Predicting Throughput and Latency in Hyperledger Fabric Blockchains Using Random Forest Regression

Deshinta Arrova Dewi[1,*], Tri Basuki Kurniawan[2]

[1]Faculty of Data Science and Information Technology, INTI International University, Malaysia

[2]Faculty of Science Technology, Universitas Bina Darma, Palembang, Indonesia

ABSTRACT

The study focuses on enhancing the performance optimization of Hyperledger Fabric blockchains through predictive modeling using Random Forest regression. It emphasizes the importance of accurately predicting two critical performance metrics—throughput (measured in transactions per second or TPS) and latency (defined as the time taken to confirm transactions). These metrics directly influence the efficiency and user experience of blockchain applications, making their accurate prediction essential for configuring blockchain networks effectively. The research leverages data collected through Hyperledger Caliper, a benchmarking tool, which provides detailed measurements of various configuration parameters, including block size, transaction arrival rate, and the number of orderer nodes. Through rigorous exploratory data analysis, the study identifies how these parameters impact throughput and latency, revealing complex interdependencies that challenge traditional optimization approaches. Using Random Forest regression, a robust ensemble learning method, the study demonstrates that the predictive model can achieve high accuracy. The performance of the model is assessed using metrics such as R-squared values, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE), which collectively underscore its ability to offer reliable predictions across varying configurations. The results of this research provide practical insights for blockchain administrators, allowing them to configure Hyperledger Fabric settings more efficiently, thereby reducing the trial-and-error process typically involved in performance tuning. Moreover, the study's findings contribute to the broader field of blockchain performance optimization by offering a data-driven framework that bridges theoretical analysis with practical application in real-world scenarios. Looking forward, the study suggests avenues for future research, including expanding the dataset to cover more diverse blockchain platforms and configurations, incorporating real-world deployment data for validation, and exploring additional machine learning algorithms for even greater predictive accuracy. This approach highlights the critical role of data-driven methodologies in optimizing blockchain network performance and encourages further collaboration and exploration in the domain.

Keywords Hyperledger Fabric optimization, Random Forest regression, blockchain throughput, latency prediction, data-driven performance tuning.

Introduction

Blockchain technology has emerged as a transformative force across various sectors, distinguished by its fundamental properties of decentralization, immutability, and consensus mechanisms. Decentralization enables a distributed network where no single entity controls the entire system, eliminating the need for intermediaries and enhancing security through technologies like digital signatures and cryptographic hashes [1]. Immutability ensures that once data is recorded on the blockchain, it cannot be altered or deleted without

Submitted 29 December 2024
Accepted 6 February 2025
Published 8 March 2025

Corresponding author
Deshinta Arrova Dewi,
deshinta.ad@newinti.edu.my

Additional Information and Declarations can be found on page 85

DOI: 10.47738/jcrb.v2i1.27

How to cite this article: D. A. Dewi, and T. B. Kurniawan, "Predicting Throughput and Latency in Hyperledger Fabric Blockchains Using Random Forest Regression," *J. Curr. Res. Blockchain,* vol. 2, no. 1, pp. 63-87, 2025.

network consensus, thereby maintaining the integrity of transactions and historical records [2], [3]. Consensus mechanisms are vital for achieving agreement among distributed nodes, ensuring that all participants validate transactions before they are added to the blockchain, which maintains the system's integrity and trustworthiness [4]. These characteristics collectively create a secure and reliable framework for digital transactions, distinguishing blockchain from traditional centralized systems and addressing challenges such as vulnerability to cyberattacks and data manipulation [5], [6], [7].

Hyperledger Fabric stands out as a prominent permissioned blockchain platform tailored for enterprises and organizations that require secure, scalable, and efficient blockchain solutions. As an open-source project under the Linux Foundation, its modular architecture allows for customization and flexibility in deployment, making it ideal for applications where privacy and access control are paramount, such as in healthcare, supply chain management, and financial services [8], [9]. The permission nature of Hyperledger Fabric restricts network participation to authorized entities, enhancing security and trust through strict access control mechanisms [9], [10]. Additionally, its support for multiple consensus protocols and smart contract implementations facilitates the creation of tailored blockchain solutions that meet specific organizational needs [11], [12]. Performance metrics such as Throughput (Transactions Per Second - TPS) and Latency are critical in evaluating the efficiency of Hyperledger Fabric deployments. High throughput enables the processing of numerous transactions efficiently, while low latency ensures timely transaction confirmations, both of which are essential for applications requiring substantial transaction volumes and real-time data processing [13], [14]. Understanding and optimizing these metrics is crucial for leveraging Hyperledger Fabric's capabilities in enterprise settings, ensuring that blockchain implementations meet the desired performance standards [15], [16].

The performance metrics of throughput, measured in TPS, and latency play pivotal roles in shaping the effectiveness of blockchain applications, directly influencing user experience. In permissioned blockchain platforms such as Hyperledger Fabric, these metrics are particularly significant as they determine the efficiency and responsiveness of the system. Throughput reflects the capacity of a blockchain to process transactions within a given time frame, which is essential for applications demanding high transaction volumes, including those in financial services, supply chain management, and healthcare. High throughput can reduce wait times for transaction confirmations, thereby enhancing user satisfaction and fostering trust in the system. For instance, Gorenflo et al. have shown that Hyperledger Fabric is capable of achieving throughput rates of over 20,000 TPS under ideal conditions, making it well-suited for enterprise applications requiring rapid transaction processing [15], [17]. However, inadequate throughput can lead to transaction bottlenecks, causing delays that frustrate users and potentially obstruct the widespread adoption of blockchain solutions [18].

Latency, defined as the time it takes for a transaction to be confirmed and added to the blockchain after initiation, is equally critical. High latency can severely affect user experience, especially in applications where real-time data processing is essential, such as emergency response systems or e-voting platforms. In Hyperledger Fabric, latency is influenced by various factors, including network configuration and the complexity of transactions. For example, Ghosh and Dutta documented an average latency of 12.16 seconds in a Hyperledger Fabric-based application, a delay that may be unsuitable for many time-sensitive scenarios [13]. In contrast, lower latency provides users with prompt transaction feedback, which is vital for maintaining engagement and satisfaction. Piao highlights that high latency may detract from the performance of applications that require immediacy, which could undermine the user experience and limit the utility of the blockchain system [12].

The interplay between throughput and latency is complex, with trade-offs often arising between these two metrics. Optimizing throughput can sometimes lead to higher latency due to increased processing loads, as noted by Wang and Chu, who observed that as transaction volumes grow, the system may experience delayed transaction confirmations if it approaches its processing capacity [16]. Consequently, balancing throughput and latency is crucial for achieving optimal performance in blockchain applications. Agbo et al. emphasize that applications such as HealthChain, which utilizes Hyperledger Fabric for managing electronic medical records, depend on high throughput and low latency to maintain efficient record management while safeguarding data confidentiality and integrity [10]. Moreover, Kadhum's research on e-government applications underlines that even with the integration of additional organizations, Hyperledger Fabric can sustain acceptable performance levels, demonstrating scalability while effectively managing both throughput and latency demands [19].

Despite the increasing adoption of Hyperledger Fabric in enterprise settings, many users continue to rely on default configuration settings without fully understanding their impact on system performance. Default parameters, such as block size and transaction arrival rates, directly influence the blockchain's throughput and latency. However, these parameters are often set to default values, leading to suboptimal performance outcomes that may hinder the application's ability to meet operational requirements. Without a thorough understanding of how configuration choices affect throughput and latency, blockchain administrators may inadvertently create performance bottlenecks, resulting in slower transaction processing times and reduced system responsiveness. There is thus a critical need for empirical data to inform optimal configuration choices, allowing users to maximize Hyperledger Fabric's capabilities while ensuring efficient and reliable transaction processing.

Furthermore, optimizing blockchain performance presents unique challenges, particularly when balancing throughput and latency across different transaction arrival rates and network configurations. As noted in existing literature, achieving high throughput may lead to increased latency under certain network conditions. This trade-off poses difficulties for administrators seeking to improve both metrics simultaneously [16]. This balance becomes even more complex with variations in transaction arrival rates, where higher rates can strain the network and elevate latency. Consequently, an empirical approach to understanding these dynamics is necessary to provide clear guidelines on how best to configure Hyperledger Fabric's parameters to achieve a favorable balance between throughput and latency. Addressing these challenges can help to create a more responsive and adaptable blockchain environment, particularly for applications requiring high transaction volumes and real-time data processing.

