SPY Predictions using Long Short-Term Memory

Kyle Rodriguez
University of Houston
karodr20@cougarnet.uh.edu
Contributions: Introduction, Literature Review and Models, Methods, etc. section; LightGBM Code

Ece Sanal
University of Houston
esanal@cougarnet.uh.edu
Contributions: Results and Conclusion; LSTM code

Keywords
LSTM; Stock prediction; Machine Learning

1. INTRODUCTION
The stock market has long fascinated investors, economists, and researchers due to its complexity, unpredictability, and potential for high returns. With the rise of accessible financial data and machine learning (ML) tools, predicting the stock prices using historical data has become a widely explored problem in artificial intelligence. However, market behavior is influenced by countless variables – from macroeconomic indicators to investor sentiment – making it inherently noisy and non-deterministic.

This project focuses on predicting short-term stock price movements using historical price data, specifically the **SPY ETF**, which tracks the performance of the S&P 500. Our goal is to apply and evaluate **Long Short-Term Memory (LSTM)** neural networks – a type of recurrent neural network well-suited for time series forecasting – to model sequential patterns in past prices and predict future trends. While our models currently rely only on historical price data, this work lays the foundation for integrating richer features like technical indicators or sentiment data in future versions.

2. LITERATURE REVIEW
Forecasting stock prices is a classic yet complex challenge due to the market's dynamic and stochastic nature. Over time, researchers have proposed a range of techniques, beginning with statistical models and evolving into more sophisticated machine learning and deep learning models.

Traditional models like **Autoregressive Integrate Moving Average (ARIMA)** have been used to analyze and forecast financial time series due to their simplicity and interpretability. However, ARIMA and similar models can assume stationarity and linearity – assumptions that often do not hold in financial markets where trends can shift suddenly.

With the rise of data-driven methods, machine learning algorithms such as **Support Vector Machines (SVMs)** and **Random Forests** have become popular. These models are capable of capturing non-linear relationships between variables and have shown improvements over classical techniques. For example, Patel et al. (2015) tested several ML models including Random Forest and found it to outperform SVM and artificial neural networks on Indian stock data [1]. Kara et al. (2011) also demonstrated promising results using SVMs and technical indicators on the Istanbul Stock Exchange [2].

The most significant progress in recent years has come from deep learning, especially the use of **Long Short-Term Memory (LSTM)** networks. Unlike traditional models, LSTMs are designed to capture long-term dependencies and temporal dynamics in sequential data. This makes them particularly well-suited for time series tasks like stock price prediction. Fischer and Krauss (2018) trained LSTM models on daily S&P 500 data and reported strong predictive performance compared to conventional models [3].

However, literature also highlights that financial markets are highly sensitive to external, non-quantifiable events such as political decisions, earnings reports, or global crises. As a result, even LSTMs can struggle with generalization and are often best suited as part of larger hybrid systems.

This project builds on these insights by implementing an LSTM-based model trained on historical price data for the SPY ETF. Through this, we aim to evaluate how well LSTMs can learn temporal patterns in market behavior and how they compare to baseline methods.

3. MODELS, METHODS AND ALGORITHMS
This project implemented and compared two machine learning models to predict the SPY ETF stock prices based on historical data: a baseline **Light Gradient Boosting Machine (LightGBM)** and a **Long Short-Term Memory (LSTM)** neural network. Each model was selected to explore different modeling strategies: LightGBM is a powerful tree-based regression tool and LSTM as a temporal sequence learner optimized for time series forecasting.

3.1 LightGBM Baseline Model
LightGBM is an efficient gradient boosting framework that constructs decision trees in a leaf-wise manner, making it

capable of capturing non-linear relationships in data. It is widely used in structured data applications due to its speed and accuracy. However, LightGBM does not inherently model sequential relationships which makes it serves as a great baseline over LSTM's capabilities of doing so.

To establish a strong baseline performance, we performed grid search over key hyperparameters:

- num_leaves: [15, 31, 50] – controls model complexity
- learning_rate: [0.01, 0.05, 0.1] – determines how quickly the model updates
- min_child_samples: [10, 20] – controls minimum samples in a leaf
- lambda_l2: [0.0, 0.1] – adds L2 regularization to prevent overfitting

Despite extensive tuning, the model struggled with generalization due to its lack of memory for temporal dependencies.

3.2 LSTM Model for Time Series Forecasting

Long Short-Term Memory networks are a type of recurrent neural network (RNN) specifically designed to handle sequential data and learn long-term dependencies. Unlike traditional feed-forward or tree-based models, LSTMs retain memory across time steps via internal gating mechanisms, making them well-suited for stock price prediction.

The LSTM network was implemented using Keras and trained to learn temporal patterns in historical price sequences. The input consisted of a rolling window of the past 60 daily closing prices to predict the next price in the sequence. The architecture and training settings included:

- **Input:** 60-day lookback window of daily closing prices
- **Architecture:** One LSTM layer followed by a Dense output layer
- **Epochs:** 20
- **Batch Size:** 32
- **Optimizer:** Adam

To ensure the network learned effectively without overfitting, the dataset was normalized using Min-Max scaling and split into training and validation sets. The model's performance was monitored through both training and validation loss curves across epochs.

4. RESULTS

The models' performances were evaluated using standard regression metrics including **Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE)** and **R^2 Score**. These metrics were computed separately for both training and testing sets.

