Intraday Stock Price Prediction Using Machine Learning: A Case Study on YFinance Stock Data

Dikshith Reddy Macherla  
*Computer Science Department*  
*Trent University*Durham GTA, Canada  
[dmacherla@trentu.ca](mailto:dmacherla@trentu.ca)

Uchechukwu Obinwanne  
*Computer Science Department*  
*Trent University*Durham GTA, Canada  
[uchechukwuobinwanne@trentu.ca](mailto:uchechukwuobinwanne@trentu.ca)

Wenying Feng  
*Computer Science Department*  
*Trent University*Durham GTA, Canada  
[wfeng@trentu.ca](mailto:wfeng@trentu.ca)

*Abstract*— This paper explores short-term stock price forecasting using high-frequency 2-minute interval data for Apple Inc. (AAPL), focusing on the application and evaluation of three predictive modeling approaches: Linear Regression, XGBoost, and Long Short-Term Memory (LSTM) neural networks. The dataset, collected via Yahoo Finance (YFinance API), was enriched with technical indicators, rolling statistics, and lag features to support predictive learning. The models were evaluated on their ability to predict the next closing price and directional movement. While Linear Regression achieved the highest R² score (0.9778) and lowest MSE (0.0502), XGBoost demonstrated competitive performance and provided valuable insights into feature importance. LSTM, though promising for capturing sequential dependencies, underperformed with a higher MSE (0.2490) and reduced R² (0.8932), likely due to the noisy and highly volatile nature of intraday data. All models showed limited directional accuracy (~52%), highlighting the challenge of predicting micro-movements in high-frequency financial time series. This study concludes with a discussion on the strengths and limitations of each approach and outlines future improvements including feature selection, hybrid modeling, and enhanced data preprocessing to boost intraday forecast reliability.

Keywords— Stock Price Forecasting; High-Frequency Data; Time Series Prediction; Machine Learning; XGBoost; LSTM; Linear Regression; Technical Indicators; Feature Engineering; Intraday Trading; Financial Time Series; Predictive Modeling; Directional Accuracy; Volatility; YFinance API

# Introduction

Predicting short-term price movement can offer significant value in applications like algorithmic trading, portfolio hedging, and intraday risk management. However, existing studies [1, 2] focus on daily or longer timeframes, leaving a gap in the application of deep learning methods to high-frequency (intraday) data. These high-frequency financial time series pose unique challenges — including rapid fluctuations, noise, and low signal-to-noise ratios — that require dedicated investigation[2] . This work contributes to that line of research by evaluating the predictive performance of three machine learning models.

This work contributes to that line of research by evaluating the predictive performance of three machine learning algorithms: Linear Regression[6], XGBoost[6] and Long Short-Term Memory (LSTM)[7] networks on high-frequency Apple Inc. (AAPL) stock price data. The dataset consists of 2-minute interval data which was collected over two months (21st April, 2025 – 20th June, 2025) via the Yahoo Finance API [5]

The choice of a 2-minute interval serves as a valuable proof of concept for applying machine learning to high-frequency stock data in challenging trading environments characterized by high noise and rapid price changes, and testing the limits of predictive models. Furthermore, this work lays the groundwork for improving models, preprocessing techniques, and feature engineering for real-time and intraday trading applications.

To improve predictive accuracy, a comprehensive set of technical indicators such as the Relative Strength Index (RSI), Exponential Moving Average (EMA), Stochastic Oscillator, and momentum; lag-based features, and rolling statistics were engineered and incorporated into the input features. These indicators are commonly used in stock price forecasting studies, especially those combining machine learning with technical analysis. For example:

In Paper[2], they included momentum and volatility measures in their deep learning framework for stock price prediction.

The value of combining LSTM with technical indicators like moving averages to enhance model input features for stock forecasting has been demonstrated in prior work [1].

By integrating these indicators, the project leverages domain knowledge to inform the machine learning models, providing them with features that human traders and traditional strategies have found useful for decades [8].