Incorporating insights from multiple domains of data-driven analytics enriches the predictive modeling framework presented in this study. The work on [20] and [21] illustrates the potent application of machine learning techniques, including sentiment analysis, for deriving predictive insights in complex data environments. Such methodologies resonate with the foundational aims of our study in leveraging data-driven models to predict outcomes with high accuracy and relevance. Further emphasizing the utility of Random Forest and other ensemble methods, the studies [22] and [23] provide a robust comparative perspective on algorithmic effectiveness in predictive analytics. Their focus on refining predictions through iterative optimization and algorithmic comparison aligns closely with the methodological rigor adopted in this work. The application of predictive modeling to blockchain systems, as highlighted in [24] demonstrates how machine learning can improve resilience and stability in blockchain networks. Similarly, [25] showcases the relevance of integrating clustering and sentiment trends to better understand data behaviors in blockchain and digital asset contexts. Broadening the scope to emerging technological applications, [26] and [27] provide critical insights into data-driven decision-making and predictive modeling in digital and virtual spaces. Collectively, these works emphasize the breadth and adaptability of predictive algorithms, such as Random Forest regression, in addressing real-world challenges and enhancing predictive accuracy across various domains.

The primary goal of this study is to develop a predictive model using Random Forest Regression to estimate TPS and average latency based on various configuration parameters within Hyperledger Fabric. By leveraging machine learning techniques, this study aims to create a model that can accurately forecast performance metrics, providing valuable insights into how different settings influence overall system behavior. In addition to the primary objective, the study seeks to identify the configuration parameters that exert the most significant impact on throughput and latency. These parameters may include block size, number of orderer nodes, and transaction arrival rates, among others. Through this identification process, the study intends to offer blockchain administrators practical recommendations on configuring Hyperledger Fabric for optimal performance in diverse operational contexts.

The secondary objectives of the study focus on providing actionable guidelines for optimizing blockchain performance. This entails not only pinpointing influential parameters but also understanding their combined effects on TPS and latency. By doing so, the study aims to contribute to the development of performance optimization strategies that can be applied across various industries utilizing Hyperledger Fabric. These guidelines will assist administrators in making informed decisions when configuring their blockchain networks, thus minimizing trial-and-error adjustments and enhancing operational efficiency. The study’s findings are expected to serve as a valuable resource for both practitioners and researchers, bridging the gap between theoretical knowledge and practical implementation in the field of blockchain performance management.

This study has significant practical implications, particularly for blockchain administrators and organizations that deploy Hyperledger Fabric. Understanding the relationship between configuration parameters and performance metrics such as TPS and latency allows administrators to make data-driven adjustments to optimize system behavior. By developing a predictive model, this research provides a valuable tool that can assist in preemptively estimating the effects of parameter changes, thus enabling more strategic and efficient configurations. Such an approach can enhance overall system performance, reduce response times, and improve user satisfaction, making Hyperledger Fabric more effective for applications with high demands on transaction processing and data throughput.

In addition to its practical applications, the study contributes to the academic understanding of performance dynamics in permissioned blockchain systems. While significant research has been conducted on general blockchain performance, there remains a need for focused studies that address the specific characteristics and challenges of permissioned platforms like Hyperledger Fabric. This research expands the existing body of knowledge by empirically examining how various parameters affect throughput and latency in a controlled blockchain environment. The findings are expected to benefit researchers seeking to explore new optimization techniques or develop advanced algorithms for improving performance. Through these contributions, the study enhances the foundational understanding of blockchain performance management, paving the way for future advancements in both practical and theoretical aspects of permissioned blockchain technology.

Literature Review

Blockchain Performance Metrics

Throughput, commonly measured as TPS, is a fundamental performance metric in blockchain networks, indicating the rate at which the system processes transactions. This metric is especially crucial in environments where rapid transaction processing is essential, such as financial services, healthcare, and supply chain management. Throughput reflects the blockchain's ability to handle a high volume of transactions over a specified period, directly impacting the efficiency and effectiveness of the system. In the context of blockchain technology, high throughput is vital for ensuring that the network can meet the demands of applications requiring real-time or near-real-time data processing. For example, Hyperledger Fabric's high throughput capacity makes it well-suited for enterprise use cases that demand quick and reliable transaction processing [15].

The significance of TPS extends beyond mere transaction speed; it also affects user experience by reducing wait times for transaction confirmations, which in turn fosters user satisfaction and trust. A high TPS rate is essential in permissioned blockchains like Hyperledger Fabric, where throughput can be optimized through various configuration parameters, including block size, the number of orderer nodes, and the consensus algorithm employed. However, achieving high throughput in blockchain networks can be challenging due to trade-offs with other performance metrics, such as latency. Studies have highlighted that while increasing TPS can improve efficiency, it may also place additional demands on network resources, potentially affecting latency and overall system stability [28], [29]. Thus, throughput remains a critical metric for evaluating blockchain performance, with significant implications for both user experience and the scalability of blockchain systems.

Several factors influence the TPS rate in blockchain networks, encompassing technical, architectural, and operational dimensions. A primary factor is the consensus algorithm, which determines how transactions are validated and recorded on the blockchain. Traditional consensus mechanisms like Proof of

Work (PoW), as used by Bitcoin, are known for their robust security but often suffer from low TPS due to resource-intensive processes. For instance, Bitcoin's PoW mechanism limits its TPS to around 7. In contrast, newer consensus models, such as Multi-Byzantine Fault Tolerance (MBFT), aim to increase TPS by improving the efficiency of transaction validation. In contrast, permissioned blockchain platforms like Hyperledger Fabric can employ more efficient consensus algorithms that support higher TPS rates by reducing the computational requirements needed for transaction validation [9].

Other factors, such as block size and network architecture, also play a significant role in determining TPS. Increasing block size theoretically allows for more transactions per block, thereby improving TPS. However, this approach can lead to longer propagation times, which may compromise network stability and security, as larger blocks require more time to be transmitted and validated across nodes. Additionally, innovations in network architecture, such as sharding and off-chain solutions, have been proposed to enhance TPS by distributing the transaction load more effectively. Sharding, for example, divides the network into smaller, more manageable pieces, allowing transactions to be processed in parallel across multiple shards. Meanwhile, off-chain solutions process transactions outside the main blockchain, reducing congestion and improving transaction throughput [30], [31]. Each of these factors must be carefully balanced to optimize TPS, ensuring the blockchain network can meet the demands of its intended applications while maintaining security and reliability.

Hyperledger Fabric Architecture

In Hyperledger Fabric, a permissioned blockchain framework, the core components—orderer nodes, peers, and channels—play essential roles in maintaining the integrity, privacy, and efficiency of transactions. Orderer nodes are responsible for sequencing transactions and creating blocks, which ensures a consistent transaction order across the network. This ordering is crucial as it enables deterministic transaction processing and consensus on the ledger's state. Different consensus protocols, such as the Practical Byzantine Fault Tolerance (PBFT) or Raft consensus, can be implemented within the orderer nodes, depending on the blockchain's specific requirements for performance and security [32], [33]. By managing the order of transactions, these nodes contribute significantly to the overall performance metrics of the system, particularly in terms of latency and throughput.

Peers are the nodes that execute and validate transactions within Hyperledger Fabric, with different roles depending on their functions in the network. Endorsing peers validate proposed transactions according to specified endorsement policies, which determine the minimum number of peer endorsements required for a transaction to be considered valid. Once transactions receive sufficient endorsements, they are submitted to the orderer nodes for sequencing. After ordering, committing peers append the transactions to their copy of the ledger, updating the blockchain's state. This separation of roles among peers enables a more efficient and scalable transaction processing framework, as endorsers can independently validate transactions without waiting for the global ordering of the entire network [32], [34]. Additionally, channels facilitate data privacy by creating isolated sub-networks within the blockchain. Channels allow specific groups of peers to conduct transactions and share data privately, ensuring confidentiality and data segregation, which is particularly valuable for consortium blockchains with multiple organizations that have distinct data-sharing policies [32], [34].

Various configuration parameters, such as block size, transaction arrival rate, and the number of orders, also influence the performance of Hyperledger Fabric. Block size refers to the maximum number of transactions that can be included in a block, a parameter that directly impacts the throughput and latency of the blockchain. Larger block sizes allow more transactions per block, potentially increasing throughput; however, this can also lead to higher latency due to extended propagation times across the network [35]. Therefore, achieving an optimal block size is a balancing act, as excessively large blocks may introduce delays in transaction finalization. Studies indicate that tuning block size based on specific transaction volumes and network capacity can enhance the blockchain's overall performance [34].

Transaction arrival rate is another critical parameter that influences blockchain performance by dictating the frequency at which transactions are introduced to the network. A higher arrival rate can lead to congestion if the network's processing capacity is exceeded, which may cause delays in transaction validation and block formation. Managing this rate is essential to minimize waiting times for transactions and ensure efficient processing under varying workloads [36]. Additionally, the number of orderer nodes affects the system's scalability and fault tolerance. Increasing the number of orderers can improve the network's robustness, as a distributed ordering service can maintain high throughput and fault tolerance by distributing the transaction load. However, a greater number of orderers may also introduce additional communication overhead, requiring more time to reach consensus [34], [35]. Therefore, the configuration of orderer nodes must consider the trade-offs between scalability, fault tolerance, and potential increases in latency to optimize system performance.

