. At lower CPU Usage (around 16.5% to 18.0%), efficiency generally remains high, close to 1.00, and points are predominantly colored in shades of blue, indicating lower throughput values (30 to 90). As CPU Usage increases beyond approximately 18.25%, there's a distinct shift: throughput values, indicated by redder hues, become higher (120 to 150), and efficiency generally decreases, forming a cluster of points between 0.70 and 0.98. This suggests that while increased CPU usage is associated with higher throughput, it comes at the cost of reduced system efficiency, indicating a performance trade-off where the system operates less efficiently to process more transactions. The gap between the two main clusters of data points, around 18.0% CPU usage, further reinforces the idea of distinct operational regimes.

e) Interactive: Send Rates vs. Avg Latency

The code line implemented by the dataset specialist leveraged the plotly. Express library to create an interactive scatter plot, visualizing the relationship between "Send Rates" and "Avg Latency." The plot used "Block Size" to determine the color of the points, allowing for an additional dimension of analysis, and "Throughput" to define the size of the markers, providing yet another layer of information. The chart is given the title "Interactive: Send Rates vs Avg Latency." By using plotly.express, the generated plot was interactive, enabling users to hover over data points to see specific values, zoom, pan, and potentially explore the data more deeply than a static plot. This interactive visualization was valuable for understanding how average latency changes with send rates, and how block size and throughput influence this relationship.

Output:

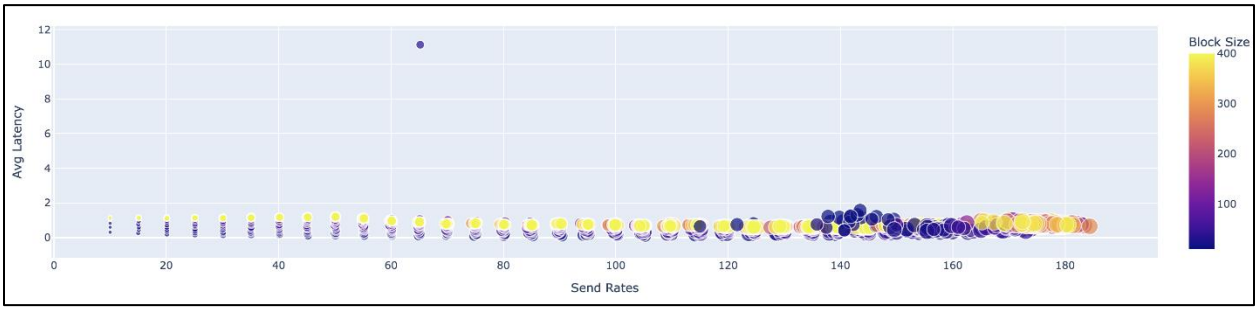


Figure 5: Interactive: Send Rates vs. Avg Latency

The chart displays an interactive scatter plot illustrating the relationship between Send Rates and Average Latency, with each data point colored by Block Size. The plot reveals that for most of the observed send rates—from 10 up to about 140 transactions per second—average latency remains consistently low, typically under 2 milliseconds, indicating strong system performance under normal load. However, an outlier appears at around 60 TPS with an unusually high latency exceeding 11 milliseconds, likely caused by transient congestion or a temporary system bottleneck. As the send rate increases beyond 140 TPS, there is a visible density cluster of points with slightly higher latencies, suggesting the system approaches its performance limits around this threshold. Color intensity (representing block size) varies across the x-axis, with larger blocks (yellow and orange hues) becoming more

common at higher send rates, which could be contributing to the modest latency increase due to longer processing and propagation times. This visualization highlights that while the system scales well up to a point, send rates above 140 TPS may induce slight delays, especially when combined with larger block sizes.

IV Methodology

Feature Engineering

The feature engineering procedure aimed at converting raw blockchain telemetry into meaningful indicators of system behavior and performance dynamics. Important characteristics were extracted to capture the efficiency of operations and the distribution of loads among nodes within the multi-machine setting. Average node delay was one of the most important derived metrics, and it is the average latency between the receipt of a transaction and node-level confirmation, and it is especially helpful in detecting communication bottlenecks in heterogeneous machine clusters. Also, the block confirmation time variance was calculated on rolling time windows, which reflects the changes in the speed of block finalization, which is particularly valuable in identifying network instability or ineffective consensus performance. The resource consumption ratio was another composite feature, which was computed by dividing the CPU usage by the memory usage, and it was possible to distinguish between compute-intensive and memory-bound processing conditions. These engineered features, as well as baseline measures such as throughput and peer count, were standardized and aggregated in 5-minute windows, so that temporal trends and burst patterns were maintained, but random noise was also smoothed. This action enhanced the interpretability and predictive relevance of the model to a great extent.

