AI-Based Blockchain Network Optimization Report

1. Introduction

In this project, we aimed to optimize blockchain network performance using AI-based models. We focused on predicting network congestion and latency by analyzing various features such as gas price, network latency, and block propagation time. Our approach involves extensive data collection, preprocessing, and model deployment to achieve significant performance improvements in blockchain networks.

2. Motivation

Blockchain networks are increasingly essential for secure and transparent transactions. However, network congestion and latency can significantly affect their performance, leading to higher transaction costs and slower processing times. By leveraging AI, we can predict and mitigate these issues, leading to more efficient blockchain operations. This project seeks to enhance the reliability and efficiency of blockchain networks through data-driven insights and optimization models.

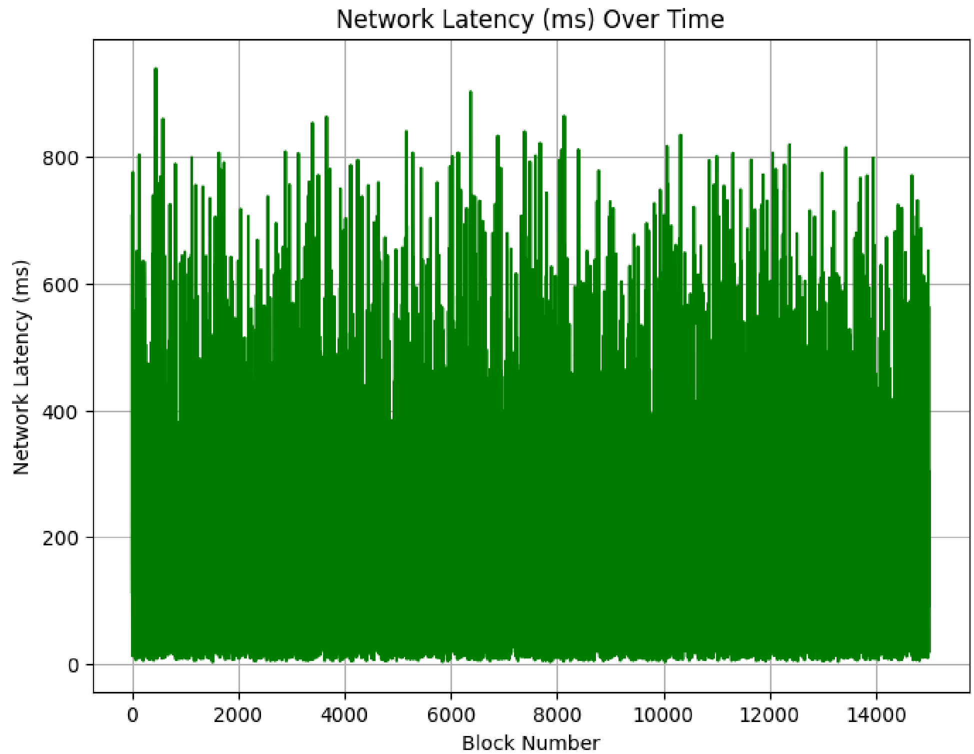
3. Dataset Description

The data for this project was collected from multiple sources, including Etherscan, Infura, and several other websites. Due to the nature of blockchain data, which is not readily available, we had to scrape and aggregate data from various platforms. This dataset includes transaction details, network performance metrics, and block information, which were essential for our analysis and model training. Key columns derived include:

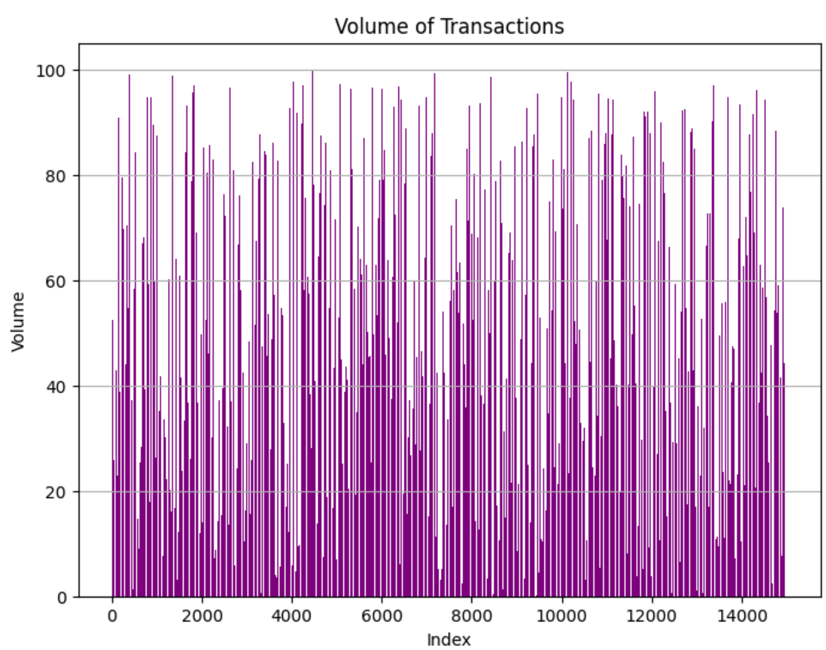
- **Transaction Details**: Transaction ID, gas price, transaction time, etc.
- **Network Metrics**: Latency, block propagation time, etc.
- **Derived Features**: Congestion category, time bins, etc.

4. Graphs and Plots

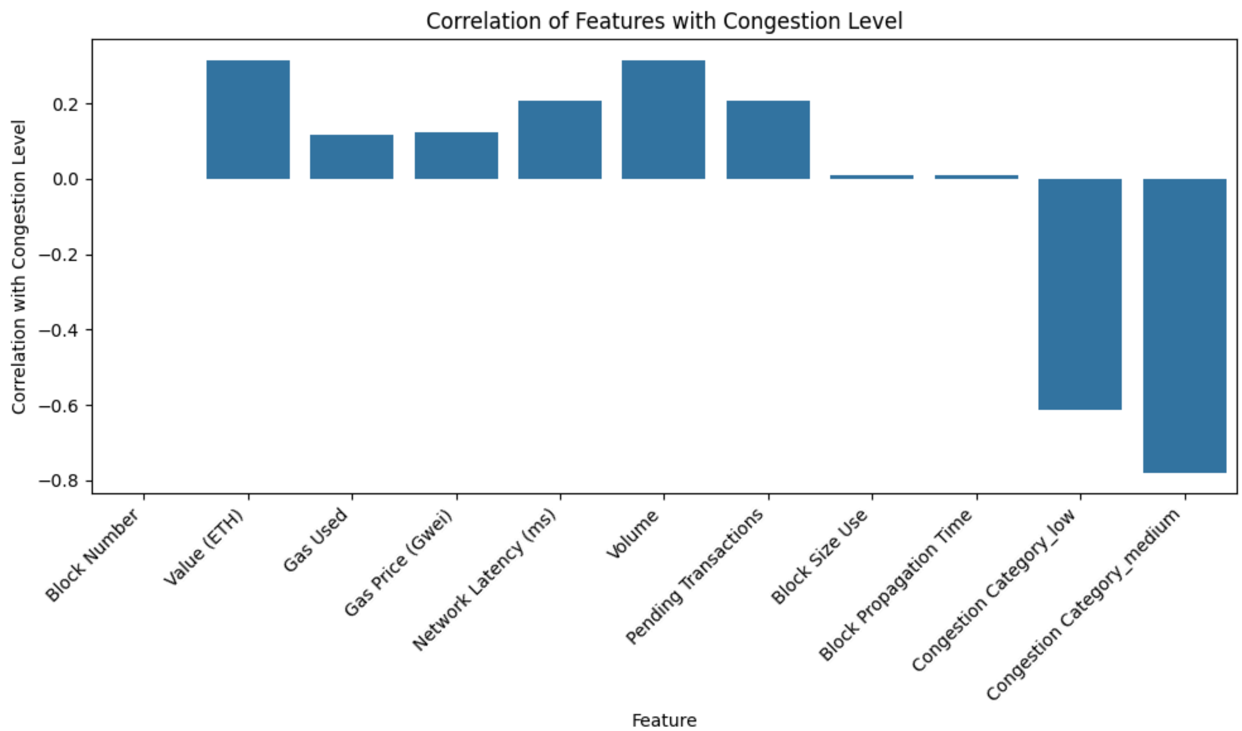
Line Plot of Network Latency (ms) Over Time (Block Number)