Data Mining and Machine Learning in Blockchain

Predictive modeling in blockchain technology has emerged as a valuable tool across various sectors, such as healthcare, finance, and logistics, where it contributes to improved decision-making, efficiency, and security. In healthcare, predictive models can leverage blockchain's decentralized structure to provide privacy-preserving, transparent data analysis. For instance, frameworks like ModelChain have been proposed to enable predictive modeling in healthcare through federated learning on private blockchain networks, which facilitates secure and decentralized use of sensitive data [37], [38]. Such applications allow for accurate predictive analytics without compromising patient confidentiality, addressing both performance and ethical concerns in data-sensitive fields [39].

In finance, predictive modeling on blockchain platforms is used for credit risk assessment and return rate prediction, among other tasks. Leveraging blockchain's immutable and secure data structure, financial predictive models ensure data integrity, which is essential for accurate risk assessment and decision-making. For example, Liu's study on credit risk prediction models in blockchain-based supply chain finance highlights the benefits of using predictive analytics for more informed credit evaluations [40]. Moreover, advanced machine learning techniques like Long Short-Term Memory (LSTM) networks have been applied to forecast return rates, demonstrating the potential of predictive modeling in enhancing financial product management on blockchain networks [41]. These applications underscore the versatility of predictive modeling in blockchain, facilitating secure, transparent, and effective analytics in diverse sectors.

Random Forest is an ensemble learning algorithm that combines multiple decision trees to improve predictive accuracy and robustness, making it particularly effective for complex data sets. As an ensemble method, Random Forest utilizes a combination of individual decision trees, each trained on a random subset of the data, to produce an aggregated prediction. This approach mitigates the high variance often associated with single decision trees and enhances generalization on unseen data. Random Forest's ensemble structure reduces the likelihood of overfitting by averaging the outcomes of multiple trees, resulting in a stable model that performs well on various types of data.

One of the notable strengths of Random Forest is its capacity to handle non-linearity and interactions between features. In blockchain-related predictive modeling, where feature interactions can be intricate and non-linear, Random Forest's ability to capture complex relationships makes it a preferred choice. For instance, Liu et al. demonstrate that Random Forest outperforms other algorithms in scenarios with high-dimensional, heterogeneous data, which is common in blockchain environments. Moreover, Random Forest provides built-in feature importance metrics, allowing researchers to identify which variables most significantly impact the predictive outcomes, further enhancing interpretability and facilitating better-informed decisions. These advantages make Random Forest a robust tool for predictive modeling, particularly suited to the challenges and complexities of blockchain data.

Method

The research method for this study consists of several steps to ensure a comprehensive and accurate analysis. The flowchart in Figure 1 outlines the detailed steps of the research method.

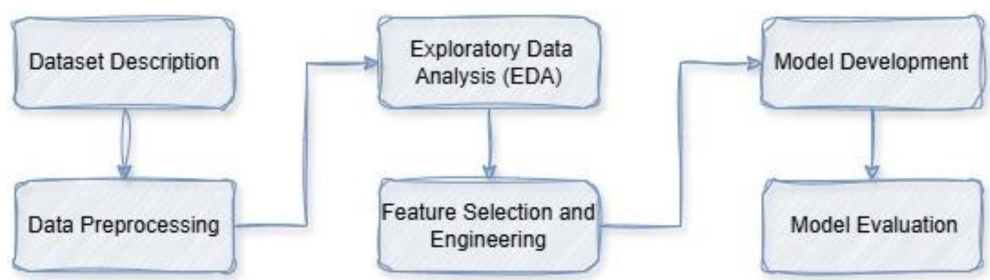

Figure 1 Research Method Flowchart

Dataset Description

The dataset used in this study was derived from performance tests conducted using Hyperledger Caliper, an open-source benchmark tool, on Hyperledger Fabric 2.3, a widely used permissioned blockchain platform. This dataset contains empirical data that reflects the system's performance under various configurations. The tests were specifically designed to capture key performance indicators across different settings, providing a robust basis for analyzing the relationship between input parameters and system performance. The data collected from these tests include metrics for transaction arrival rates, block sizes, the number of orderer nodes, and critical performance outputs such as throughput and latency. This comprehensive collection offers a granular view of how different configurations impact the overall performance of Hyperledger Fabric, facilitating predictive modeling.

The structure of the dataset consists of 24,687 entries, each representing a unique set of configuration parameters and their corresponding performance outcomes. This extensive dataset ensures a diverse range of scenarios for analysis, enhancing the reliability and generalizability of the results. The dataset includes eight fields: Set Transaction Arrival Rate, Actual Transaction Arrival Rate, Block Size, Number of Orderers, Throughput (Transactions Per Second - TPS), Average Latency, Minimum Latency, and Maximum Latency. Each entry corresponds to a unique test run, capturing the average values from 1,000 experiments to minimize error and ensure accuracy.

Data preprocessing was conducted to ensure the dataset was clean and suitable for analysis. An initial step involved checking for missing values across all columns. As indicated, no missing values were found, which negated the need for imputation or removal of records. Subsequently, outlier detection and treatment were performed using the interquartile range (IQR) method to identify and cap extreme values in numerical columns, such as Set Transaction Arrival Rate, Block Size, and Throughput. Outlier treatment is essential to prevent skewed analyses that could misrepresent model performance. Boxplots were used to visualize the distribution of values before and after capping, ensuring a balanced approach to handling outliers.

Normalization and scaling of features were carried out to standardize the data for modeling. This step ensures that all input features operate on a comparable scale, improving the stability and accuracy of the machine learning models. StandardScaler from scikit-learn was applied to transform the features, centering them around zero with unit variance. In some cases, MinMaxScaler was also considered for alternative scaling needs. The scaled dataset was saved for reproducibility, and descriptive statistics were reviewed to confirm that the transformations were correctly applied. These preprocessing steps were vital for preparing the data for predictive modeling using Random Forest Regression, enabling a consistent and effective analysis of throughput and latency predictions.

Exploratory Data Analysis (EDA)

The initial stage of exploratory data analysis involved generating descriptive statistics for the key numerical features in the dataset, including Set Transaction Arrival Rate, Actual Transaction Arrival Rate, Block Size, Orderers, Throughput, Avg Latency, Min Latency, and Max Latency. Summary statistics, including mean, median, standard deviation, minimum, maximum, and quartile values, were calculated for each feature to understand their central tendency and dispersion. For example, the mean and median values for Set Transaction Arrival Rate and Actual Transaction Arrival Rate were approximately zero after normalization, indicating a balanced distribution around the mean. The Block Size had a mean of approximately zero with a standard deviation of one, reflecting a standardized scaling. Understanding the spread of the data was essential for identifying any potential skewness or outliers that could impact model performance.

Further examination revealed that the Orderers variable had an average value of approximately six, with values ranging from three to nine, reflecting the different network configurations tested in the dataset. The throughput and latency metrics displayed a broad range, with throughput values spanning from a minimum of approximately -1.76 (normalized) to a maximum of 2.26. The distribution of latency metrics, including Avg Latency, Min Latency, and Max Latency, highlighted the variability inherent in transaction processing times. The descriptive statistics provided a foundational understanding of the dataset's structure and variability, forming a basis for further analysis.

To explore potential relationships between input parameters and performance metrics, a correlation analysis was conducted. The correlation matrix revealed the degree of linear association between features, with values ranging from -1 (perfect negative correlation) to +1 (perfect positive correlation). A heatmap of the correlation matrix was generated to visually depict these relationships, with stronger correlations indicated by darker hues. This analysis was crucial for identifying potential predictors of throughput and latency. For instance, a positive correlation was observed between Block Size and Throughput, suggesting that increasing the block size may lead to higher transaction processing rates. However, the correlation between Block Size and Avg Latency was more complex, indicating a potential trade-off that required further investigation.

The correlation analysis also highlighted weaker correlations between certain input parameters, such as the number of Orderers and throughput. This finding suggested that while increasing the number of orderer nodes might influence fault tolerance and network stability, its impact on throughput was less direct compared to other factors. The Pearson correlation coefficient formulawas used to quantify the strength of these linear relationships, offering insights into potential predictors and interactions that warranted deeper exploration.

A series of visualizations were generated to further explore the relationships between key features and performance metrics. Scatter plots were employed to visualize the relationship between Block Size and Throughput, as well as Actual Transaction Arrival Rate and Avg Latency. The scatter plot of Block Size versus Throughput revealed a trend indicating that larger block sizes were generally associated with higher throughput, though the relationship exhibited variability based on other factors such as the number of orderers. Similarly, the scatter plot of Actual Transaction Arrival Rate against Avg Latency illustrated that higher transaction rates tended to increase latency, reflecting the system's processing limits.

Box plots provided additional insights into the distribution of latency metrics across different numbers of orderers. The box plot for Avg Latency, for example, demonstrated that as the number of orderers increased, the variability in average latency also tended to rise, indicating potential network congestion or increased coordination overhead. This pattern was similarly observed for throughput distributions, highlighting the influence of network configuration on performance metrics. These visualizations offered a comprehensive view of data trends and variability, enabling a more informed approach to predictive modeling.