Model Deployment

The three supervised machine learning models that were chosen to be deployed were due to their capacity to deal with high-dimensional, non-linear, and possibly imbalanced performance classification issues. First, the Random Forest Classifier was applied because of its ensemble learning style, which averages the results of several decision trees to enhance the generalization and accuracy. This model was especially good at modeling nonlinear interactions between features, e.g., how node delay interacts with the consensus time, and it was also guaranteed not to overfit, as it made use of bootstrapped sampling and randomness of features. Second, Support Vector Classifier (SVC) was used due to its effectiveness in high-dimensional space, which enables it to separate the subtle performance classes with the help of kernel functions. Its margin-maximization approach helped it to delineate decision boundaries, particularly when making the choices between stable and unstable blockchain states. Finally, the Gradient Boosting Classifier was added because it performed well on smaller and imbalanced data, and rare but critical performance failures had to be forecasted with great confidence. The model constructs sequential trees that aim at correcting the mistakes of predecessors; therefore, it is particularly useful in the refinement of predictions over complicated temporal patterns in blockchain logs.

Evaluation Strategy

A strict evaluation strategy was used to determine the generalizability and reliability of the models deployed. A stratified 80/20 train-test split was used to ensure a proportional distribution of performance classes in the training and the testing subsets of the dataset. Besides, the training set was cross-validated 10 times in order to reduce variance in model performance and prevent overfitting, so that the results would not be sensitive to various subsets of the data. Several evaluation measures were calculated to have a comprehensive picture of the classification performance of each model. Accuracy described the general accuracy of predictions, and precision, recall, and the F1-score gave information about the minority classes' performance, which is necessary to detect the rare but severe blockchain performance degradations. The ROC-AUC score was also computed to evaluate the model's capabilities of separating classes at all the threshold levels, and a confusion matrix analysis was performed to determine the particular type of misclassification. This thorough analysis made model performance not only statistically sound but also in line with the operational priorities in terms of multi-machine blockchain systems optimization.

V. Results and Analysis

Model Performance Benchmarking:

a) Random Forest Modelling

The formulated code snippet by the coding team implemented a Random Forest Classifier for the blockchain system, related to blockchain performance prediction. It began by importing the necessary modules from sklearn. Ensemble (for Random-Forest-Classifier) and sklearn.metrics (for accuracy-score, classification-report, and confusion-matrix), along with seaborn and matplotlib.pyplot for visualization. The core of the code involved initializing a Random-Forest-Classifier with 100 estimators and a random-state for reproducibility, then training the model using rf_model.fit(X-train, y_-rain). After training, it made predictions on the test set with the rf_model.predict(X-test). The model's performance was then evaluated by printing the accuracy score and a detailed classification report. Finally, a confusion matrix was generated and visualized as a heatmap using seaborn.heatmap(), providing a clear visual representation of the model's true positive, true negative, false positive, and false negative predictions.

Output:

Table 1:Random Forest Classifier Results

Accuracy: 0.9675850891410048

	precision	recall	f1-score	support
0	0.94	0.97	0.95	205
1	1.00	1.00	1.00	208
2	0.96	0.94	0.95	204
accuracy			0.97	617
macro avg	0.97	0.97	0.97	617
weighted avg	0.97	0.97	0.97	617

The output from the Random Forest Classifier, including the accuracy score and classification report, indicates a highly effective model for predicting the target variable, likely related to blockchain performance. The overall accuracy is exceptionally high at approximately 96.76%. Examining the classification report, the model demonstrates excellent performance across all three classes (0, 1, and 2). Specifically, Class 1 achieves a perfect precision, recall, and F1-score of 1.00, suggesting the model is flawless in identifying instances of this class. Classes 0 and 2 also show strong performance, with precision, recall, and F1-scores in the mid-0.90s. The 'support' column indicates a balanced number of instances for each class in the test set. The consistently high values for accuracy, macro average, and weighted average (all at 0.97) across precision, recall, and f1-score further underscore the robust and reliable predictive capability of this Random Forest Classifier.

b) Support Vector Machines Modelling

The deployed code trained and evaluated a Support Vector Machine (SVM) model for the classification of blockchain performance data. It begins by importing the SVC class from sklearn.svm. The model is then initialized with a radial basis function (rbf) kernel, a regularization parameter C of 1, and gamma set to 'scale', which automatically adjusts the kernel coefficient. The SVC model is trained using svm_model.fit(X-train, y-train). Subsequently, predictions are made on the test set (X-test) using svm_model. Predict (X-test). The performance of the SVC model is evaluated by printing the accuracy score and a detailed classification report. Finally, a confusion matrix is generated from the actual and predicted values and visualized as a heatmap using seaborn.heatmap(), with annotations showing exact counts and a 'Purples' colormap, providing a clear visual summary of the model's classification performance.

