Analysis of LSTM Models for NVIDIA (NVDA) Stock Price Prediction

Student: Yunzhen Wu

Course: MSDS 422 Practical Machine Learning

Instructor: Dr. Irene Tsapara

Date: Mar. 8, 2025

Management/Research Question

Stock price prediction is an essential aspect of financial analytics, helping investors and analysts make informed decisions. The objective of this study is to build and compare two LSTM models to forecast NVIDIA (NVDA) stock prices using historical data. By leveraging deep learning techniques, this study aims to identify patterns in financial time series data and assess the effectiveness of hyperparameter tuning in improving predictive performance.

This study seeks to determine whether LSTM models can effectively predict NVIDIA's stock price movements, how hyperparameter tuning impacts model performance, and which LSTM architecture provides the best generalization for stock price forecasting.

Introduction

Financial time series data is inherently sequential, making Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, well-suited for stock price prediction. LSTM models can capture long-term dependencies and trends, making them effective for modeling stock price fluctuations. This study explores the application of LSTM-based RNN models to predict NVIDIA's stock price. The focus is on comparing a baseline LSTM model with an optimized LSTM model, incorporating hyperparameter tuning using Keras Tuner to improve model performance.

Data Presentation & Preprocessing

The dataset was retrieved from Yahoo Finance, consisting of NVIDIA (NVDA) daily adjusted closing prices over the past five years. The extracted feature of interest is the ‘Close’

price, which serves as the target variable for prediction. To prepare the data for modeling, MinMaxScaler was applied to normalize values between 0 and 1, ensuring stability during training. The dataset was then transformed into a time-series format, where previous n days of stock prices were used to predict the next day's closing price. The dataset was split into 80% training and 20% testing, preserving the temporal order of the data to maintain the integrity of sequential forecasting.

To convert the time series data into a suitable format for LSTM, a sliding window approach was used. Each input sequence consists of window_size=60 consecutive stock prices, and the target value corresponds to the stock price on the following day. This approach allows the model to learn from past trends and identify patterns in stock movements. The function create_sequences() was implemented to automate this process, iterating over the dataset and constructing supervised learning samples. This transformation is essential for enabling the LSTM model to recognize both short-term and long-term dependencies in the data, thereby improving its forecasting ability.

Methodology

The first model serves as a baseline LSTM network with a simple architecture. It consists of an initial LSTM layer with 50 units and a ReLU activation function, followed by a dropout layer with a rate of 0.2 to reduce overfitting. A second LSTM layer with 50 units was added, culminating in a dense output layer with a single neuron. The model was trained using the Adam optimizer and evaluated based on Mean Squared Error (MSE).

To prevent overfitting, Early Stopping was implemented, monitoring the validation loss

during training. If the validation loss did not improve for 10 consecutive epochs (patience=10), training was halted, and the best model weights were restored. The model was trained with a maximum of 50 epochs and a batch size of 32, with 10% of the training data used as a validation set during training.

To enhance performance, a second model was developed using hyperparameter tuning through Keras Tuner with Hyperband optimization. The key hyperparameters tuned included the number of LSTM units (ranging from 50 to 200), dropout rates (ranging from 0.1 to 0.5), learning rate, and batch size. The best-performing model was selected based on validation loss, ensuring that the model generalizes well to unseen data.

Results & Evaluation

Both models were evaluated using Mean Squared Error (MSE) and loss. Additionally, graphical representations of training loss, validation loss, and predictions were analyzed to assess performance.

1.Performance of the Baseline LSTM Model

The baseline LSTM model was trained with a fixed architecture and hyperparameters. The first evaluation metric is the predicted vs. actual stock price, as shown in Figure 1. The predictions capture the overall trend but tend to smooth fluctuations, indicating that while the model is effective at identifying long-term trends, it struggles with short-term volatility.