To gain a holistic understanding of the interactions between multiple variables, pair plots were generated, showcasing scatter plots and distribution plots for selected features, such as Block Size, Throughput, Avg Latency, and Actual Transaction Arrival Rate. These plots provided a comprehensive view of potential non-linear relationships and highlighted clusters or patterns within the data. The inclusion of hue-based differentiation by the number of orderers further illustrated how network configuration influenced these relationships. For example, distinct clusters were observed based on orderer count, indicating that network size played a role in shaping performance outcomes. The pair plots underscored the complexity of interactions within the dataset and highlighted the need for robust predictive modeling techniques to capture these dynamics accurately.

Feature Selection and Engineering

The process of feature selection was guided by analyzing the inherent feature importance metrics provided by the Random Forest model. Random Forest, as an ensemble learning method, inherently ranks features based on their contribution to reducing prediction error. This analysis provided insights into which input parameters most significantly influenced the prediction of throughput and average latency in Hyperledger Fabric. For throughput prediction, key features included Set Transaction Arrival Rate, Block Size, and the number of Orderers, each contributing differently to the model's predictive performance. The importance scores, visualized using bar plots, highlighted the relative influence of each feature, providing a clear understanding of their impact on throughput variations. The same approach was applied for average latency prediction, with feature importance indicating how network configuration parameters and transaction characteristics affected latency behavior.

The insights from the feature importance analysis not only guided the feature engineering process but also underscored the complexity of interactions within the dataset. While some features exhibited a more direct influence, others had subtler, interaction-based impacts. This process allowed for a more targeted approach to feature engineering and provided a strong foundation for optimizing predictive model performance through the inclusion of meaningful and impactful input variables.

Building on the insights gained from feature importance analysis, potential feature engineering techniques were applied to capture non-linear relationships and interactions within the dataset. Interaction terms, such as the product of Block Size and Transaction Arrival Rate, were created using polynomial feature expansion. This approach allowed for the modeling of complex dependencies and interactions that might otherwise be missed by considering individual features alone. The generated interaction features were added to the dataset, enriching the model's ability to capture nuances in how different parameters influenced throughput and latency.

Log transformations were considered and applied to handle skewed distributions in the target variables, such as Throughput and Avg Latency. Highly skewed features can adversely impact model performance by distorting error distributions, making it difficult for the model to generalize effectively. Applying log transformations reduced skewness, normalizing the distribution of these variables and improving model stability. This transformation was particularly beneficial for features exhibiting extreme values or exponential growth patterns, ensuring that all input variables contributed meaningfully to the prediction process.

After incorporating the engineered features, a second round of feature importance analysis was conducted using the Random Forest model. The updated model, which now included interaction terms and log-transformed features, was evaluated to determine the impact of these additions on predictive performance. The feature importance scores were recalculated and visualized, revealing how the engineered features influenced the model's accuracy and stability. Interaction terms that captured non-linear relationships were observed to have substantial importance, demonstrating their role in enhancing the model's predictive power. This iterative process highlighted the value of feature engineering in improving model performance and offered a data-driven approach to refining feature selection.

The updated analysis underscored the importance of continuously re-evaluating features as new transformations and interactions are introduced. By focusing on features that provided the greatest predictive value, the model was optimized for accuracy and interpretability, ensuring that all relevant interactions were effectively captured. This step was crucial in achieving a comprehensive understanding of how input parameters influenced the key performance metrics of throughput and latency in Hyperledger Fabric.

The final step in feature engineering involved updating the feature set to include all engineered features alongside the original input variables. This comprehensive feature set, which now captured interactions, non-linearities, and transformations, formed the basis for the predictive modeling process. The enriched dataset was then used to re-train the Random Forest model, further improving its ability to predict throughput and latency. This iterative approach to feature selection and engineering ensured that all relevant aspects of the data were accounted for, leading to more accurate and robust predictions of Hyperledger Fabric's performance metrics. The final feature set was saved for future model iterations and analysis, ensuring a consistent and reproducible workflow for evaluating performance in blockchain networks.

Model Development

Random Forest Regression was selected for this study due to its proven effectiveness in handling complex, non-linear relationships in large datasets, as highlighted in multiple studies. Random Forest, being an ensemble learning method, constructs multiple decision trees and aggregates their outputs to produce more accurate and robust predictions. Its ability to capture intricate feature interactions and handle diverse data distributions makes it highly suitable for modeling complex phenomena such as throughput and latency in Hyperledger Fabric blockchains. Unlike traditional linear models, Random Forest does not assume linearity in data relationships, which is crucial for accurately modeling the non-linear and interdependent nature of blockchain performance metrics. Additionally, the inherent feature importance metrics provided by the Random Forest model facilitate a better understanding of how input parameters influence performance, as demonstrated in previous literature on predictive modeling.

To ensure that the Random Forest model performed optimally, a systematic approach to model development was undertaken. The initial step involved splitting the dataset into training and testing sets, using an 80:20 ratio to allow for robust model evaluation. The training set was used to build and fine-tune the model, while the testing set served as an independent measure of performance. This approach minimized data leakage and ensured that the model's performance metrics were reflective of its generalization capabilities. The model was then trained using a Random Forest Regressor initialized with default parameters to establish a baseline for predictive performance.

The implementation of the Random Forest model began with data preparation, including the splitting of the dataset into features (predictor variables) and targets (response variables). Separate models were trained to predict throughput and average latency, leveraging the selected input features. The initial model training used default hyperparameters, after which hyperparameter tuning was conducted to optimize the model's performance. Grid Search Cross-Validation was employed to systematically explore combinations of key hyperparameters, including the number of trees (n_estimators), the maximum depth of trees (max_depth), and the minimum number of samples required to split a node (min_samples_split). This exhaustive search identified the optimal parameter settings for both throughput and latency prediction models, enhancing their predictive accuracy and stability.

After identifying the best hyperparameters, the final Random Forest models were trained using these settings. This ensured that the models were fine-tuned to maximize predictive accuracy while minimizing overfitting. The performance of the models was then evaluated using key metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2. These metrics provided a comprehensive assessment of the models' ability to predict throughput and latency based on the given input parameters. The evaluation process also included cross-validation to assess the stability and generalization capability of the models.

Model Evaluation

Model evaluation focused on measuring the accuracy and stability of the Random Forest models for both throughput and latency prediction. Performance metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2 were calculated on the test sets to provide a quantitative assessment of model accuracy. MAE measured the average magnitude of errors in the predictions, providing a straightforward indication of the prediction error's magnitude. RMSE, on the other hand, penalized larger errors more heavily, making it useful for understanding the distribution of errors in the predictions. The R^2 metric assessed the proportion of variance in the target variable explained by the model, indicating the model's overall predictive power. For throughput prediction, the Random Forest model achieved high R^2 values, demonstrating its ability to capture the key predictors and their interactions accurately. Similarly, the latency prediction model exhibited strong predictive accuracy, with low MAE and RMSE values, confirming the robustness of the approach.

To further assess model stability and generalization capability, k-fold cross-validation with k=5 was implemented. This technique involved partitioning the data into five subsets, training the model on four subsets, and validating it on the fifth, rotating this process until each subset had served as a validation set. The cross-validation results indicated that the Random Forest models consistently achieved high accuracy across different folds, confirming their robustness. Additionally, visualizations such as Actual vs. Predicted plots and residual plots were generated. The scatter plots of actual versus predicted values revealed a strong linear correlation, with points clustering closely around the line of perfect prediction, indicating accurate predictions. Residual plots showed a roughly normal distribution of residuals, suggesting that the models captured the underlying patterns in the data well and that errors were evenly distributed. This comprehensive evaluation approach demonstrated the models' effectiveness in predicting Hyperledger Fabric performance metrics accurately.

Result and Discussion

Exploratory Data Analysis Findings

The EDA began with a summary of the descriptive statistics for the key numerical features, which provided an overview of the data distribution, including the mean, median, standard deviation, minimum, and maximum values for each parameter. For example, the Set Transaction Arrival Rate had a mean of 105.0 with a median of 105.2 and a standard deviation of 56.27, indicating a fairly symmetric distribution around the central value. Similarly, the Block Size exhibited a mean of 404.56 and a median of 405.0 with a standard deviation of 229.55, reflecting considerable variability in the data. The number of orderers varied from 3 to 9, while the throughput and average latency showed considerable variation, with mean values of 91.71 TPS and 0.89 seconds, respectively. These insights provided a baseline for understanding the dataset's distribution and variability, which informed subsequent modeling steps.

The descriptive statistics further revealed potential areas for model optimization. For example, the broad range of values observed for throughput (minimum of 9.8 TPS to a maximum of 197.3 TPS) suggested that performance variations were highly influenced by input parameters such as transaction arrival rate and block size. Similarly, latency metrics (average, minimum, and maximum) exhibited high variability, indicating the potential impact of network configuration on performance. Identifying and addressing this variability through predictive modeling was essential to enhancing Hyperledger Fabric's performance.