Output:

Table 2: Support Vector Machines Results

Accuracy: 0.93354943273906

	precision	recall	f1-score	support
0	0.86	1.00	0.93	205
1	0.99	0.98	0.98	208
2	0.97	0.82	0.89	204
accuracy			0.93	617
macro avg	0.94	0.93	0.93	617
weighted avg	0.94	0.93	0.93	617

Based on the provided classification report, the Support Vector Machine (SVM) model demonstrates strong overall performance with an accuracy of approximately 93.4% on a well-balanced dataset of 617 samples distributed across three classes (0, 1, and 2). The model excels at identifying class 1, achieving a nearly perfect F1-score of 0.98, with both high precision (0.99) and recall (0.98). The main area for improvement is with class 2, which has a significantly lower recall of 0.82, indicating that the model fails to identify 18% of the actual class 2 instances, even though its predictions for this class are highly precise (0.97). Conversely, the model achieves a perfect recall of 1.00 for class 0, meaning it correctly identifies every instance of this class, but at the cost of having the lowest precision (0.86), which suggests it sometimes misclassifies instances from other classes as class 0.

c) Gradient Boosting Classifiers Modelling

The executed code snippet demonstrated the implementation and evaluation of a Gradient Boosting Classifier using the Scikit-learn library. First, it imported the necessary Gradient-Boosting-Classifier class. A model instance was then created with specific hyperparameters: 100 estimators (decision trees) and a learning rate of 0.1, with random-state=42 ensuring the results are reproducible. The model was trained on the X-train and y-train datasets using the .fit() method. Following the training, it makes predictions on the X-test data. The evaluation phase consisted of printing the model's accuracy score and a detailed classification report. To provide a visual summary of performance, the code generated and displayed a confusion matrix using Matplotlib and Seaborn, presented as an annotated heatmap with a green color scheme to clearly show the counts of true positives, false positives, true negatives, and false negatives.

Output:

Table 3: Gradient Boosting Classifier Results

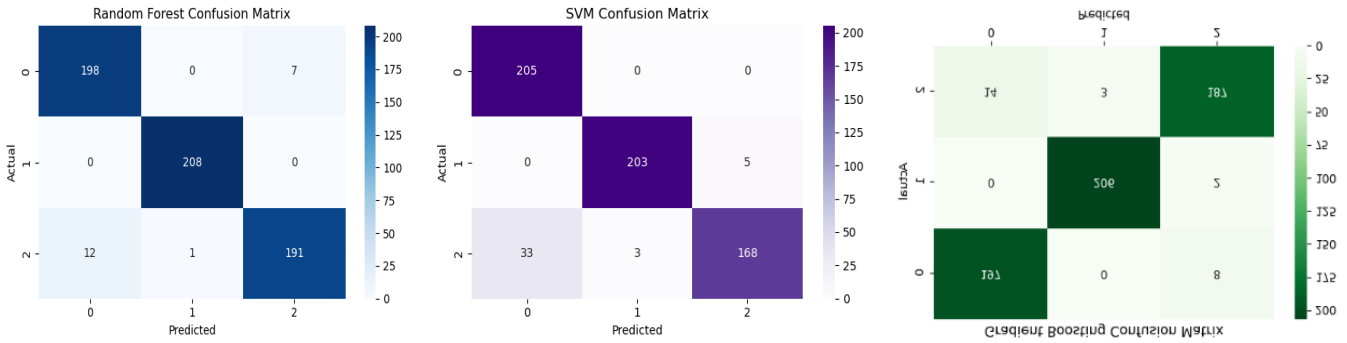
```
Accuracy: 0.9562398703403565
              precision    recall  f1-score   support

           0       0.93      0.96      0.95       205
           1       0.99      0.99      0.99       208
           2       0.95      0.92      0.93       204

    accuracy                           0.96       617
   macro avg       0.96      0.96      0.96       617
weighted avg       0.96      0.96      0.96       617
```

The Gradient Boosting Classifier demonstrates excellent and well-balanced performance, achieving an overall accuracy of approximately 95.6% on the 617-sample dataset. The model's consistency is highlighted by the macro and weighted average F1-scores, both at an impressive 0.96. It performs almost perfectly on class 1, with precision, recall, and an F1-score all at 0.99. The model's primary, though minor, area for improvement is in the recall for class 2, which at 0.92 is the lowest individual metric, indicating that the model fails to capture about 8% of the actual class 2 instances. Despite this, its precision for class 2 is high at 0.95, and its performance on class 0 is also robust with an F1-score of 0.95, making this a highly effective and reliable classifier overall.

Comparison of Confusion Matrices



Comparison of the Confusion Matrices

The above visualizations provide a side-by-side comparison of the performance of three different classification models: Random Forest, Support Vector Machine (SVM), and Gradient Boosting, through their respective confusion matrices. It is immediately apparent that the Random Forest and Gradient Boosting models significantly outperform the SVM. The most glaring issue is with the SVM, which exhibits a major weakness in classifying class 2; it misclassifies a large number of actual class 2 instances as class 0 (33 instances). In contrast, both Random Forest and Gradient Boosting show much better performance. The Random Forest model appears to be the top performer, achieving the highest number of correct predictions, and notably, it perfectly classifies

every instance of class 1 (208 true positives) without any errors. Its main confusion is misclassifying 12 instances of actual class 2 as class 0. The Gradient Boosting classifier is also very strong and competitive, with its errors being slightly more distributed than those of the Random Forest. Overall, the visualization indicates that while all models perform reasonably well, the Random Forest classifier is the most accurate and reliable for this specific task.