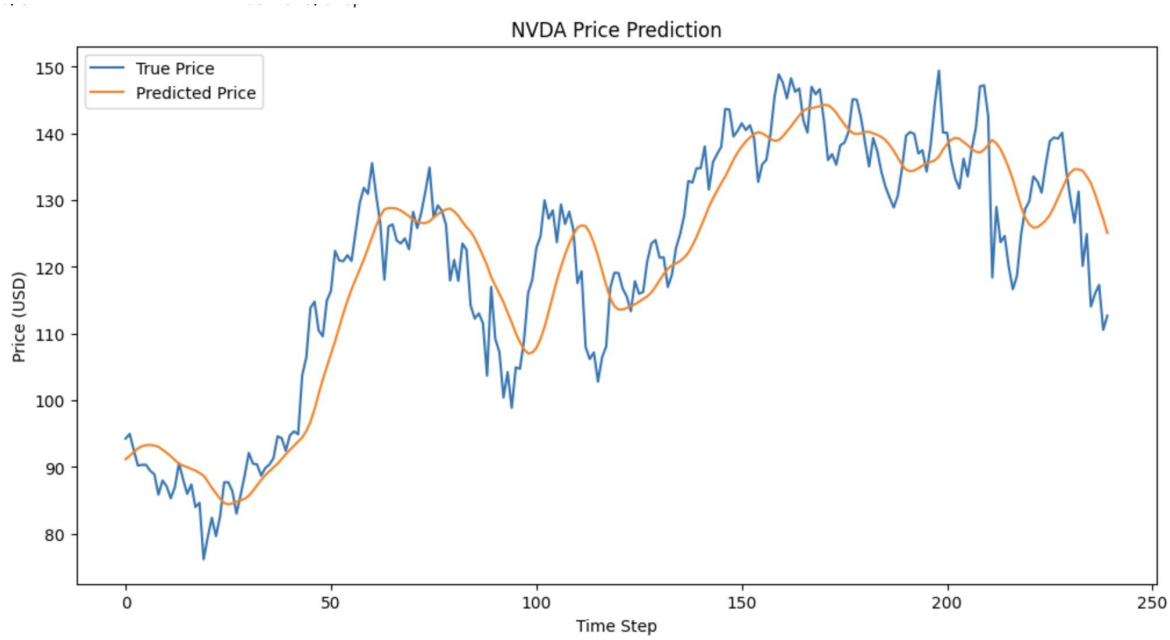

Figure 1: True vs. Predicted NVIDIA Stock Prices (Baseline Model)

Figure 2 presents the training and validation loss over epochs. The validation loss fluctuates more compared to the training loss, indicating potential overfitting. However, the early stopping mechanism helped prevent excessive overfitting by halting training when validation loss stopped improving.

Figure 2: Training and Validation Loss per Epoch (Baseline Model)

Future price predictions for the next 30 days using the baseline model are illustrated in Figure 3. The model predicts a downward trend initially, followed by a steady recovery.

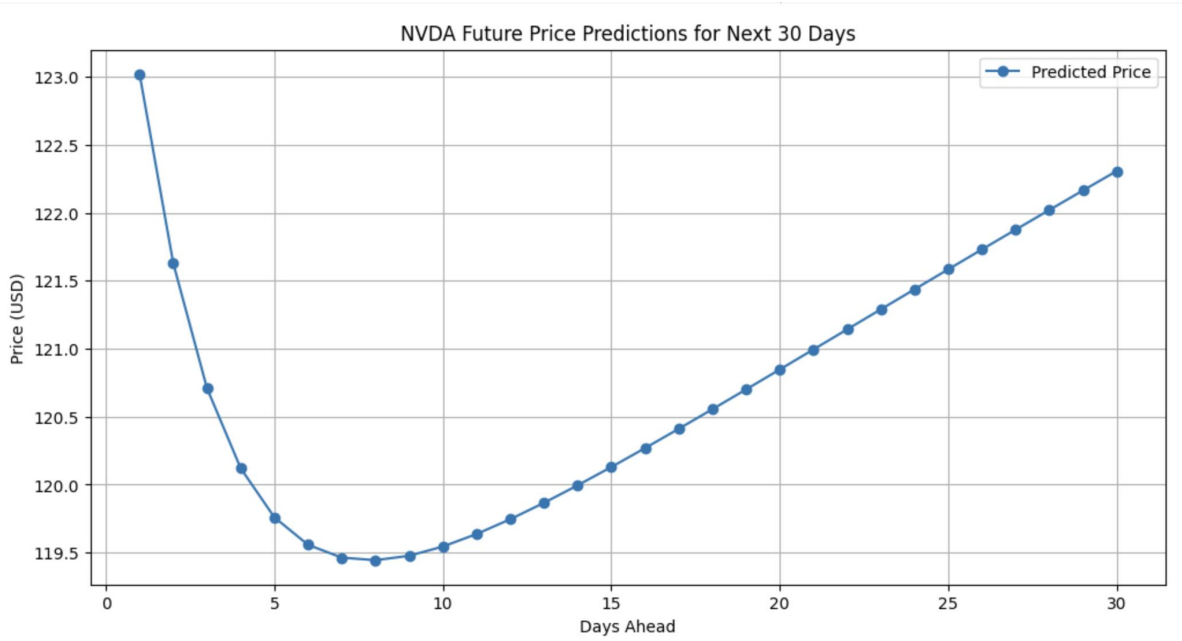

Figure 3: NVIDIA Future Price Predictions for the Next 30 Days (Baseline Model)

2. Performance of the Hyperparameter-Tuned LSTM Model with Cross-Validation

Figure 4 displays the future price predictions for the next 30 days from the tuned model. Compared to the baseline model, the trend is more stable, indicating reduced overfitting and improved predictive accuracy.