Recent literature highlights the growing use of machine learning in financial forecasting. Traditional statistical models such as ARIMA are limited in handling non-stationary or non-linear dynamics [3]. In contrast, tree-based models like XGBoost offer superior performance with structured tabular data. Recurrent neural networks, especially LSTM, have shown promise in capturing temporal dependencies in time series. However, as demonstrated in prior research [1, 4], LSTM may underperform when applied to noisy, high-frequency financial data, especially without sufficient filtering or domain-specific tuning. By integrating these modeling techniques with robust feature engineering, this study aims to empirically evaluate their effectiveness and uncover practical insights for real-world trading systems.

The primary goals of this study are:

# To develop and compare different machine learning models for next-step stock price prediction;

# To evaluate performance based on traditional regression metrics (MSE, MAE, R²) and directional accuracy;

# To assess the practicality of each approach in real-time forecasting scenarios, particularly for high-frequency financial data.

# METHODOLOGY

This section outlines the data sources, feature engineering techniques, predictive modeling approaches, and experimental setup used to evaluate the short-term stock price forecasting of Apple Inc. (AAPL) using high-frequency data.

## Data Source and Description

The dataset was retrieved using the Yahoo Finance API (yfinance)[5], covering approximately 60 days (i.e, from 21st April, 2025 to 20th June, 2025) of 2-minute interval trading data for Apple Inc. (AAPL). The raw data includes the following columns: Datetime, Open, High, Low, Close, and Volume. This granular time series data provides a rich foundation for modeling intraday market behavior.

Figure 1 shows the part of dataset and the variables are defined as follows:

* Datetime: The timestamp indicating the specific date and time of the price record.
* Open: The stock price at the start of the interval.
* High: The highest price reached during the interval.
* Low: The lowest price reached during the interval.
* Close: The final price at the end of the interval.
* Volume: The total number of shares traded during the interval.

A screenshot of a computer screen

AI-generated content may be incorrect.

**Figure 1: Sample 2-minute interval trading data for Apple Inc. (AAPL).**

A total of over 6,000 timestamped records were collected, enabling the application of machine learning models to predict the next closing price based on short-term patterns and technical signals.

## Feature Engineering

To enrich the raw dataset, a comprehensive set of technical indicators and statistical features were computed using the ta (technical analysis) Python package. These indicators are commonly used in financial modeling to represent momentum, volatility, and trend-based behavior.

**Technical indicators used:**

Technical indicators such as RSI, MACD, Bollinger Bands, EMA, and others were chosen because they are widely recognized tools in both academic research and financial practice for capturing momentum, trend strength, volatility, and price reversals in time series data. These indicators are designed to help reveal patterns in noisy financial data that raw prices or returns alone might not expose.

* RSI (Relative Strength Index) – Measures the speed and change of price movements, useful for detecting overbought or oversold conditions, which may signal upcoming reversals.
* MACD (Moving Average Convergence Divergence) – Captures trend-following momentum by comparing short and long-term EMAs. It helps identify shifts in strength, direction, and duration of price trends.
* Bollinger Bands – Reflect volatility and price dispersion around a moving average. They provide dynamic support/resistance levels and help spot breakout or mean-reversion opportunities.
* Exponential Moving Average (EMA) – Emphasizes recent price changes, making it responsive to new information while smoothing out noise.
* Stochastic Oscillator, Momentum, ATR (Average true range) – These further assist in gauging market momentum, price speed, and volatility, enriching the feature set to detect various market regimes.
* Rolling statistics: Rolling mean and standard deviation over 5 bars.
* Returns and differentials: One-step percentage return, high-low range, open-close difference.

**Other derived features:**

**Lag features:** Previous close, volume, high, and low (Close\_t-1, Volume\_t-1, etc.)

In this project, lagged features—such as the previous closing price (Close\_t-1), prior volume (Volume\_t-1), and recent highs and lows—were a key part of the feature engineering process. These features represent values from earlier time intervals and provide the model with crucial context about recent market behavior.