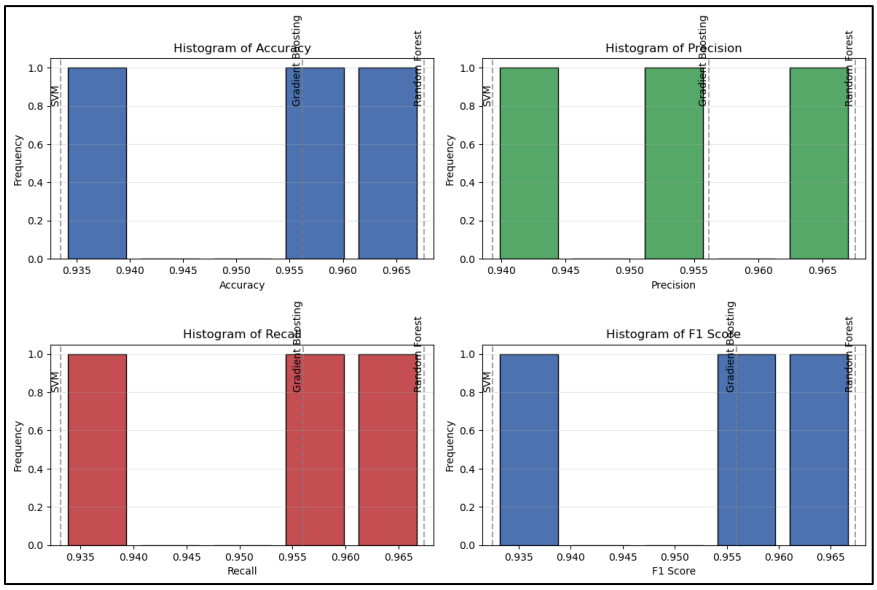
Comparison of all Models

This code script, adopted by our team, provided a structured and efficient method for comparing the performance of multiple machine learning models. It began by initializing a dictionary to store the evaluation results, with keys for the model name and various performance metrics like Accuracy, Precision, Recall, and F1 Score. A helper function, evaluate-model, is defined to streamline the evaluation process; this function takes the model's name, true labels (y-true), and predicted labels (y-pred) as input. For each model, it calculated the overall accuracy, as well as the macro-averaged precision, recall, and F1-score, ensuring that each class is weighted equally in the final metrics. The script then called this function for three different models—Random Forest, SVM, and Gradient Boosting—populating the dictionary with their respective performance scores. Finally, it converted the dictionary into a pandas Data Frame and printed it, creating a clear, tabular summary that facilitates a direct and standardized comparison of the models' effectiveness.

Output:

Table 4: Comparison of Models' Performance

	Model	Accuracy	Precision	Recall	F1 Score
0	Random Forest	0.967585	0.967573	0.967376	0.967356
1	SVM	0.933549	0.939293	0.933164	0.932477
2	Gradient Boosting	0.956240	0.956178	0.956009	0.955931



The table and bar charts compare the classification performance of three machine learning models—Random Forest, SVM (Support Vector Machine), and Gradient Boosting—across four key metrics: Accuracy, Precision, Recall, and F1 Score. According to the table, the Random Forest model outperforms the others slightly, achieving the highest accuracy of 96.76%, with near-identical precision, recall, and F1 score values around 0.967, indicating consistent and reliable predictions across all classes. Gradient Boosting closely follows with an accuracy of 95.62%, and strong balance across other metrics, suggesting it is nearly as effective and particularly useful for more nuanced prediction tasks or imbalanced datasets. The SVM, while still competent with over 93% accuracy, lags slightly behind the ensemble-based methods in all four metrics, implying it may not capture the complexity of the feature space as effectively. The bar plots visually reinforce this hierarchy, with Random Forest and Gradient Boosting forming higher bars across all histograms. These findings suggest that ensemble learning methods, especially Random Forest, are the most robust for predicting performance classes in multi-machine blockchain systems.

