** Work Report: Financial Time Series Prediction with LSTM and GRU Models**

*Prepared by [**Nahid Shahab**] on [**2025-01-28**]*

"Here is the link to my report: Google Colab Link."

https://colab.research.google.com/drive/1YufPjwyDXU3g-c0dSd_743XcF49yuG8K?authuser=1

And "Here is the link to my report :GitHub link

https://github.com/NShahab/Uniswap-Decentralized-Marketplace

---

1. Introduction

The objective of this project is to predict financial time series data using advanced deep learning techniques. The focus is on leveraging Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks to capture the temporal dependencies in the data. The project specifically deals with cryptocurrency price prediction, using historical data to forecast future trends. Python, TensorFlow, and related libraries were employed to implement the models.

---

2. Data Loading and Preprocessing

2.1 Data Loading

- The historical cryptocurrency price data was loaded from a CSV file using Pandas.
- Relevant columns (open_time, open, high, low, close, and volume) were extracted.

2.2 Data Transformation

- The open_time column was converted to datetime format and set as the index for better time series handling.
- Data was resampled to different timeframes (e.g., hourly, 4-hourly, daily) to accommodate different analysis resolutions.

2.3 Feature Engineering

- Technical indicators were added to enrich the feature set:
    - **Simple Moving Average (SMA):** Calculated using a rolling window.
    - **Relative Strength Index (RSI):** Computed to capture momentum in the price movements.

- **Bollinger Bands:** Added to indicate price volatility.

---

3. Data Normalization and Sequence Creation

3.1 Normalization

- All features were normalized using MinMaxScaler from the sklearn library to scale values between 0 and 1, ensuring stable model training.

3.2 Sequence Creation

- Historical sequences of 50 time steps were created to train the models.
- A sliding window approach was used to form input-output pairs where each sequence corresponds to the next time step's prediction.

---

4. Model Architecture

4.1 LSTM Model

- **Layers:**
  - Two LSTM layers with 64 units each and ReLU activation.
  - Dropout layers (rate = 0.2) to mitigate overfitting.
  - Dense layers for final predictions.
- The model was compiled with the Adam optimizer and Mean Squared Error (MSE) loss function.

4.2 GRU Model

- **Layers:**
  - Two GRU layers with 64 units each and ReLU activation.
  - Dropout layers (rate = 0.2).
  - Dense layers for output generation.
- Similar compilation settings as the LSTM model.

---

5. Model Training and Evaluation

5.1 Training Process

- Data was split into training (80%) and testing (20%) sets.
- Models were trained using Early Stopping to monitor validation loss and halt training when no improvement was observed.
- Batch size and epochs were tuned for optimal performance.

5.2 Evaluation Metrics

The models were evaluated on the testing set using the following metrics:

- **Mean Squared Error (MSE):** Measures the average squared difference between predictions and actual values.
- **Root Mean Squared Error (RMSE):** Square root of MSE for interpretability.
- **Mean Absolute Error (MAE):** Average absolute difference between predictions and actual values.
- **R-squared (R^2):** Indicates the proportion of variance explained by the model.

---

6. Results and Analysis

6.1 Quantitative Results

Comparison of evaluation metrics and next predicted prices for different timeframes:

1 Hour:

- **Model: LSTM**
  - MSE: 52,597,412.63
  - RMSE: 7,252.41
  - MAE: 6,706.79
  - R^2: 0.86
  - Next Predicted Price: 84,193.25
- **Model: GRU**
  - MSE: 97,053,193.73
  - RMSE: 9,851.56
  - MAE: 8,549.18
  - R^2: 0.75
  - Next Predicted Price: 76,525.74

4 Hours:

- **Model: LSTM**
  - MSE: 5,087,262.81
  - RMSE: 2,255.50

    - MAE: 1,669.39
    - R^2: 0.99
    - Next Predicted Price: 98,266.94
- **Model: GRU**
    - MSE: 49,097,389.30
    - RMSE: 7,006.95
    - MAE: 6,526.16
    - R^2: 0.87
    - Next Predicted Price: 84,181.45