The main reason for including lagged features is that, in real-time prediction scenarios, we do not have access to future data at the point of making a prediction. Unlike in a static test dataset, models operating in production must generate forecasts based solely on information available up to the current moment. Lagged features ensure that the model learns to predict the next value based on historical data only, reflecting how real trading systems operate.

Additionally, lagged features allow the models to capture short-term patterns such as momentum, mean reversion, or persistence in price movements—patterns that are particularly important in high-frequency trading environments.

These engineered features aim to capture short-term dependencies, volatility patterns, and technical signals crucial for high-frequency trading models.

## Predictive Models

Three modeling approaches were applied for next-close price forecasting:

* **Linear Regression (LR):** A baseline model to capture linear dependencies between features and the next closing price. It is simple, interpretable, and computationally efficient.
* **XGBoost (Extreme Gradient Boosting):** A robust, scalable tree-based ensemble model. It also provides feature importance scores, helping identify key signals influencing price movement.
* **LSTM (Long Short-Term Memory):** A type of recurrent neural network (RNN) well-suited for sequence learning and capturing temporal dependencies[1]. The model was implemented using TensorFlow/Keras and trained on sequential sliding windows of feature sequences.

## Experimental Design

To evaluate model performance, the dataset was split into training (95%) and testing (5%) sets based on temporal order to preserve the time series structure.

**Preprocessing steps:**

**StandardScaler** was applied to scale inputs for the Linear Regression and XGBoost models and **MinMaxScaler** was used for both features and targets in the LSTM model to improve neural network convergence.

**Performance metrics used:**

* **Root Mean Square Error (RMSE)** – penalizes large errors and measures overall prediction accuracy.
* **Mean Absolute Error (MAE)** – reflects average magnitude of prediction error.
* **R² Score (Coefficient of Determination)** – indicates how well the model explains the variance in the target variable.
* **Directional Accuracy** – percentage of times the predicted and actual price moved in the same direction.

Also graphs and plots were generated to visually compare predicted and actual closing prices, examine residual patterns, and observe directional accuracy trends over time.

# Experimental Results and Analysis

This section summarizes the predictive performance of the three models — Linear Regression, XGBoost, and LSTM — on the out-of-sample test set. Both quantitative metrics and visual analyses were conducted to evaluate their predictive accuracy, directional capability, and practical implications for short-term stock price forecasting.

## Performance Metrics

**Table 1: Predictive performance metrics of Linear Regression, XGBoost, and LSTM models for 2-minute interval AAPL price forecasting on the test set.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **MSE** | **MAE** | **R2 Score** | **Directional Accuracy** |
| **Linear Regression** | 0.0502 | 0.1376 | 0.9778 | 0.5274 |
| **XGBoost** | 0.0588 | 0.1584 | 0.9740 | 0.5152 |
| **LSTM** | 0.2490 | 0.3332 | 0.8932 | 0.5260 |

Linear Regression yielded the lowest MSE and MAE and the highest R² score, making it the most accurate in terms of absolute price level predictions. XGBoost achieved competitive results close to those of Linear Regression and also provided feature importance, allowing insight into which features had the most predictive power. LSTM, despite its capacity for sequential modeling, underperformed due to higher error rates and lower R², suggesting difficulties in capturing short-term fluctuations with limited training data. Directional accuracy was close to 52–53% across all models, barely surpassing random chance, indicating the difficulty of predicting the direction of short-term price movements.

A bar chart with numbers and a bar

AI-generated content may be incorrect.

**Figure 2: Directional accuracy over time for Linear Regression model predictions.**

**A bar code with numbers and symbols

AI-generated content may be incorrect.**

**Figure 3: Directional accuracy over time for XGBoost model predictions.**

**A bar code with text and numbers

AI-generated content may be incorrect.**

**Figure 4: Directional accuracy over time for LSTM model predictions.**

Directional Accuracy Plots:

The directional accuracy plots revealed all models performed only slightly better than random guessing, suggesting that while the price level was well estimated, predicting the short-term up-or-down movements remains a challenge.