Correlation analysis was performed to explore relationships among the input parameters and performance metrics. A correlation matrix heatmap (Figure 2) provided a visual representation of these relationships, with correlation coefficients ranging from -1 to 1. Strong positive correlations were observed between block size and throughput, indicating that larger blocks generally accommodated higher transaction rates. However, there was a trade-off with latency, as larger blocks also showed positive correlations with average and maximum latency, suggesting potential performance bottlenecks due to delayed transaction processing. The correlation analysis highlighted the interdependencies among input parameters, emphasizing the need for careful configuration optimization to balance throughput and latency.

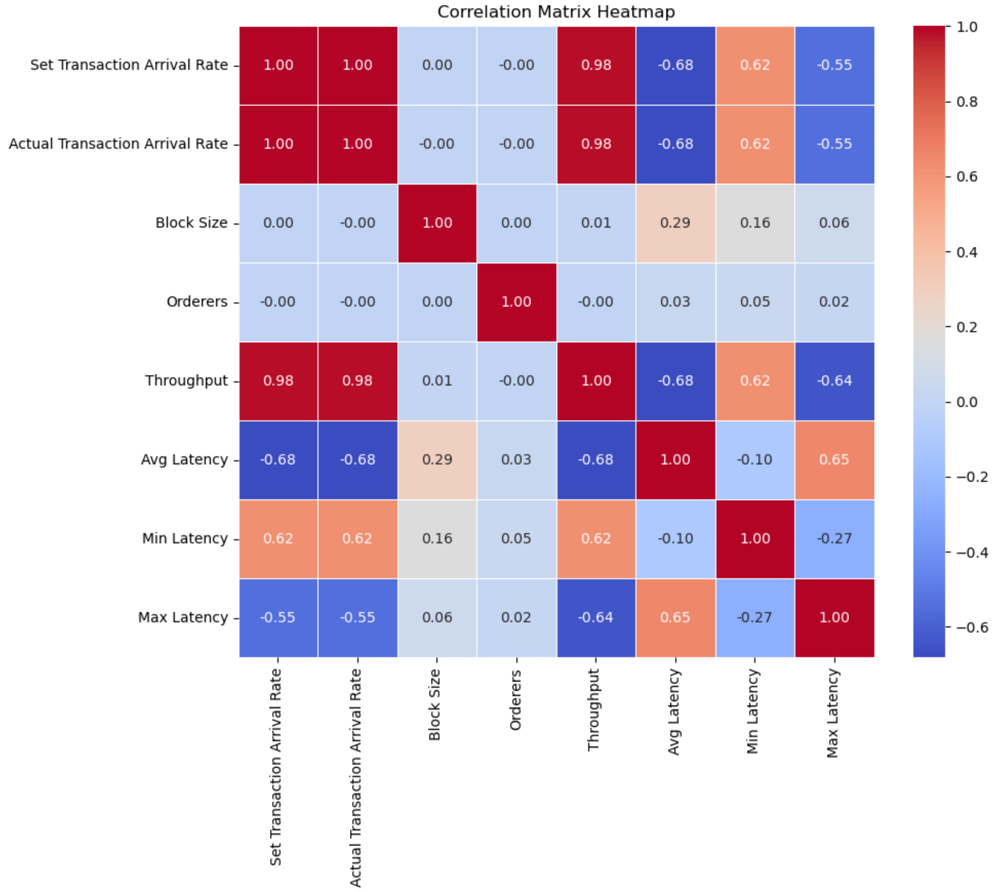

Figure 2 **Correlation Matrix Heatmap**

The correlation coefficients also revealed the limited influence of certain parameters on performance metrics. For instance, the number of orderers demonstrated a relatively weaker correlation with throughput but had a more pronounced impact on latency metrics, suggesting its role in transaction ordering and propagation delays. This insight guided feature selection and engineering processes to enhance the predictive model's accuracy by focusing on the most influential parameters.

To gain further insights into data relationships, scatter plots were used to visualize the interaction between key features and performance metrics. For example, the scatter plot of block size versus throughput, shown in Figure 3, illustrated a positive trend, with throughput generally increasing with block size up to a certain threshold before plateauing. This behavior suggested diminishing returns at larger block sizes, likely due to network saturation or increased propagation delays.

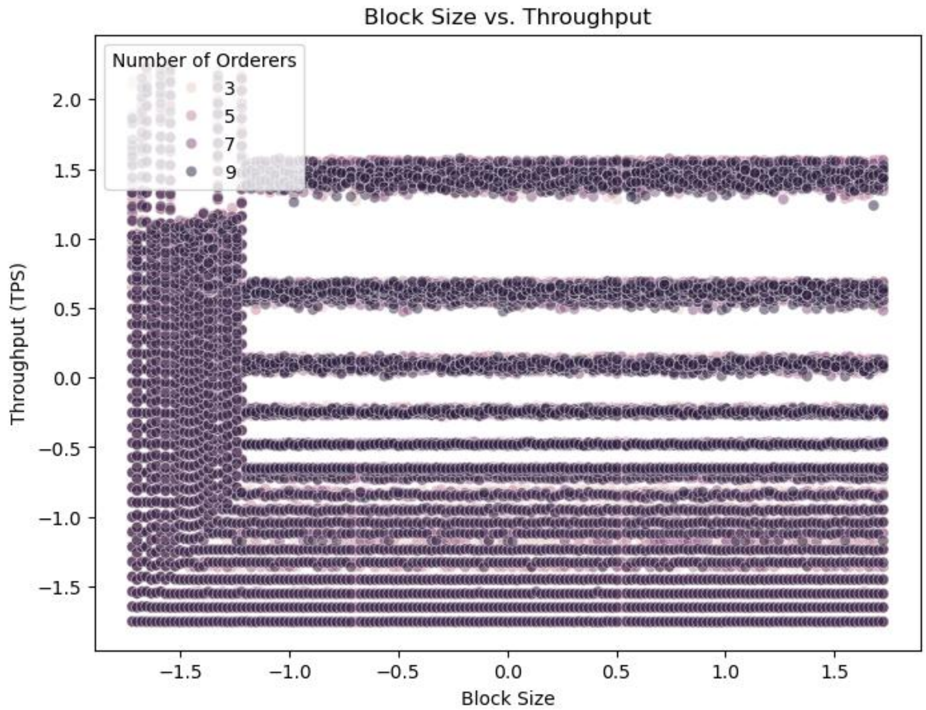

Figure 3 **Block Size vs Throughput Scatter Plot**

Similarly, a scatter plot of transaction arrival rate versus average latency (Figure 4) demonstrated an increasing trend, indicating that higher transaction rates contributed to network congestion and slower transaction confirmations.

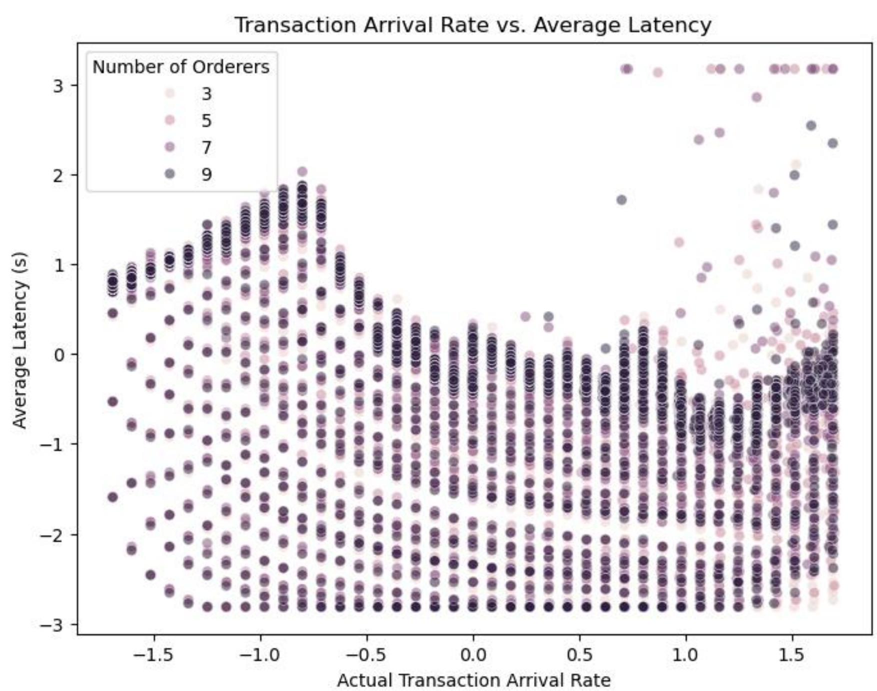

Figure 4 **Transaction Arrival Rate vs Average Latency Scatter Plot**

Box plots were employed to explore the distribution of latency metrics across different numbers of orderers, shown in Figure 5. The analysis revealed that higher numbers of orderers tended to increase average and maximum latency due to the increased complexity of reaching consensus. These findings highlighted the trade-offs involved in network configuration and the need to balance performance and fault tolerance when determining the optimal number of orderers in Hyperledger Fabric.