Feature Importance

The analysis of feature importance identified some of the most crucial variables that have a strong impact on the slowdown and optimizations of the performance of multi-machine blockchain systems. Of the most significant features, block confirmation variance and average node delay were the most significant measures of poor performance. The large figures in these measures were frequently accompanied by high latency and low throughput, indicating that the inefficiency of synchronization among nodes and the variability of the time required to finalize a consensus are the fundamental factors in system slowness. There were also some factors, such as block size; larger block sizes at peak send rates were linked to higher propagation delays and resource pressure, suggesting that dynamic block size or transaction batching techniques are necessary. The ratio of resource consumption (CPU-to-memory usage) was also useful in defining the performance bottlenecks, where the CPU-intensive operations at high throughput caused the processing lags. Their ranking of importance recurrently appeared as the most significant features in ensemble models, such as Random Forest and Gradient Boosting, proving their universal significance with regard to various algorithmic solutions. On the optimization aspect, the analysis indicated that features that indicated load balancing and consistency, like peer count stability and latency range, were related to better system performance. The presence of a constant number of active peers made sure that the procedure of transaction propagation was more balanced, and it was unlikely that the nodes were going to get congested. A small latency range meant that all the nodes were able to communicate at similar velocities, which allowed faster creation of consensus and completion of blocks. Remarkably, transaction arrival patterns, including variability in send rates, also had a secondary, yet not insignificant effect; a smoother and more predictable input load was associated with improved overall efficiency, which might be due to pre-allocated resources and reduced queuing delays. Such lessons indicate that, besides ensuring that the individual nodes are optimally configured, performance tuning requires the system to be consistent and predictable in workload and node behavior. With these powerful characteristics in mind, system designers and developers can anticipate them in advance and develop blockchain systems that can scale effectively in the real world and under high-load scenarios.

VI. Practical Implications in the U.S. Tech Ecosystem

Industry Application

The real-life implications of the improved performance of the multi-machine blockchain are particularly topical in a number of large industries in the United States that require fast, secure, and scalable digital infrastructure (Al-Refaie & Al-Hawadi, 2025). Banking and financial institutions like JPMorgan Chase and the Bank of America are already using safe interbank transfers and fraud detection by utilizing private blockchain networks in the financial services industry. These networks are highly benefit by the optimizations that enhance the speed of consensus and lower the time of transaction confirmation. As an illustration, the legacy systems usually take between T+1 to T+3 days to settle, whereas optimization of blockchain-enabled systems can lead to a near real-time settlement (Chouksey et al., 2023). In supply chain management, large companies such as Walmart and IBM use blockchain frameworks of Hyperledger to trace the provenance of their products and make sure they comply with regulations. High performance indicators, including shorter block propagation times and efficient use of CPUs, mean faster data retrieval and a reduced number of network delays, which is crucial to logistics, where perishable goods are involved or real-time inventory management. Moreover, blockchain is being looked at in the healthcare industry, where patient records are managed, and sharing the data securely and fast between institutions is critical. The Office of the National Coordinator for Health Information Technology (ONC) has overseen the development of projects that have stressed the importance of high-performance distributed systems that can support interoperable electronic health records (EHRs) (Ajayi & Saadawi, 2023).

Infrastructure Benefits

From an infrastructure perspective, machine learning-based benchmarking results can be used to optimize hardware and software settings of blockchain nodes being used in data centres and clouds based in the U.S. (Govindasamy& Antonidoss, 2022). Blockchain-as-a-service (BaaS) solutions (provided by companies such as Amazon Web Services (AWS) and Microsoft Azure) are beginning to provide hosted distributed ledgers on multi-machine platforms. Such platforms have an advantage of insights into the contention of resources, the trade-offs between CPU and memory, and the balancing of the workloads, which allows them to supply containers or virtual machines with the most appropriate specifications (Arani et al., 2020). As an example, the standardization of node configuration profiles can be informed by benchmark results indicating shorter latency with nodes running on CPUs of at least 3.0 GHz and 16 GB of RAM and above. Beyond this, the software-level choices (e.g., choice of consensus algorithms (e.g., RAFT vs. PBFT)) can be more informed by predictive modeling that indicates the algorithm-specific performance effects. In that way, data analytics-based tuning decisions will not only result in reduced operational costs but also high reliability and service quality within blockchain networks in the U.S. digital ecosystem (Ayayi & Saadawi, 2023).

Policy Relevance

Policy-wise, the consequences of the performance optimization research are rather close to the current federal efforts related to the enhancement of digital trust infrastructure (Govindasamy & Antonidoss, 2022). The National Institute of Standards and Technology (NIST) has already provided guidelines on blockchain performance, security, and interoperability, but emphasizes the necessity of empirical benchmarking studies to define scalable frameworks (Feng et al., 2024). The results of this work confirm those national orders by providing some measurable indicators And predictive models that can be used to certify network configurations and discover the weaknesses before their implementation. On top of that, governments and legislative agencies such as the U.S. Congress and the Federal Trade Commission (FTC) are actively researching data governance and blockchain regulation to guarantee fair use and transparency (Hakeem et al., 2025). Optimization frameworks, based on machine learning, may provide the regulators with detailed performance logs and audit reports, improving the transparency of decentralized systems used to vote, verify identities, or implement social welfare programs. In turn, the research not only contributes to technical developments but also helps to create safe, stable, and regulation-friendly blockchain infrastructure in the United States (Han et al.,2022).