1 Day:

- **Model: LSTM**
    - MSE: 11,292,165.31
    - RMSE: 3,360.38
    - MAE: 2,412.14
    - R^2: 0.97
    - Next Predicted Price: 96,741.01
- **Model: GRU**
    - MSE: 14,843,537.47
    - RMSE: 3,852.73
    - MAE: 2,776.86
    - R^2: 0.96
    - Next Predicted Price: 94,420.26

6.2 Visualization

- Plots were generated to compare predicted vs. actual prices for different timeframes.
- Loss curves for training and validation were analyzed to ensure convergence.

6.3 Model Comparison

- LSTM demonstrated better performance on long-term dependencies, while GRU was faster to train and computationally efficient.

---

7. Conclusion

This project successfully implemented LSTM and GRU models for cryptocurrency price prediction. Key takeaways include:

- LSTM performed better for capturing long-term dependencies.
- Feature engineering significantly improved model accuracy.

- Early Stopping and dropout effectively mitigated overfitting.

Future Work

- Experiment with hybrid models combining LSTM and GRU.
- Incorporate external factors (e.g., market news) as additional features.
- Optimize hyperparameters using grid search or Bayesian optimization.

---

8. Appendix

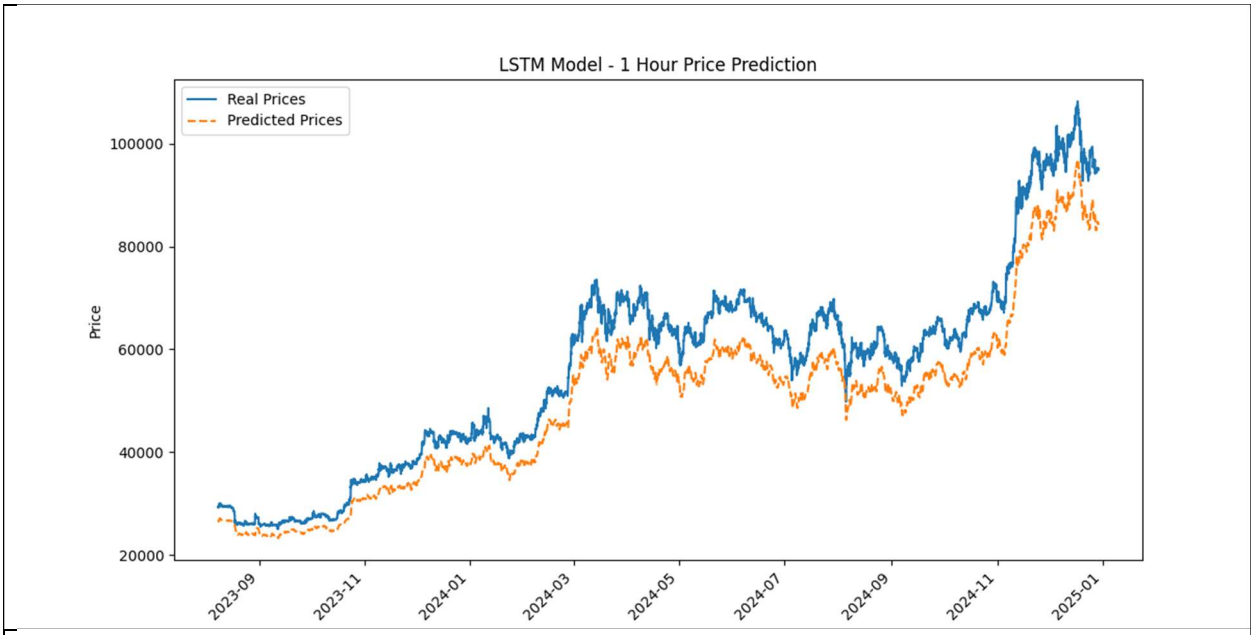
Sample Code Snippet

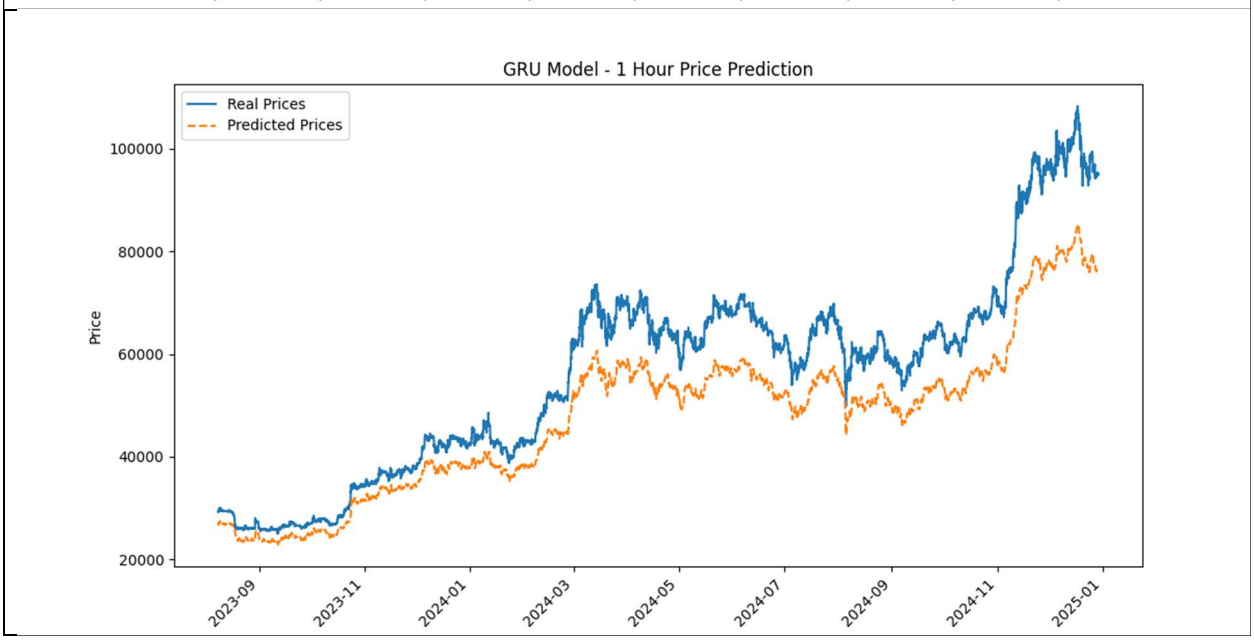
Example of LSTM model implementation

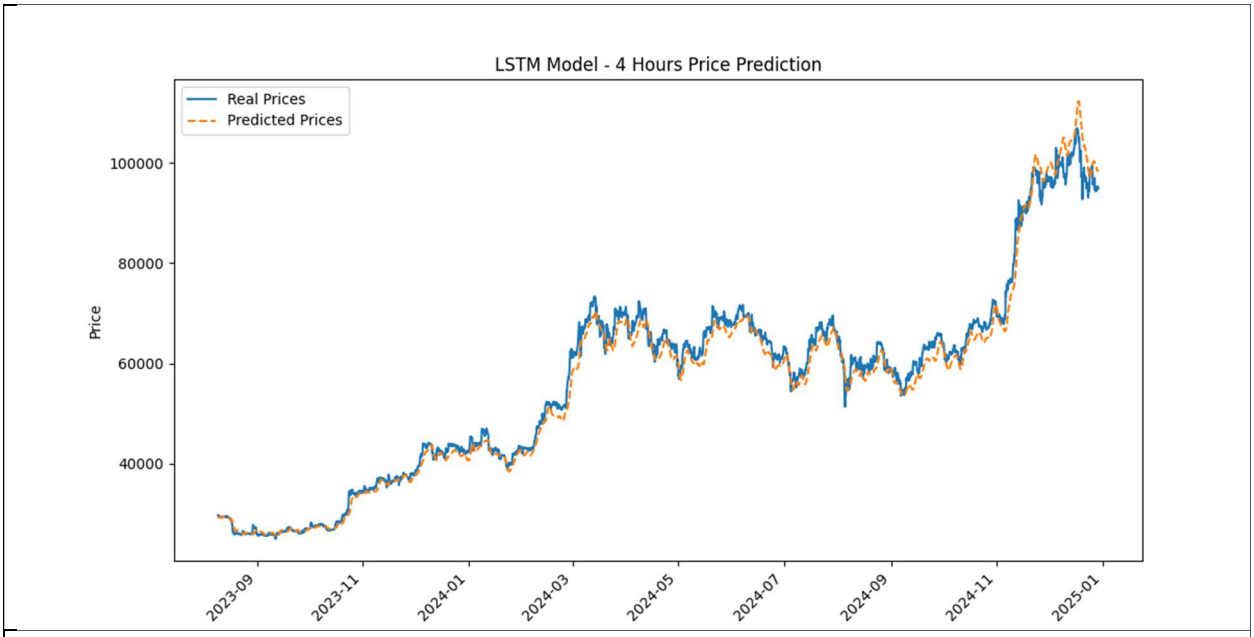
```
# Function to build the LSTM model
def build_lstm_model(input_shape):
    lstm_model = Sequential([
        LSTM(64, activation='relu', return_sequences=True,