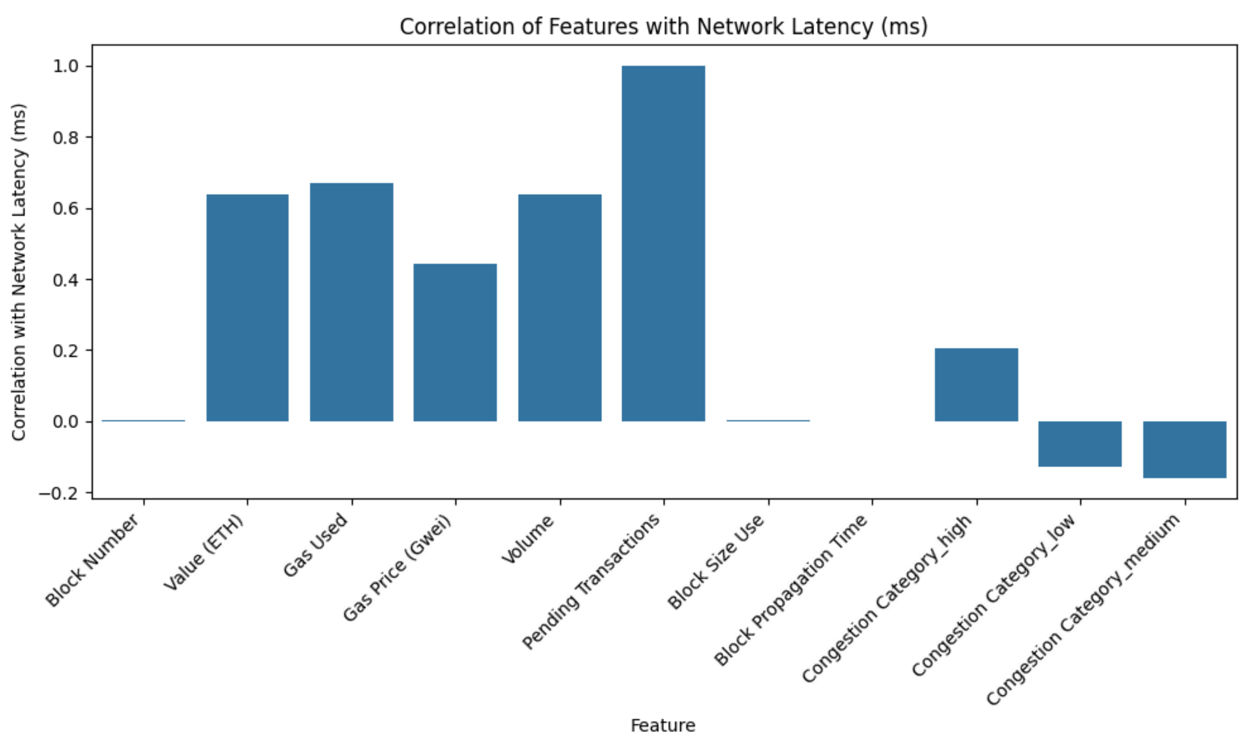
Bar Chart of Volume



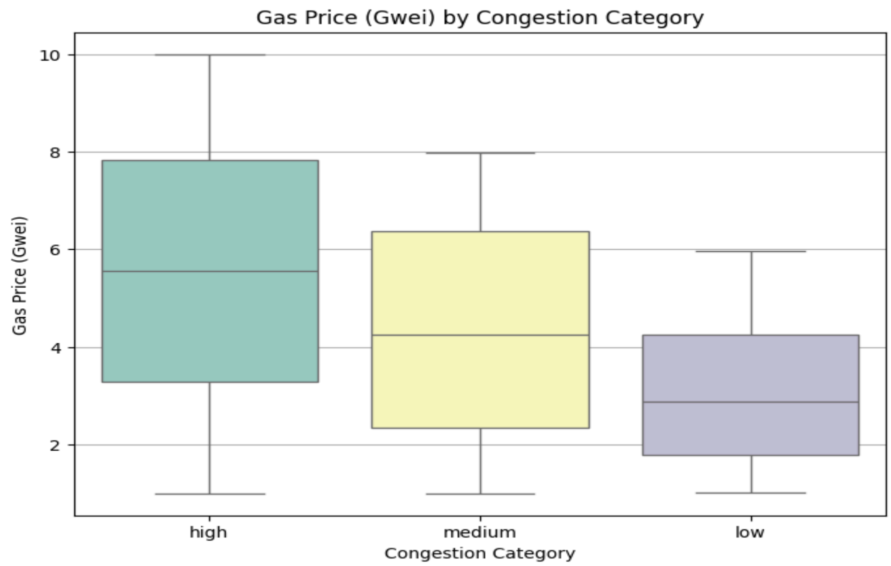
Correlation of Other