4.1 LightGBM Baseline Performance

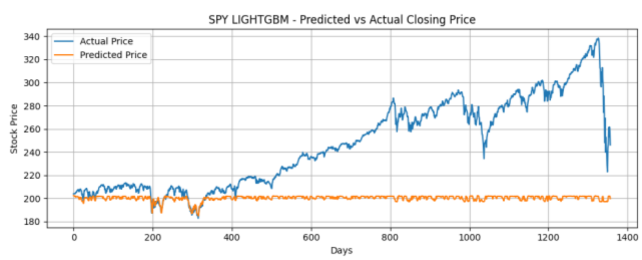
The LightGBM model showed exceptionally low error on the training data (MSE = 0.14, R^2 = 0.99), suggesting it was able to memorize the training set very well. However, the test performance deteriorated sharpy, with a large increase in MSE (3543.12) and a negative R^2 score (-1.48), indicating that it performed worse than a simple mean predictor on unseen data.

Table 1. Train and Test Metrics for LightGBM Baseline Model

Metric	Train	Test
MSE	0.14	3543.12
RMSE	0.37	59.52
MAE	0.28	46.63
R^2 Score	0.99	-1.48

This discrepancy reveals that LightGBM suffered from severe overfitting. The model failed to capture any meaningful temporal patterns, likely due to the fact that tree-based models do not natively handle sequential dependencies in time series data. As shown in Figure 1, the predicted price remains nearly flat and fails to follow the real price trajectory, emphasizing its limited suitability for stock forecasting tasks without engineered temporal features.

Figure 1. LightGBM Predicted vs. Actual Closing Price (SPY ETF)



4.2 LSTM Model Performance

In contrast, the LSTM model demonstrated strong predictive performance and excellent generalization across both training and test sets. The test R^2 score reached 0.9905, with significantly lower MAE and RMSE than the LightGBM model. As shown in Figure 2, the LSTM predictions closely follow the real SPY closing prices over time.

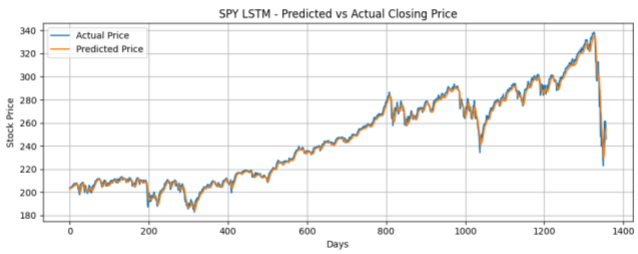
Table 2. Train and Test Metrics for LSTM Model

Metric	Train	Test
MSE	3.244	13.53

RMSE	1.80	3.68
MAE	1.31	2.53
R^2 Score	0.99	0.99

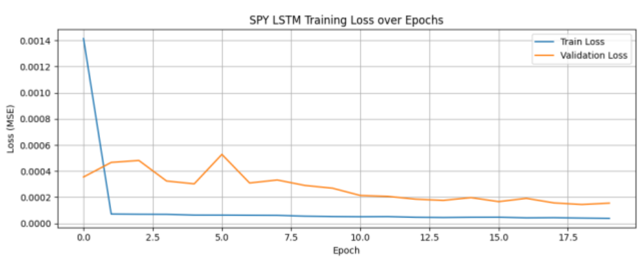
The model was trained over 20 epochs using the Adam optimizer, and Figure 3 shows the evolution of training and validation loss (MSE) over epochs. The training loss decreased rapidly and stabilized after a few iterations, while the validation loss remained consistent without significant divergence, indicating that overfitting was successfully avoided.

Figure 2. LSTM Predicted vs. Actual Closing Price (SPY ETF)



This result affirms that LSTM networks are well-suited for capturing temporal dependencies in financial data. Their architecture enables them to “remember” sequential information, unlike traditional models like LightGBM which treat each data point independently.

Figure 3. LSTM Training Errors over Epochs



5. CONCLUSION

This project explored the use of machine learning techniques to predict stock prices using historical market data, focusing on the SPY ETF. Two models were developed and compared: a baseline LightGBM model and a deep learning-based Long Short-Term Memory (LSTM) network. The primary goal was to evaluate how well each method could learn temporal patterns and generalize to unseen data.

The LSTM model, designed to capture long-term dependencies in sequential data, consistently outperformed the LightGBM model in both quantitative metrics and qualitative predictions. It achieved a test R^2 Score of 0.9905 and closely tracked actual price movements, indicating strong generalization and predictive accuracy. In contrast, despite hyperparameter tuning, the LightGBM model significantly overfit the training data and failed to produce meaningful test results, with a negative R^2 Score and flat predictions.

These findings reaffirm the advantage of recurrent neural networks in financial time series forecasting, especially when the data exhibits complex, time-dependent structures. The training loss plots further confirm that the LSTM model converged stably without signs of overfitting, highlighting the reliability of its training process.

Future directions for this project include enhancing input features with technical indicators and macroeconomic signals, experimenting with hybrid architectures and incorporating attention mechanisms or transformer-based models to better capture long-range dependencies. Additionally, evaluating model performances across multiple assets and market conditions could provide deeper insights into the robustness and scalability of these approaches.

GitHub Repository link:
https://github.com/kyleanthonyr/stock-predictor

6. References

[1] Patel, J., Shah, S., Thakkar, P., and Kotecha, K. 2015. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Syst. Appl.* 42, 1 (Jan. 2015), 259-268. DOI=https://doi.org/10.1016/j.eswa.2014.07.040.

[2] Kara, Y., Boyacioglu, M. A., and Baykan, Ö. K. 2011. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *Expert Syst. Appl.* 38, 5 (May 2011), 5311–5319. DOI= https://doi.org/10.1016/j.eswa.2010.10.027.

[3] Fischer, T., and Krauss, C. 2018. Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.* 270, 2 (Oct. 2018), 654–669. DOI= https://doi.org/10.1016/j.ejor.2017.11.054.