## Visual Analysis of Predictions

Both Linear Regression (Figure 5) and XGBoost (Figure 6) closely followed the overall price trend and successfully tracked most price movements. LSTM (Figure 7) showed smoother predicted prices and lagged slightly behind actual movements during abrupt price fluctuations, reducing its responsiveness.

A graph with lines and numbers

AI-generated content may be incorrect.

**Figure 5. Predicted vs actual closing prices for the Linear Regression model.**

A graph with blue and orange lines

AI-generated content may be incorrect.

**Figure 6. Predicted vs actual closing prices for the XGBoost model.**

**A graph with blue and orange lines

AI-generated content may be incorrect.**

**Figure 7. Predicted vs actual closing prices for the LSTM model.**

Residual plots revealed mostly random error terms for the regression and XGBoost models, suggesting these models successfully captured most systematic variations.

The LSTM residuals were larger and showed signs of underfitting to short-term noise.

A graph with blue lines

AI-generated content may be incorrect.

**Figure 8. Residuals over time for the Linear Regression model predictions.**

A graph showing a line

AI-generated content may be incorrect.

**Figure 9. Residuals over time for the XGBoost model predictions.**

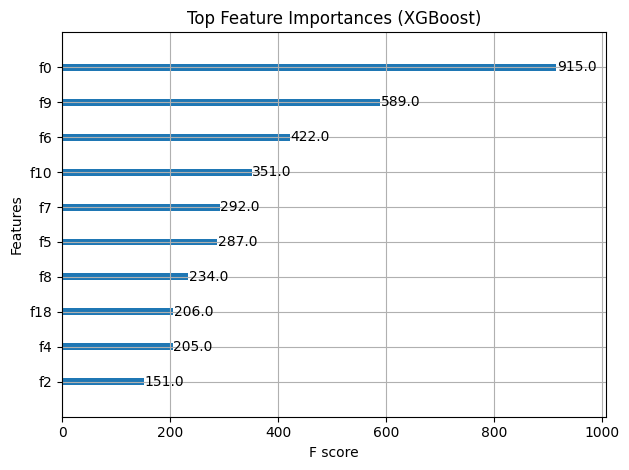
**A graph showing a line

AI-generated content may be incorrect.**

**Figure 10. Residuals over time for the LSTM model predictions.**

## **Feature Importance (XGBoost)**

In figure 11, the feature importance analysis indicated that Close\_t-1 and Close\_return\_1 were the most predictive features. Rolling statistics (that is, the. mean and std) and technical indicators such as RSI and MACD provided significant signals. This supports the hypothesis that short-term price history and momentum-based indicators are most useful for immediate future price predictions..

**Figure 11. Top feature importances from the XGBoost model based on F score.**

**Table 2. XGBoost Feature Mapping to Original Features**

|  |  |
| --- | --- |
| **XGBoost Feature ID** | **Original Feature Name** |
| f0 | Close\_t-1 |
| f1 | Close\_t-2 |
| f2 | Volume\_t-1 |
| f3 | Volume\_t-2 |
| f4 | High\_t-1 |
| f5 | Low\_t-1 |
| f6 | Close\_roll\_mean\_5 |
| f7 | Close\_roll\_std\_5 |
| f8 | Volume\_roll\_mean\_5 |
| f9 | Close\_return\_1 |
| f10 | Close\_diff\_1 |
| f11 | High\_Low\_range |
| f12 | OC\_diff |
| f13 | RSI\_14 |
| f14 | MACD |
| f15 | MACD\_signal |
| f16 | bb\_high |
| f17 | bb\_low |
| f18 | bb\_pct |
| f19 | ATR\_14 |
| f20 | EMA\_10 |

Table 2 is explaining the XGBoost Feature Mapping from Figure 11 to describing their Original Feature including all 20 features

# Discussions

The results highlight some important observations about the feasibility of short-term stock price forecasting on high-frequency data using different modeling techniques:

Despite their simplicity, Linear Regression and XGBoost performed surprisingly well. Linear Regression achieved the lowest MSE and highest R² on the test set, suggesting that most of the short-term price variations can be explained by straightforward relationships between lagged prices and technical indicators. XGBoost also provided competitive accuracy and offered the additional benefit of feature importance ranking.