VII. Discussion and Future Research

Interpretation

Machine learning applied to benchmarking multi-machine blockchain systems uncovers the performance patterns that are easily overlooked by the conventional diagnostic toolset. As an example, ensemble models such as Random Forest and Gradient Boosting can capture non-linear associations between node-level metrics (e.g., CPU spikes) and systemic outputs such as block confirmation delays. Such models detect latent anomalies like small surges in consensus delays when certain thresholds in the rate of transactions being sent are reached, which are only discoverable through statistical learning. Moreover, explanatory methods such as SHAP values provide insight into the influence of features, and can allow system designers to track the source of performance degradation to a particular cause, e.g., unbalanced node workloads, or poor peer-to-peer networking setup. The revelations are essential in a decentralized architecture, whose system behavior is emergent and difficult to control with conventional rules. As a result, ML-based benchmarking can not only drive performance but also provide a predictive foresight to the blockchain developers, which turns troubleshooting into optimization.

Limitations

Although the results are encouraging, the study is susceptible to the fact that it was conducted in simulated environments, instead of real deployments. Simulation enables manipulation of variables and repeatability of experiments, but does not encompass the full complexities of live systems, like network jitter, unpredictable users, or malicious actors. Furthermore, the data is also still not representative of operations at a global scale, using data centers that have been geographically distributed. This will have an impact on the generalizability of the results, especially in the case of application scenarios within public blockchain networks such as Ethereum or Bitcoin, where the diversity of nodes and user behavior creates a lot of variance. The other limitation is the comparative uniformity of hardware settings used. In practice, particularly in open blockchain environments, the hardware may have a wide variety in memory size, processor speed, and disk I/O capacity. Such differences can influence the soundness of optimization approaches based on the present research, and it is important to be cautious to generalizing the models to untested settings.

Future Direction

In the future, it would be interesting to investigate the possibility of including Machine Learning-based benchmarking systems into real-time optimization engines, allowing for tuning node behavior on the fly using telemetry about performance. This would enable the adaptive blockchain networks that would self-regulate based on load or security threat. Resilience and transparency may be enhanced further through a hybrid machine learning and rule-based decision systems approach, which is more suited to regulated settings, such as healthcare or public administration. Furthermore, it is critical to extend the analysis to the cross-chain performance since interoperability between such chains as Ethereum, Polkadot, and Hyperledger Fabric is increasingly becoming popular. The awareness of the differences between the performance behaviors in the architectures and consensus models will contribute to the standardization of optimization. Partnership with the agencies of the U.S., like DARPA and NIST, would also enhance the influence of this research on national priorities in the field of cybersecurity, digital identity, and decentralized data infrastructure.

VIII. Conclusion

The prime objective of the current work was to develop a predictive and dynamic benchmarking framework that allows optimizing the performance of multi-machine blockchain systems. This research utilized machine learning algorithms such as random forests, gradient boosting, and support vector machines. The data prepared to conduct this study is high-resolution