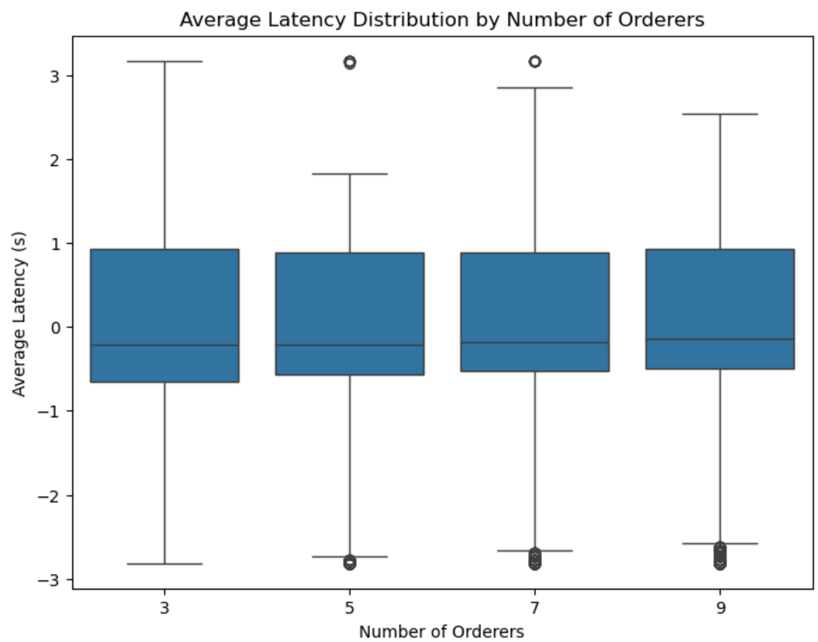

Figure 5 Average Latency Distribution Boxplot

The distribution of latency metrics was further examined using box plots, which provided a comprehensive view of variability across different configurations. Figure 6 demonstrated a skewed distribution, with a few outlier values contributing to higher maximum latency. This variability suggested that network congestion and configuration settings, such as block size and transaction arrival rate, played a significant role in determining latency. Identifying and mitigating these outliers through feature engineering and predictive modeling was crucial for improving system performance and achieving consistent transaction processing times.

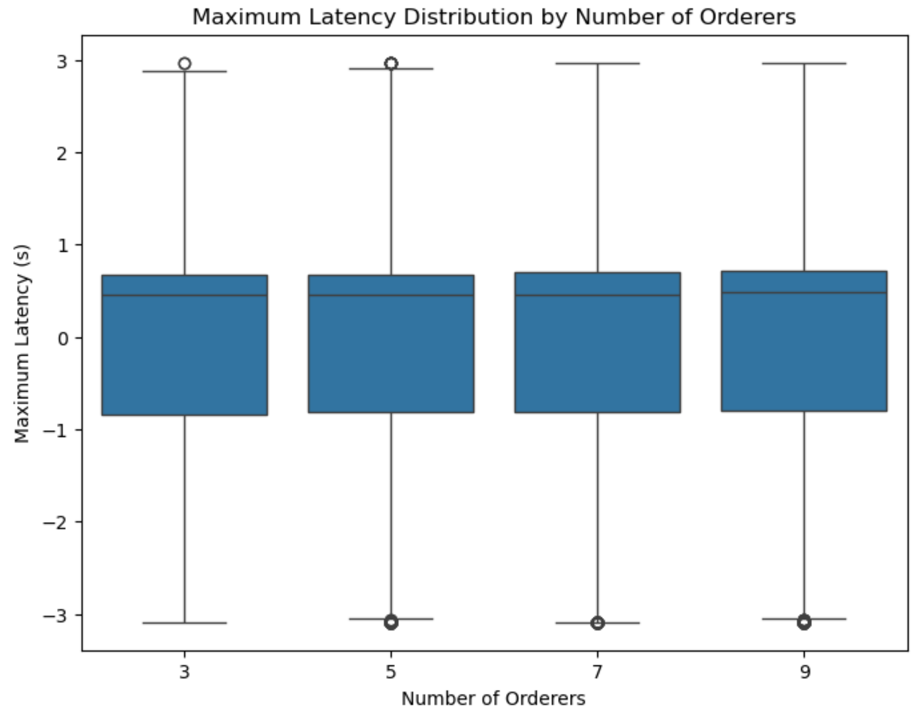

Figure 6 **Maximum Latency Distribution Boxplot**

These EDA findings laid the foundation for developing accurate predictive models to estimate throughput and latency based on network configuration parameters.

Model Performance

The Random Forest models used for predicting throughput and average latency underwent hyperparameter tuning to optimize their performance. The best hyperparameters for the throughput prediction model included a maximum depth of 20, a minimum samples split of 5, and 200 estimators. For the average latency prediction model, the optimal hyperparameters were identified as a maximum depth of 20, a minimum samples split of 2, and 200 estimators. The selection of these parameters was crucial in ensuring model robustness and accuracy, as deeper trees and an increased number of estimators allowed for better learning of complex patterns within the data, while controlling the minimum samples split mitigated the risk of overfitting.

The tuning process was performed using Grid Search with cross-validation, providing a systematic approach to identifying the best parameters. This iterative search evaluated different combinations of hyperparameters and selected the configuration that minimized prediction errors while maximizing model generalization. The results demonstrated the effectiveness of tuning in improving the model's predictive power and stability across diverse data scenarios.

The model's performance was evaluated using key metrics, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2). For throughput prediction, the Random Forest model achieved an impressive MAE of 0.0013, an RMSE of 0.0047, and an R^2 score of 1.0000, indicating a near-perfect fit with minimal deviation between predicted and actual values.

Similarly, the average latency prediction model demonstrated robust performance with an MAE of 0.0041, an RMSE of 0.0319, and an R^2 score of 0.9990. These metrics highlighted the model's accuracy and reliability in predicting throughput and latency metrics based on the selected input features. The high R^2 values indicated that the Random Forest models effectively captured the variance in the data, while the low MAE and RMSE values confirmed minimal prediction errors. This level of accuracy is critical for applications in Hyperledger Fabric, where precise prediction of performance metrics can guide network configuration and optimization efforts.

To further assess model stability, k-fold cross-validation with k=5 was implemented. The throughput prediction model demonstrated consistent performance across folds, with an average MAE of 0.0016 (±0.0003), an average RMSE of 0.0047 (±0.0009), and an average R^2 of 1.0000 (±0.0000). These results highlighted the model's robustness and its ability to generalize well across different subsets of data. For the average latency prediction model, cross-validation yielded an average MAE of 0.0063 (±0.0017), an average RMSE of 0.0386 (±0.0122), and an average R^2 of 0.9983 (±0.0011). The slightly higher variability in the latency prediction metrics suggested some sensitivity to data partitioning, but the overall performance remained highly accurate and consistent.

Cross-validation is an essential step in model validation, providing a comprehensive assessment of predictive accuracy and generalizability. The results confirmed that the Random Forest models maintained strong predictive performance even when applied to unseen data, underscoring their suitability for throughput and latency estimation in Hyperledger Fabric networks.

The efficacy of the developed Random Forest models was evident from their high predictive accuracy and stability. The near-perfect R^2 values suggested that the models captured almost all variance in the data, making them valuable tools for performance prediction in Hyperledger Fabric environments. The ability to predict throughput and latency with such precision can inform critical decisions regarding network configuration, block size, and transaction rates. Furthermore, the low MAE and RMSE values underscored the models' reliability in providing accurate predictions with minimal error. These outcomes demonstrated the potential of data-driven approaches, such as Random Forest regression, to enhance performance tuning and optimization in blockchain networks.

Discussion of Results

The Random Forest regression model demonstrated high accuracy and reliability in predicting throughput (Transactions Per Second - TPS) in Hyperledger Fabric networks. The model's near-perfect R^2 value and low error metrics underscored its effectiveness in capturing the complex relationships between input parameters and throughput. This high degree of accuracy was crucial for understanding how different network configurations influenced throughput performance. Specifically, the results showed that block size played a critical role in determining TPS, with larger block sizes generally enhancing throughput. However, this effect was moderated by network capacity and the number of orderers, as increasing block size beyond a certain threshold could lead to propagation delays and bottlenecks.

The number of orderers and transaction arrival rate were also identified as significant factors affecting throughput. A higher number of orderers improved fault tolerance but sometimes led to increased consensus delays, particularly under high transaction loads. The model's ability to accurately predict TPS under various configurations provided valuable insights into optimizing these parameters to maximize throughput without compromising network stability or performance.

In the context of average latency prediction, the Random Forest model also exhibited strong predictive performance, as evidenced by its low Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) values. The model effectively captured the non-linear interactions between configuration parameters and latency, offering a nuanced understanding of how different settings influenced transaction confirmation times. Results highlighted that higher transaction arrival rates often led to increased latency due to congestion within the network. Similarly, block size was found to have a complex effect on latency; while larger blocks could enhance throughput, they also required more time for propagation and validation, thereby increasing latency.