input_shape=input_shape),
        Dropout(0.2),
        LSTM(64, activation='relu', return_sequences=False),
        Dropout(0.2),
        Dense(32, activation='relu'),
        Dense(1)
    ])
    lstm_model.compile(optimizer='adam', loss='mse')
    return lstm_model
# Function to train the model
def train_model(model, X_train, y_train, X_val, y_val, epochs=100,
batch_size=32):
    early_stopping = EarlyStopping(monitor='val_loss', patience=10,
restore_best_weights=True)

    history = model.fit(
        X_train, y_train,
        validation_data=(X_val, y_val),
        epochs=epochs,
        batch_size=batch_size,
        callbacks=[early_stopping],
        verbose=1
```

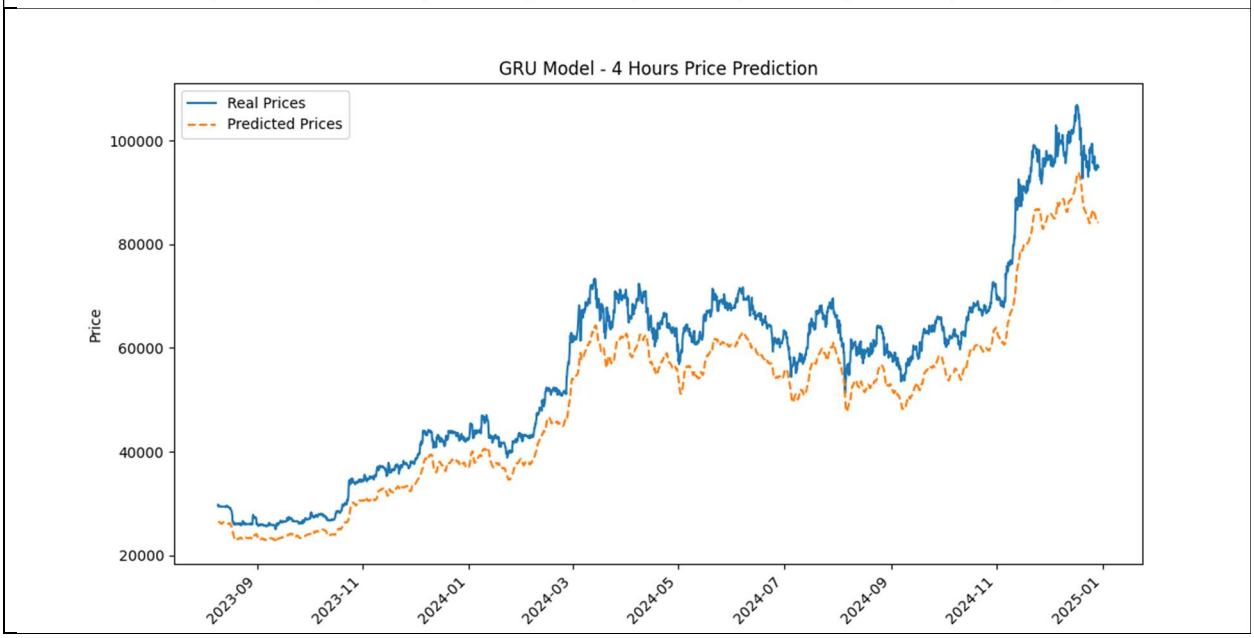
```
    )

    return model, history
```