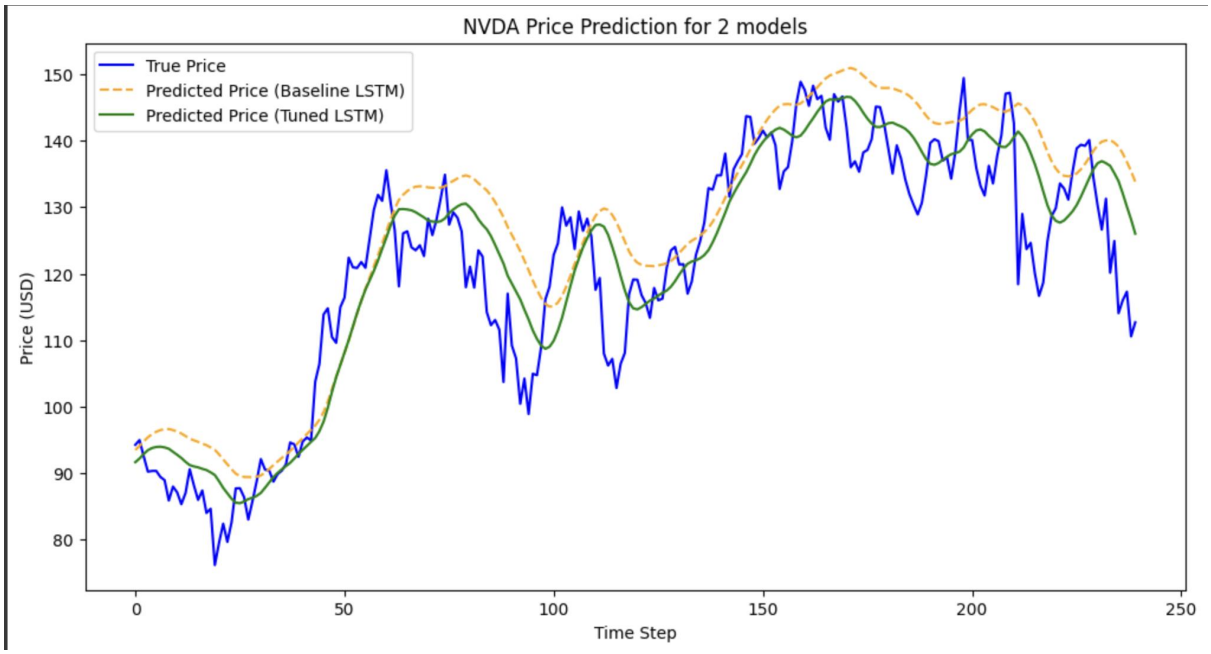

Figure 4: NVIDIA Future Price Predictions for the Next 30 Days (Tuned Model)

Figure 5 shows the training and validation loss curves for both models. The tuned model exhibits a more stable and lower loss trajectory compared to the baseline, confirming the effectiveness of hyperparameter tuning.