The number of orderers similarly impacted latency, with higher numbers generally resulting in slower transaction processing due to the increased communication overhead in reaching consensus. These findings underscored the need to strike a balance between maximizing throughput and minimizing latency to optimize Hyperledger Fabric's performance.

The insights derived from the model's predictions offered practical recommendations for configuring Hyperledger Fabric networks. Optimal performance could be achieved by carefully tuning block size and transaction arrival rates to balance throughput and latency. For example, increasing block size and maintaining a moderate number of orderers appeared beneficial for enhancing throughput while controlling latency. Moreover, dynamically adjusting transaction rates based on network capacity could help prevent congestion and maintain efficient transaction processing times. These recommendations provide a data-driven basis for configuring Hyperledger Fabric settings in enterprise deployments, leading to improved network efficiency and reliability.

The findings from this study were consistent with existing research, which has highlighted the trade-offs between throughput and latency in blockchain networks. Previous studies, such as those by Thakkar et al. (2018), similarly emphasized the influence of block size and network architecture on performance metrics. However, the present study extended these insights by offering a predictive model capable of quantifying these effects under different configurations. Unlike traditional approaches, the use of Random Forest regression allowed for a more granular understanding of parameter interactions and their impact on performance. While some variations in findings were observed due to differences in network setups and workloads, the overall alignment with prior research validated the robustness and applicability of the proposed model for Hyperledger Fabric optimization.

Limitations

One of the primary limitations of this study was the dataset's specificity to Hyperledger Fabric version 2.3. While this dataset provided valuable insights into performance metrics for a particular configuration of the blockchain platform, the findings may not generalize to newer versions or other blockchain frameworks, such as Ethereum or Corda. Differences in architecture, consensus mechanisms, and transaction processing between various blockchain systems could lead to distinct performance outcomes. Furthermore, the dataset was collected using a controlled environment with predefined parameters, which may not capture the complexities and variations observed in real-world deployments. This limited scope highlights the potential challenges of applying the model's predictions across diverse operational contexts without additional fine-tuning and validation.

Despite the inherent robustness of Random Forest regression, the model's performance is subject to certain limitations. While Random Forest's ensemble approach helps mitigate overfitting, there remains a potential risk, especially when applied to highly specific datasets with limited diversity. The model's predictive accuracy could be compromised when exposed to configurations or parameter ranges outside those present in the training data. Additionally, the predictive capability of the model was constrained by the range of configuration parameters tested, such as block size, transaction arrival rate, and the number of orderers. Expanding this range or incorporating more nuanced features could potentially improve model accuracy but would require more comprehensive data collection efforts.

Future research should focus on incorporating more diverse datasets to enhance the model's generalizability and robustness across different versions of Hyperledger Fabric and other blockchain platforms. Collecting performance data from a broader range of configurations, workloads, and network topologies would provide a more comprehensive basis for prediction and optimization. Additionally, exploring alternative machine learning algorithms, such as gradient boosting, neural networks, or hybrid models, could offer new perspectives on capturing complex, non-linear interactions between configuration parameters and performance outcomes. These avenues for future work are critical for building models that are both accurate and adaptable to the rapidly evolving blockchain landscape.

In summary, while the model demonstrated strong predictive capabilities within the constraints of the study, addressing these limitations is necessary for broader applicability and improved accuracy. Future efforts to expand dataset diversity and explore new algorithms have the potential to further enhance predictive modeling for blockchain performance, offering more precise and adaptable solutions for optimizing Hyperledger Fabric and other blockchain implementations.

Conclusion

This study focused on developing and evaluating a predictive model for estimating throughput (Transactions Per Second, TPS) and average latency in Hyperledger Fabric blockchains using Random Forest regression. The primary objectives were to identify the most influential configuration parameters affecting these performance metrics and to create an accurate predictive model to guide optimization efforts. These objectives were successfully achieved through extensive data collection using Hyperledger Caliper, exploratory data analysis, feature engineering, and model development. The model demonstrated high accuracy in predicting both throughput and latency, as indicated by performance metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2). Key findings highlighted the significant influence of parameters such as block size, transaction arrival rate, and the number of orderers on both throughput and latency.

The Random Forest regression model proved highly effective in capturing the complex relationships among configuration parameters and performance metrics. The identified hyperparameters further optimized the predictive accuracy of the model. The insights derived from these predictions underline the practical utility of data-driven approaches in enhancing blockchain performance, providing a robust framework for Hyperledger Fabric optimization.

This study contributed to the understanding of how various configuration parameters in Hyperledger Fabric affect key performance metrics like throughput and latency. By utilizing Random Forest regression, a reliable predictive tool was developed, enabling precise estimations of performance outcomes based on specific settings. This work extends the current knowledge of performance dynamics within permissioned blockchain environments and demonstrates the potential of machine learning in enhancing system efficiency. The predictive model serves as a valuable resource for researchers and practitioners, providing a data-driven methodology for fine-tuning blockchain configurations to meet specific operational needs.

The study also bridges the gap between theoretical performance optimization and practical implementation. The model's ability to predict performance outcomes based on configurable parameters can guide both academic research and industrial applications, fostering more efficient and scalable blockchain solutions.

The findings of this study offer actionable guidelines for blockchain administrators seeking to optimize Hyperledger Fabric configurations. By leveraging the predictive model, administrators can make informed decisions regarding parameter adjustments to maximize throughput and minimize latency, reducing the need for costly trial-and-error experiments. This approach enhances the efficiency and reliability of blockchain deployments, ultimately leading to improved user experiences and operational performance.

For example, administrators can optimize block size and transaction arrival rates based on predicted performance outcomes, ensuring that their configurations align with desired throughput and latency targets. This capability has far-reaching implications for sectors reliant on Hyperledger Fabric, such as finance, healthcare, and supply chain management, where performance optimization is critical for achieving timely and accurate data processing.

Future research efforts could extend the current study to include other blockchain platforms and newer versions of Hyperledger Fabric. Expanding the dataset to incorporate real-world deployment data would enhance the robustness and generalizability of the predictive model. Such an extension would also enable validation and refinement of the model under diverse operational conditions, improving its accuracy and applicability across a broader range of scenarios.

Additionally, exploring the integration of additional performance metrics and configuration parameters, such as network latency, consensus protocols, and system scalability factors, would provide a more comprehensive understanding of blockchain performance. This expanded scope would further strengthen the utility of predictive modeling in optimizing blockchain technology.

The results of this study underscore the importance of data-driven approaches in optimizing blockchain performance. Accurate predictive models, such as the one developed here, offer practical solutions for enhancing the efficiency and scalability of blockchain systems, reducing the complexity of manual configuration and testing. Ongoing research and collaboration in this domain are crucial for driving innovation and enabling broader adoption of blockchain technology. Through continuous efforts to refine predictive tools and extend their applications, the blockchain community can achieve more effective and reliable systems, fostering growth and advancement across diverse sectors.

Declarations

Author Contributions

Conceptualization: D.A.D.; Methodology: D.A.D.; Software: T.B.K.; Validation: D.A.D.; Formal Analysis: T.B.K.; Investigation: D.A.D.; Resources: T.B.K.; Data Curation: T.B.K.; Writing—Original Draft Preparation: D.A.D.; Writing—Review and Editing: T.B.K.; Visualization: T.B.K. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement

The data presented in this study are available on request from the corresponding author.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. A. Monrat, O. Schelén, and K. Andersson, "A Survey of Blockchain From the Perspectives of Applications, Challenges, and Opportunities," Ieee Access, vol. 7, pp. 117134–117151, 2019, doi: 10.1109/access.2019.2936094.

[2] R. K. Ray, "Blockchain Applications in Retail Cybersecurity: Enhancing Supply Chain Integrity, Secure Transactions, and Data Protection," J. Bus. Manag. Stud., vol. 6, no. 1, pp. 206–214, 2024, doi: 10.32996/jbms.2024.6.1.13.

[3] M. Alizadeh, K. Andersson, and O. Schelén, "Efficient Decentralized Data Storage Based on Public Blockchain and IPFS," 2020, doi: 10.1109/csde50874.2020.9411599.

[4] B. Kaur, K. Sood, S. Grima, and R. Rupeika-Apoga, "Digital Banking in Northern India: The Risks on Customer Satisfaction," Risks, vol. 9, no. 11, p. 209, 2021, doi: 10.3390/risks9110209.