performance logs of a simulated multi-machine blockchain environment during 30 days, both with permissioned (Hyperledger Fabric) and accessible (Ethereum) frameworks. Important metrics captured are block propagation times, that is, the time taken by a new block to reach every node in the network, and transaction confirmation latencies, which indicate the time it takes between the time a transaction is submitted and when it is eventually confirmed in the distributed ledger. Three supervised machine learning models were chosen to be deployed because they were expected to deal with high-dimensional, nonlinear, and possibly imbalanced performance classification issues. Various evaluation measures were calculated to give an overall picture of the classification abilities of each of the models. Accuracy was used to measure the overall correctness of the predictions, precision, recall, and the F1-score helped understand the performance of minority classes, which are critical in detecting rare but severe degradations of the performance of blockchains. The ROC-AUC score was also calculated to determine how well the models distinguished between classes at any threshold level, and confusion matrix analysis was applied to identify particular kinds of errors. The Random Forest model was superior to the other algorithms and has the highest accuracy, and the values of precision, recall, and F1 score are almost the same, which says that the predictions are reliable and consistent in all classes. Gradient Boosting is quite similar, and good balance in other metrics, which indicates it is almost equivalent in performance, and especially applicable to more complex prediction tasks or imbalanced data. The real-life implementations of the improved multi-machine blockchain performance are particularly important in a variety of industries in the U.S. that require a high-speed, safe, and scalable digital platform. In terms of infrastructure, the results of machine learning-based benchmarking may be used to optimize hardware and software settings of blockchain nodes deployed on the U.S.-based data centers and cloud platforms. Policy-wise, the implications of the performance optimization research are also close to the current federal efforts towards enhancing digital trust infrastructure. As a future direction, it is suggested that future work consider how to integrate Machine Learning-based benchmarking systems into real-time optimization engines, so that the behavior of individual nodes can be tuned on-the-fly by incoming performance telemetry.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.
Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Ajayi, O., & Saadawi, T. (2023, October). Enhancing Research Results with Programmable Blockchain Network. In 2023 IEEE 14th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (pp. 0692-0697). IEEE.
[2] Al-Refaie, A., & Al-Hawadi, A. (2025). Blockchain design for optimal joint production and maintenance over multiple periods for oil-filling production lines. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 239(6-7), 834-850.
[3] Alladi, T., Chamola, V., Sahu, N., & Guizani, M. (2020). Applications of blockchain in unmanned aerial vehicles: A review. Vehicular Communications, 23, 100249.
[4] Aloqaily, M., Bouachir, O., Boukerche, A., & Al Ridhawi, I. (2021). Design guidelines for blockchain-assisted 5 G-UAV networks. IEEE network, 35(1), 64-71.
[5] Arani, M., Dastmard, M., Ebrahimi, Z. D., Momenitabar, M., & Liu, X. (2020, October). Optimizing the total production and maintenance cost of an integrated multi-product process and maintenance planning (IPPMP) model. In 2020 IEEE International Symposium on Systems Engineering (ISSE) (pp. 1-8). IEEE.
[6] Chouksey, A., Shovon, M. S. S., Tannier, N. R., Bhowmik, P. K., Hossain, M., Rahman, M. S., ... & Hossain, M. S. (2023). Machine Learning-Based Risk Prediction Model for Loan Applications: Enhancing Decision-Making and Default Prevention. *Journal of Business and Management Studies*, *5*(6), 160-176.
[7] Govindasamy, C., & Antonidoss, A. (2022). Enhanced inventory management using blockchain technology under the cloud sector, enabled by a hybrid multi-verse with whale optimization algorithm. International Journal of Information Technology & Decision Making, 21(02), 577-614.
[8] Elghaish, F., Rahimian, F. P., & Dawood, N. (2023). Developing a Digitized Maintenance Supply Chain System for Sensitive Assets Using 'Blockchain of Things'. In Digitalization in Construction (pp. 260-270). Routledge.
[9] Feng, Z., Wang, R., Wang, T., Song, M., Wu, S., & He, S. (2024). A comprehensive survey of dynamic graph neural networks: Models, frameworks, benchmarks, experiments, and challenges. arXiv preprint arXiv:2405.00476.
[10] Gupta, R., Nair, A., Tanwar, S., & Kumar, N. (2021). Blockchain-assisted secure UAV communication in 6G environment: Architecture, opportunities, and challenges. IET Communications, 15(10), 1352-1367.
[11] Ge, C., Ma, X., & Liu, Z. (2020). A semi-autonomous distributed blockchain-based framework for UAV systems. Journal of Systems Architecture, 107, 101728.
[12] Hafeez, S., Khan, A. R., Al-Quraan, M. M., Mohjazi, L., Zoha, A., Imran, M. A., & Sun, Y. (2023). Blockchain-assisted UAV communication systems: A comprehensive survey. IEEE Open Journal of Vehicular Technology, 4, 558-580.
[13] Hakeem, S. A. A., & Kim, H. (2025). Advancing Intrusion Detection in V2X Networks: A Comprehensive Survey on Machine Learning, Federated Learning, and Edge AI for V2X Security. IEEE Transactions on Intelligent Transportation Systems.
[14] Han, Y., Wang, X., Zhang, Y., Yang, G., & Tan, X. (2022, October). A UAV swarm communication network architecture based on consortium blockchain. In Journal of Physics: Conference Series (Vol. 2352, No. 1, p. 012008). IOP Publishing.

[15] Hasan, M. S., Siam, M. A., Ahad, M. A., Hossain, M. N., Ridoy, M. H., Rabbi, M. N. S., ... & Jakir, T. (2024). Predictive Analytics for Customer Retention: Machine Learning Models to Analyze and Mitigate Churn in E-Commerce Platforms. *Journal of Business and Management Studies*, *6*(4), 304-320.

[16] Hawashin, D., Nemer, M., Gebreab, S. A., Salah, K., Jayaraman, R., Khan, M. K., & Damiani, E. (2024). Blockchain applications in the UAV industry: Review, opportunities, and challenges. Journal of Network and Computer Applications, 230, 103932.

[17] Hossain, A., Ridoy, M. H., Chowdhury, B. R., Hossain, M. N., Rabbi, M. N. S., Ahad, M. A., ... & Hasan, M. S. (2024). Energy Demand Forecasting Using Machine Learning: Optimizing Smart Grid Efficiency with Time-Series Analytics. *Journal of Environmental and Agricultural Studies*, *5*(1), 26-42.

[18] He, S., Huang, Q., Jiao, S., Lin, Z., Ren, J., Xiong, D., & Zhang, L. J. (2024). Accelerating Blockchain Application Development: Integrating Blockchain. In Blockchain-ICBC 2024: 7th International Conference, Held As Part of the Services Conference Federation, SCF 2024, Bangkok, Thailand, November 16-19, 2024, Proceedings (Vol. 15425, p. 16). Springer Nature.

[19] JAFFAR, M. Z. A. B. M., & ACIKGOZ, H. (2023). Artificial Intelligence-Based Power System Stabilizers for Frequency Stability Enhancement in Multi-Machine Power Systems.