Columns with Network Congestion



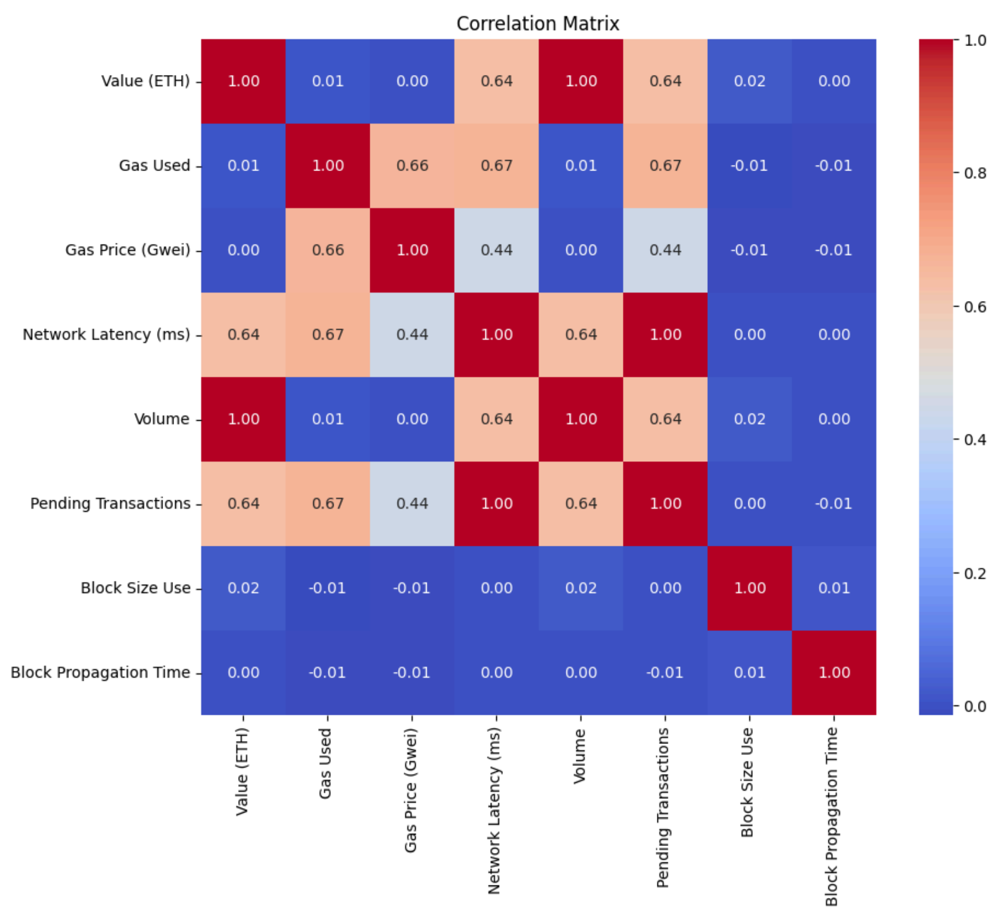
Correlation of Other Columns with Network Latency