[5] J. Govea, "Securing Critical Infrastructure With Blockchain Technology: An Approach to Cyber-Resilience," Computers, vol. 13, no. 5, p. 122, 2024, doi:

10.3390/computers13050122.
[6] K. Naranjo, "Security Aspects in the Implementation of Blockchain in Payment Gateway Transactions in Ecuador," Int Jour Eng Ins, vol. 2, no. 1, pp. 1–6, 2024, doi: 10.61961/injei.v2i1.13.
[7] R. Shrestha and S. Y. Nam, "Regional Blockchain for Vehicular Networks to Prevent 51% Attacks," Ieee Access, vol. 7, pp. 95033–95045, 2019, doi: 10.1109/access.2019.2928753.
[8] E. Androulaki et al., "Hyperledger Fabric," 2018, doi: 10.1145/3190508.3190538.
[9] K. Y. Thakkar, "The Power of Creativity in Print Advertising: A Study of Its Relevance in Today\'s Market," J. Manag. Res. Anal., vol. 10, no. 3, pp. 169–172, 2023, doi: 10.18231/j.jmra.2023.029.
[10] C. C. Agbo, Q. H. Mahmoud, and J. Eklund, "Blockchain Technology in Healthcare: A Systematic Review," Healthcare, vol. 7, no. 2, p. 56, 2019, doi: 10.3390/healthcare7020056.
[11] C. Stamatellis, P. Papadopoulos, N. Pitropakis, S. K. Katsikas, and W. J. Buchanan, "A Privacy-Preserving Healthcare Framework Using Hyperledger Fabric," Sensors, vol. 20, no. 22, p. 6587, 2020, doi: 10.3390/s20226587.
[12] X. Piao, "Performance Analysis of Endorsement in Hyperledger Fabric Concerning Endorsement Policies," Electronics, vol. 12, no. 20, p. 4322, 2023, doi: 10.3390/electronics12204322.
[13] S. Ghosh and M. Dutta, "Indriya: Building a Secure and Transparent Organ Donation System With Hyperledger Fabric," 2023, doi: 10.36227/techrxiv.22225999.
[14] J. Sousa, A. Bessani, and M. Vukolić, "A Byzantine Fault-Tolerant Ordering Service for the Hyperledger Fabric Blockchain Platform," 2018, doi: 10.1109/dsn.2018.00018.
[15] C. Gorenflo, S. Lee, L. Golab, and S. Keshav, "FastFabric: Scaling Hyperledger Fabric to 20,000 Transactions Per Second," 2019, doi: 10.1109/bloc.2019.8751452.
[16] C. Wang and X. Chu, "Performance Characterization and Bottleneck Analysis of Hyperledger Fabric," 2020, doi: 10.1109/icdcs47774.2020.00165.
[17] C. Gorenflo, S. Lee, L. Golab, and S. Keshav, "FastFabric: Scaling Hyperledger Fabric to 20 000 Transactions Per Second," Int. J. Netw. Manag., vol. 30, no. 5, 2020, doi: 10.1002/nem.2099.
[18] Q. Nasir, I. Qasse, M. A. Talib, and A. B. Nassif, "Performance Analysis of Hyperledger Fabric Platforms," Secur. Commun. Netw., vol. 2018, pp. 1–14, 2018, doi: 10.1155/2018/3976093.
[19] O. I. Kadhum, "Evaluating the Performance of a Multi-Organizational E-Government Platform on Hyperledger Fabric With Fuzzy Logic-Enhanced Multi-Channel Connectivity," Ingénierie Systèmes Inf., vol. 29, no. 2, pp. 571–580, 2024, doi: 10.18280/isi.290218.
[20] J. R. Batmetan and T. Hariguna, "Sentiment Unleashed: Electric Vehicle Incentives Under the Lens of Support Vector Machine and TF-IDF Analysis," J. Appl. Data Sci., vol. 5, no. 1, Art. no. 1, Jan. 2024, doi: 10.47738/jads.v5i1.162.
[21] M. Liebenlito, N. Inayah, E. Choerunnisa, T. E. Sutanto, and S. Inna, "Active learning on Indonesian Twitter sentiment analysis using uncertainty sampling," J. Appl. Data Sci., vol. 5, no. 1, Art. no. 1, Jan. 2024, doi: 10.47738/jads.v5i1.144.
[22] B. H. Hayadi and I. M. M. E. Emary, "Predicting Campaign ROI Using Decision Trees and Random Forests in Digital Marketing," J. Digit. Mark. Digit. Curr., vol. 1, no. 1, Art. no. 1, May 2024, doi: 10.47738/jdmdc.v1i1.5.
[23] Henderi and Q. Siddique, "Comparative Analysis of Sentiment Classification Techniques on Flipkart Product Reviews: A Study Using Logistic Regression, SVC, Random Forest, and Gradient Boosting," J. Digit. Mark. Digit. Curr., vol. 1, no. 1, Art. no. 1, May 2024, doi: 10.47738/jdmdc.v1i1.4.
[24] Hery and A. E. Widjaja, "Predictive Modeling of Blockchain Stability Using Machine Learning to Enhance Network Resilience," J. Curr. Res. Blockchain, vol. 1, no. 2, Art. no. 2, Sep. 2024, doi: 10.47738/jcrb.v1i2.15.

[25] T. Wahyuningsih and S. C. Chen, "Analyzing Sentiment Trends and Patterns in Bitcoin-Related Tweets Using TF-IDF Vectorization and K-Means Clustering," J. Curr. Res. Blockchain, vol. 1, no. 1, Art. no. 1, Jun. 2024, doi: 10.47738/jcrb.v1i1.11.

[26] B. Srinivasan and T. Wahyuningsih, "Navigating Financial Transactions in the Metaverse: Risk Analysis, Anomaly Detection, and Regulatory Implications," Int. J. Res. Metaverese, vol. 1, no. 1, Art. no. 1, Jun. 2024, doi: 10.47738/ijrm.v1i1.5.

[27] A. R. Yadulla, M. H. Maturi, K. Meduri, and G. S. Nadella, "Sales Trends and Price Determinants in the Virtual Property Market: Insights from Blockchain-Based Platforms," Int. J. Res. Metaverese, vol. 1, no. 2, Art. no. 2, Sep. 2024, doi: 10.47738/ijrm.v1i2.9.

[28] Y. Zhou and K.-D. Kang, "A Federated Approach for Increasing the Timely Throughput of Real-Time Data Services," 2012, doi: 10.1109/rtas.2012.8.

[29] K. Ren, A. Thomson, and D. J. Abadi, "Lightweight Locking for Main Memory Database Systems," Proc. VLDB Endow., vol. 6, no. 2, pp. 145–156, 2012, doi: 10.14778/2535568.2448947.

[30] Z. Hong, S. Guo, E. Zhou, W. Chen, H. Huang, and A. Y. Zomaya, "GriDB: Scaling Blockchain Database via Sharding and Off-Chain Cross-Shard Mechanism," Proc. VLDB Endow., vol. 16, no. 7, pp. 1685–1698, 2023, doi: 10.14778/3587136.3587143.

[31] C. Sguanci, "Layer 2 Blockchain Scaling: A Survey," 2021, doi: 10.48550/arxiv.2107.10881.

[32] M. Brandenburger, "Blockchain and Trusted Computing: Problems, Pitfalls, and a Solution for Hyperledger Fabric," 2018, doi: 10.48550/arxiv.1805.08541.

[33] S. Rusch, K. Bleeke, and R. Kapitza, "Bloxy: Providing Transparent and Generic BFT-Based Ordering Services for Blockchains," 2019, doi: 10.1109/srds47363.2019.00043.

[34] P. Thakkar, S. Nathan, and B. Viswanathan, "Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform," 2018, doi: 10.1109/mascots.2018.00034.

[35] V. Amelin, E. Gatiyatullin, N. S. Romanov, R. Samarkhanov, R. Vasilyev, and Y. Yanovich, "Black-Box for Blockchain Parameters Adjustment," Ieee Access, vol. 10, pp. 101795–101802, 2022, doi: 10.1109/access.2022.3208702.

[36] J. Qi, J. Yu, and S. Jin, "Nash Equilibrium and Social Optimization of Transactions in Blockchain System Based on Discrete-Time Queue," Ieee Access, vol. 8, pp. 73614–73622, 2020, doi: 10.1109/access.2020.2986084.

[37] T.-T. Kuo, "ModelChain: Decentralized Privacy-Preserving Healthcare Predictive Modeling Framework on Private Blockchain Networks," 2018, doi: 10.48550/arxiv.1802.01746.

[38] X. Liang, "Architectural Design of a Blockchain-Enabled, Federated Learning Platform for Algorithmic Fairness in Predictive Health Care: Design Science Study," J. Med. Internet Res., vol. 25, p. e46547, 2023, doi: 10.2196/46547.

[39] L. Fusco, L. S. Sica, A. Parola, and L. Aleni Sestito, "Vocational identity flexibility and psychosocial functioning in Italian high school students," Int. J. Sch. Educ. Psychol., vol. 10, no. 1, pp. 144–154, Dec. 2020, doi: 10.1080/21683603.2020.1841050.

[40] Y. Liu, "Research on Credit Risk Prediction Method of Blockchain Applied to Supply Chain Finance," Icst Trans. Scalable Inf. Syst., 2024, doi: 10.4108/eetsis.5300.

[41] N. Metawa, M. I. Alghamdi, I. M. El‐Hasnony, and M. Elhoseny, "Return Rate Prediction in Blockchain Financial Products Using Deep Learning," Sustainability, vol. 13, no. 21, p. 11901, 2021, doi: 10.3390/su132111901.