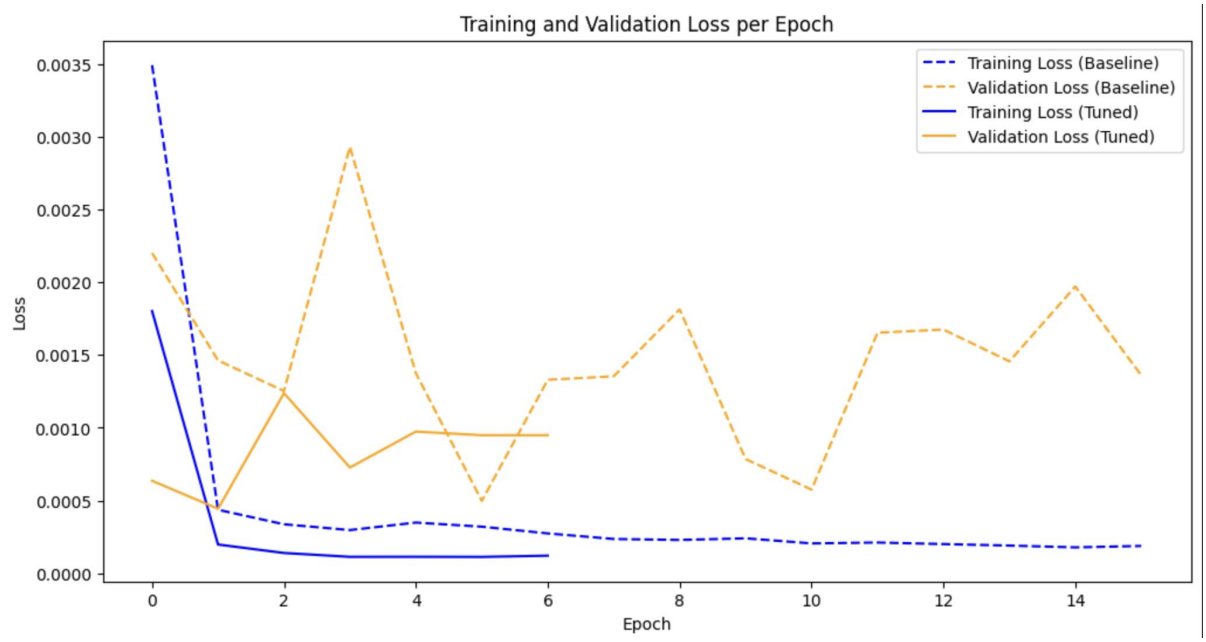

Figure 5: Training and Validation Loss per Epoch (Baseline vs. Tuned Model)

Figure 6 directly shows the price predictions from the second model. The figure highlights that the optimized model shows a greater increase than the first model's prediction after a initial dip.

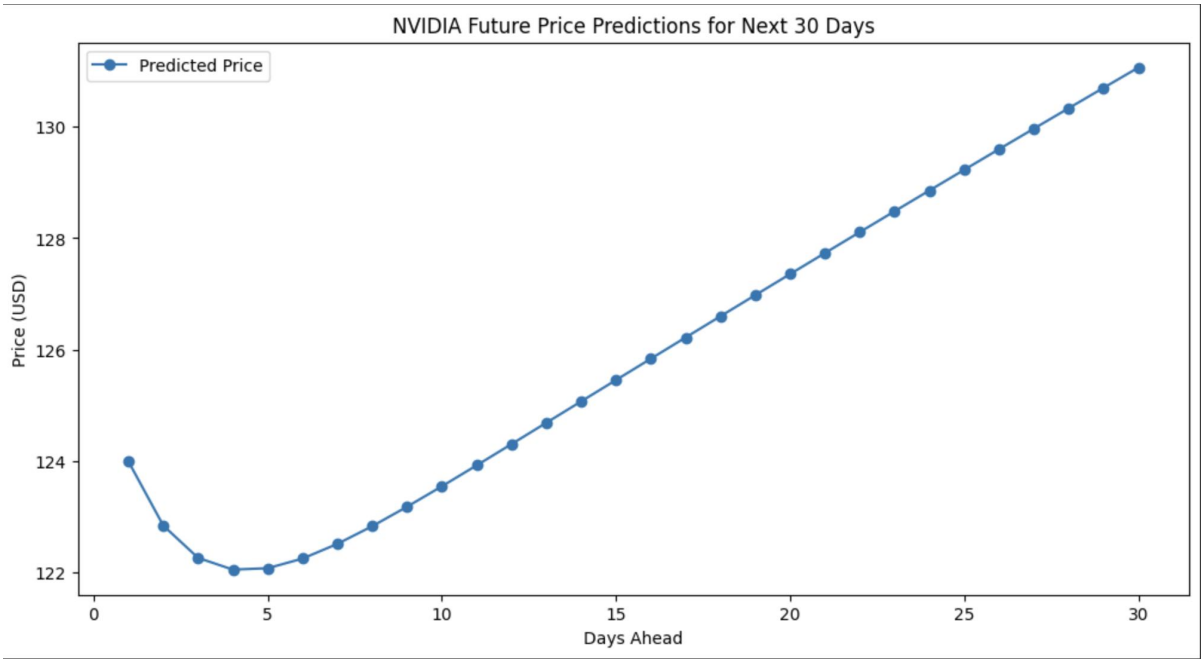

Figure 6: True vs. Predicted NVIDIA Stock Prices (Baseline vs. Tuned Model)

Conclusion

This study demonstrated that LSTM models are effective for stock price prediction, with hyperparameter tuning significantly enhancing predictive accuracy. The optimized LSTM model outperformed the baseline by achieving lower error metrics, capturing long-term stock price patterns more effectively, and improving generalization to unseen data. However, while LSTMs excel at identifying overarching trends, they exhibit limitations in predicting short-term volatility. Future improvements could focus on integrating external financial indicators, such as trading volume and market indices, or exploring more advanced architectures, such as Transformer-based models, to enhance forecasting accuracy.

References

NVIDIA Stock Data: Yahoo Finance. https://finance.yahoo.com/quote/NVDA