Box Plot for Congestion Category



Correlation Matrix



5. Models Used and Their Results

In this section, we discuss the AI models used for predicting network congestion and latency, along with the results achieved.

5.1 Random Forest Regressor

- **Optimization Model for Network Latency with Random Forest Regressor**

```
Mean Squared Error (Latency): 2.9980119060071857e-07
```

- **Optimization Model for Network Congestion with Random Forest Regressor**

```
Accuracy (Network Congestion): 0.9996666666666667
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2903
           1       1.00      1.00      1.00        33
           2       1.00      0.98      0.99        64

   micro avg       1.00      1.00      1.00      3000
   macro avg       1.00      0.99      1.00      3000
weighted avg       1.00      1.00      1.00      3000
 samples avg       1.00      1.00      1.00      3000
```

5.2 Long Short-Term Memory (LSTM)

- **Network Congestion with LSTM**

```
Accuracy (LSTM): 0.985
Classification Report:
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      2903
           1       0.81      0.79      0.80        33
           2       0.69      0.55      0.61        64

    accuracy                           0.98      3000
   macro avg       0.83      0.78      0.80      3000
weighted avg       0.98      0.98      0.98      3000
```

- **Network Latency with LSTM**

```
Mean Squared Error: 0.055321014233807315
```

5.3 Artificial Neural Network (ANN) with GridSearch CV

```
Best parameters for network latency prediction:
{'activation': 'relu', 'alpha': 0.01, 'hidden_layer_sizes': (100, 100), 'learning_rate': 'constant', 'solver':
'adam'}
Mean Squared Error for network latency prediction: 2.1166627729410488e-05
```

5.4 Ridge Regression

- **Network Latency with Ridge Regression MSE (Mean Squared Error)**

```
2.0370881711152637e-06
```

6. Problems Faced

Data Collection Challenges

Blockchain data is not easily accessible, requiring extensive web scraping to gather the necessary information. This process was time-consuming and required significant effort to ensure data quality and completeness.

Data Preprocessing Issues

The collected data had to be preprocessed to derive useful columns and categorize the data into meaningful bins. Ensuring the accuracy of derived features and handling missing or inconsistent data were major challenges.

Model Training Difficulties

Training models on large and complex datasets posed challenges, including long training times and the risk of overfitting. Balancing model complexity with generalization was crucial to achieving reliable predictions.

7. Solutions

Enhanced Data Scraping Techniques

We developed robust scraping scripts and used multiple data sources to ensure comprehensive data collection. This included error handling and data validation steps to improve data quality.

Model Optimization

To address training challenges, we used hyperparameter tuning techniques like GridSearch CV and regularization methods. These approaches helped in optimizing model performance and reducing overfitting.

8. Conclusion

This project successfully demonstrated the potential of AI-based models in optimizing blockchain network performance. By predicting network congestion and latency, we can take proactive measures to improve transaction efficiency and reduce costs. The insights gained from this project can be applied to enhance the scalability and reliability of blockchain networks, paving the way for broader adoption and use in various applications.