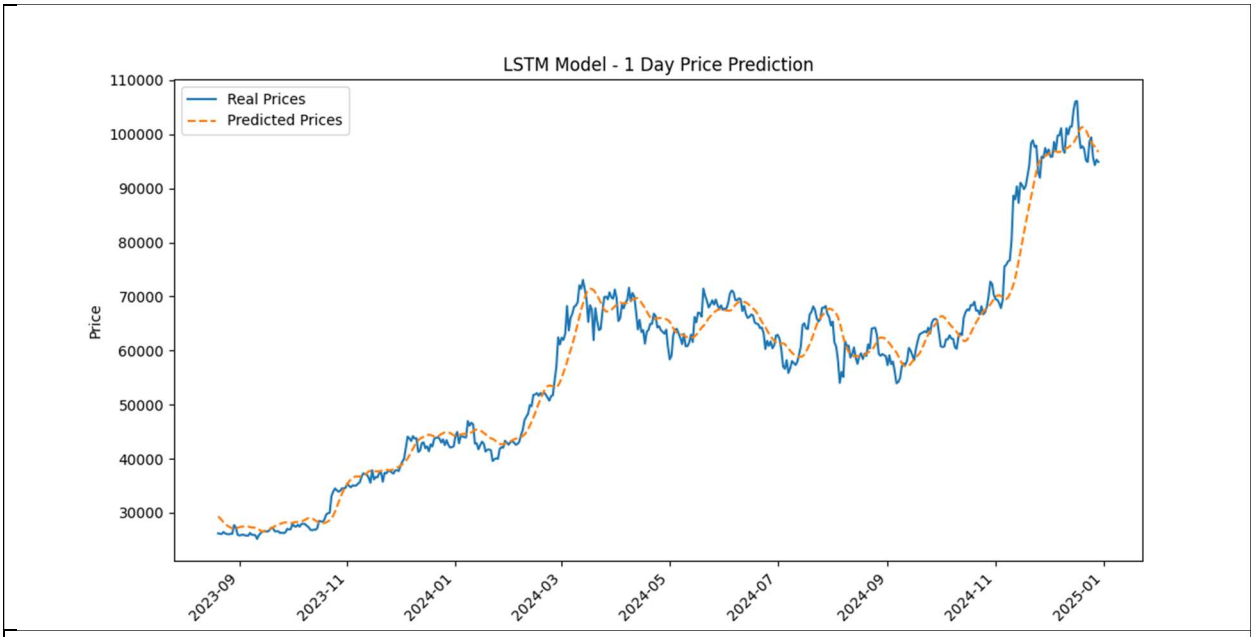
This document provides a comprehensive overview of the project, detailing each step and its outcomes.