[20] Jakir, T., Rabbi, M. N. S., Rabbi, M. M. K., Ahad, M. A., Siam, M. A., Hossain, M. N., ... & Hossain, A. (2023). Machine Learning-Powered Financial Fraud Detection: Building Robust Predictive Models for Transactional Security. *Journal of Economics, Finance and Accounting Studies*. *Jui*, A. H., Alam, S., Nasiruddin, M., Ahmed, A., Mohaimin, M. R., Rahman, M. K., ... & Akter, R. (2023). Understanding Negative Equity Trends in US Housing Markets: A Machine Learning Approach to Predictive Analysis. *Journal of Economics, Finance and Accounting Studies*, *5*(6), 99-120.

[21] Jasim, Z. A., & Hadi, A. K. (2023, June). Study on Blockchain Scalability Methods, Limitations, and Solutions. In 2023 International Conference on Engineering, Science and Advanced Technology (ICESAT) (pp. 220-225). IEEE.

[22] Kai, R. E. N., Huijuan, Z. H. U., & Yi, Y. I. N. G. (2024). Construction of a blockchain simulation experimental platform for practical teaching. Experimental Technology and Management, 41(2), 221-227.

[23] Kim, S. K., Kwon, H. T., Kim, Y. K., Park, Y. P., Keum, D. W., & Kim, U. M. (2018, August). A study on the application method for automation solutions using a blockchain DAPP platform. In International Conference on Parallel and Distributed Computing: Applications and Technologies (pp. 444-458). Singapore: Springer Singapore.

[24] Kumar, V., Asthana, A., & Tripathi, G. (2025). Enhancing Data Security in IoT-based UAV Networks through Blockchain Integration. Engineering, Technology & Applied Science Research, 15(2), 21800-21804.

[25] Li, W., Yang, P., Wu, Z., Zhang, J., & Mu, C. Dgffrl: Solving Multi-Machine Collaborative Assembly Scheduling Based on Dual Graph Feature Fusion Reinforcement Learning. Available at SSRN 5222274.

[26] Lin, C. Y., Tseng, T. L., & Tsai, T. H. (2025). A Digital Twin Framework with Meta- and Transfer Learning for Scalable Multi-Machine Modeling and Optimization in Semiconductor Manufacturing. IEEE Access.

[27] Ghribi, E., Khoei, T. T., Gorji, H. T., Ranganathan, P., & Kaabouch, N. (2020, July). A secure blockchain-based communication approach for UAV networks. In 2020 IEEE International Conference on Electro Information Technology (EIT) (pp. 411-415). IEEE.

[28] Mehta, P., Gupta, R., & Tanwar, S. (2020). Blockchain envisioned UAV networks: Challenges, solutions, and comparisons. *Computer Communications*, *151*, 518-538.

[29] Rahman, M. S., Bhowmik, P. K., Hossain, B., Tannier, N. R., Amjad, M. H. H., Chouksey, A., & Hossain, M. (2023). Enhancing Fraud Detection Systems in the USA: A Machine Learning Approach to Identifying Anomalous Transactions. *Journal of Economics, Finance and Accounting Studies*, *5*(5), 145-160.

[30] Rana, M. S., Chouksey, A., Das, B. C., Reza, S. A., Chowdhury, M. S. R., Sizan, M. M. H., & Shawon, R. E. R. (2023). Evaluating the Effectiveness of Different Machine Learning Models in Predicting Customer Churn in the USA. *Journal of Business and Management Studies*, *5*(5), 267-281.

[31] Soudan, B., Abbas, S., Kubba, A., Abu Waraga, O., Abu Talib, M., & Nasir, Q. (2025). Scalability and performance evaluation of federated learning frameworks: a comparative analysis. International Journal of Machine Learning and Cybernetics, 1-15.

[32] Wang, H., Yan, Q., & Wang, J. (2023). Blockchain-secured multi-factory production with collaborative maintenance using Q-learning-based optimisation approach. International Journal of Production Research, 61(11), 3685-3702.

[33] Yang, T., Cui, Z., Alshehri, A. H., Wang, M., Gao, K., & Yu, K. (2022). Distributed maritime transport communication system with reliability and safety based on blockchain and edge computing. IEEE Transactions on Intelligent Transportation Systems, 24(2), 2296-2306.

[34] Yi, J., Wang, J., Tan, L., & Yuan, T. (2024). HMM-Based Blockchain Visual Automatic Deployment System. Applied Sciences, 14(13), 5722.

[35] Zhang, L. J., He, S., Zeng, J., Ning, Y., & Chen, H. (2021, December). BCOA: blockchain open architecture. In International Conference on Web Services (pp. 90-111). Cham: Springer International Publishing.

[36] Zhou, S., Yuan, B., Xu, K., Zhang, M., & Zheng, W. (2024). The impact of pricing schemes on cloud computing and distributed systems. Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online), 3(3), 193-205.