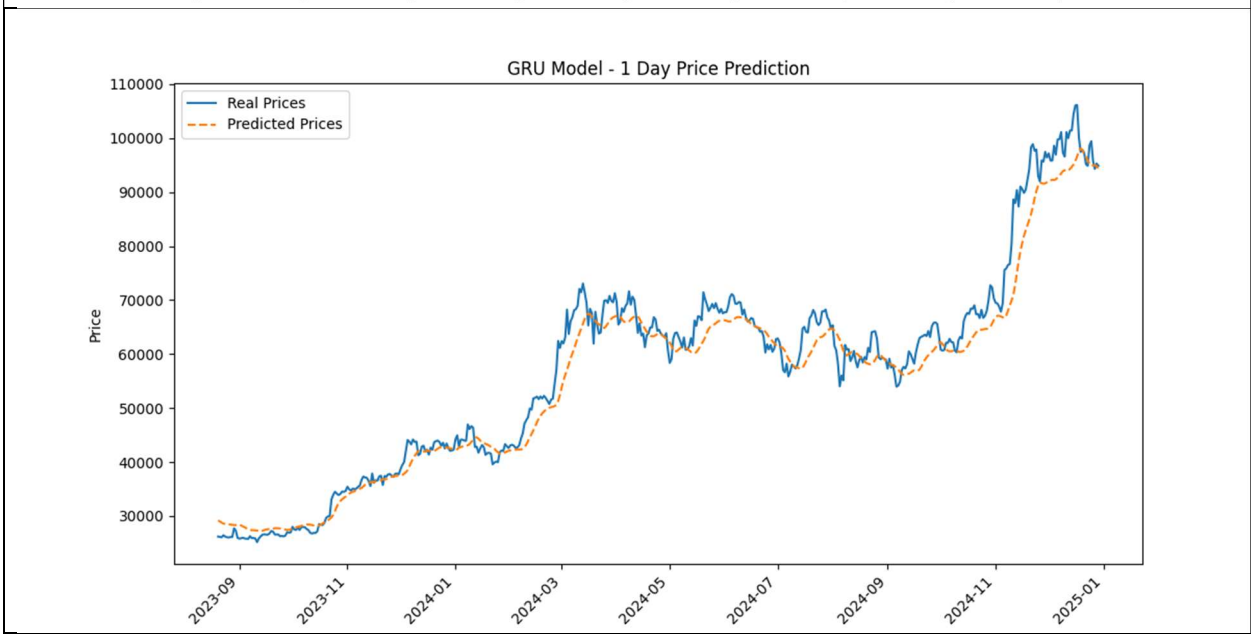


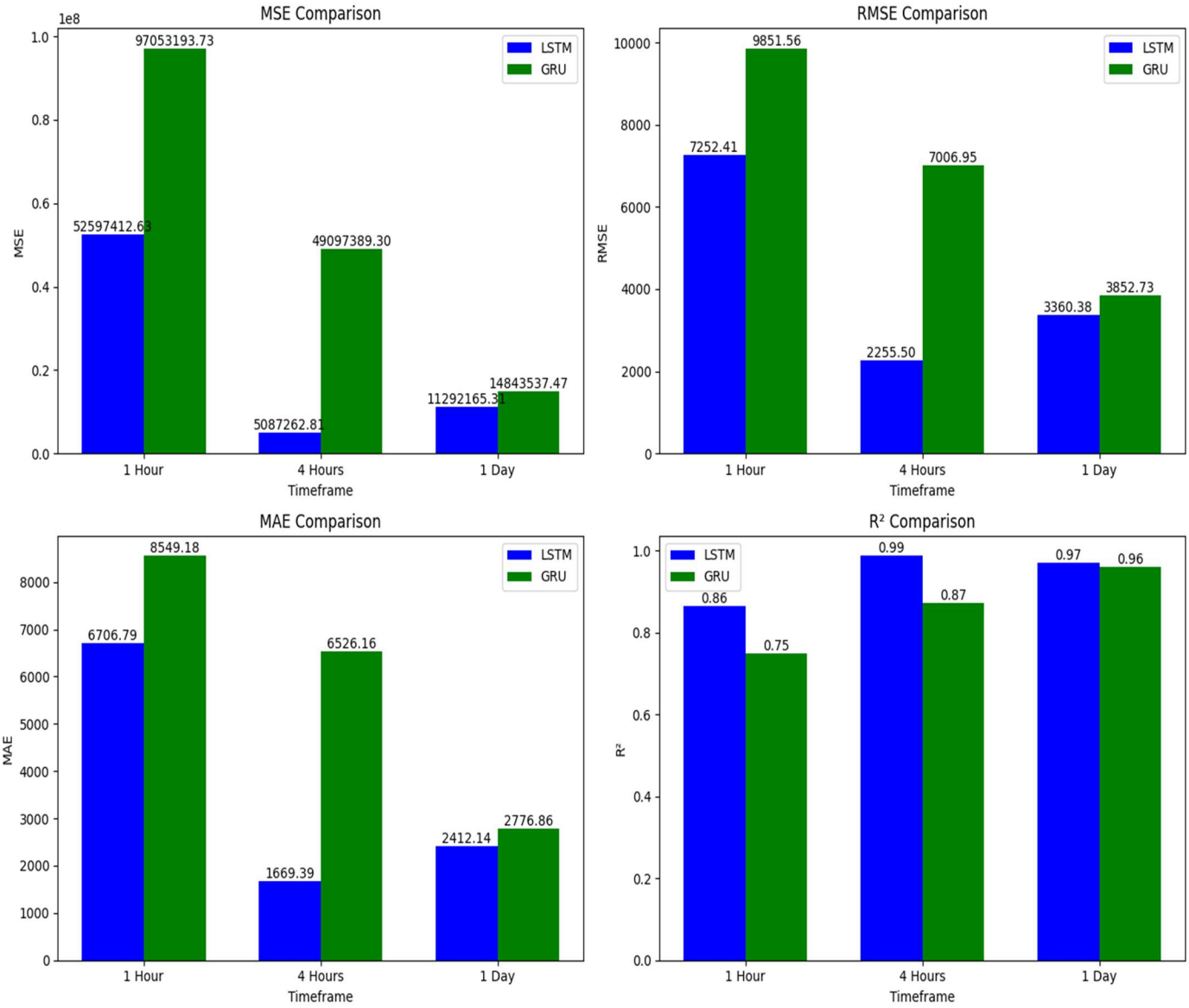



LSTM Model - 4 Hours Price Prediction
Real Prices
Predicted Prices
Price
100000
80000
60000
40000
2023-09
2023-11
2024-01
2024-03
2024-05
2024-07
2024-09
2024-11
2025-01


GRU Model - 4 Hours Price Prediction
Real Prices
Predicted Prices
Price
100000
80000
60000
40000
20000
2023-09
2023-11
2024-01
2024-03
2024-05
2024-07
2024-09
2024-11
2025-01


LSTM Model - 1 Day Price Prediction
Real Prices
Predicted Prices
Price
110000
100000
90000
80000
70000
60000
50000
40000
30000
2023-09
2023-11
2024-01
2024-03
2024-05
2024-07
2024-09
2024-11
2025-01


GRU Model - 1 Day Price Prediction
Real Prices
Predicted Prices
Price
110000
100000
90000
80000
70000
60000
50000
40000
30000
2023-09
2023-11
2024-01
2024-03
2024-05
2024-07
2024-09
2024-11
2025-01


MSE Comparison
1e8
MSE
LSTM
GRU
52597412.63
97053193.73
5087262.81
49097389.30
11292165.31
14843537.47
1 Hour
4 Hours
1 Day
Timeframe
RMSE Comparison
RMSE
7252.41
9851.56
2255.50
7006.95
3360.38
3852.73
MAE Comparison
MAE
6706.79
8549.18
1669.39
6526.16
2412.14
2776.86
R² Comparison
R²
0.86
0.75
0.99
0.87
0.97
0.96