All models showed directional accuracy close to 52–53%, indicating difficulty predicting the next price move’s sign. This is not surprising given the noisy and stochastic nature of intraday price fluctuations and is a common issue noted in financial time series literature. Even powerful models failed to capture very short-term directional signals, reinforcing the need for alternative or richer data sources. The LSTM model, designed to leverage temporal dependencies, underperformed relative to expectations. This is likely due to:

* High noise and volatility at the 2-minute scale, which reduces the advantage of long-term dependency modeling.
* Limited historical context and relatively small training set.
* Data preprocessing choices that may not fully account for micro-structural noise or regime shifts.

Feature importance rankings from XGBoost suggested that recent price lags and short-term returns had the most predictive value. Technical indicators like RSI and MACD also contributed but were less dominant. This supports the well-known idea that short-horizon price dynamics are mostly driven by price momentum and mean-reversion.

## Implications for Application

While Linear Regression and XGBoost yielded high regression accuracy, directional accuracy was modest across all models. This outcome reflects the inherent noise and randomness in short-term price fluctuations. The challenge of predicting intraday price movements, which are often driven by microstructure noise, news, and order flow dynamics. The LSTM’s underperformance on this task is in line with prior research that long short-term memory models often require more data or richer feature engineering to outperform simpler algorithms on highly volatile time series.

These findings suggest that Linear models and tree-based ensemble methods can be surprisingly competitive even at short horizons. Directional accuracy can potentially be improved by incorporating alternative data sources (e.g. order book data, sentiment) or applying more specialized models for directional classification. LSTMs and other deep learning methods might offer advantages on longer time horizons, where temporal dependencies and trends are more prominent.

# Discussions

This project investigated the predictive performance of three machine learning approaches — Linear Regression, XGBoost, and LSTM — for short-term stock price forecasting using 2-minute interval data for Apple Inc. (AAPL). The findings can be summarized as follows:

* Linear Regression and XGBoost achieved high regression accuracy, with low errors and R² scores close to 0.98, demonstrating that simple and tree-based models can successfully model short-horizon price changes.
* Directional accuracy was modest (~52–53%) for all models, highlighting the substantial difficulty of predicting the direction of short-term price movements, especially at high frequency.
* LSTM, despite its capability for sequence modeling, did not outperform traditional methods due to the noisy, highly volatile data at short intervals.

## Future Work:

Data enhancement, such as incorporating news sentiment, order book data, or macroeconomic variables to provide richer predictive signals. Model improvements, including hybrid architectures (e.g. CNN-LSTM), transformer-based time series models, or ensembling multiple predictive algorithms. Noise-reduction techniques, like wavelet transforms or empirical mode decomposition, to isolate meaningful signal components. Testing on different time scales and across different assets to evaluate model generalizability.

In sum, this project underscores the practical challenge of very short-term price forecasting and the value of combining traditional statistical tools with machine learning techniques. It lays a foundation for further exploration into predictive modeling and feature engineering in high-frequency trading contexts.

##### Acknowledgment

The authors would like to thank Dr. Wenying Feng, Acting Dean of Trent University, for her continuous support, expert guidance, and leadership throughout the duration of this research. Appreciation is also extended to Uchechukwu Obinwanne for coordinating the research efforts, providing technical feedback, and ensuring alignment with academic and formatting standards. The Computer Science Department at Trent University, Durham GTA, is gratefully acknowledged for providing the resources and environment necessary to carry out this project. This research was conducted as part of the 2025 Summer Research Project at Trent University.

The authors also acknowledge the use of the YFinance API for financial data access and the *ta* Python library for technical indicator computation, both of which were instrumental in building the forecasting models.

##### References

1. Y. Yu, "LSTM-Based Time Series Prediction Model: A Case Study with YFinance Stock Data," in *ITM Web of Conferences*, vol. 70, p. 03015, 2025. [Online]. Available: https://doi.org/10.1051/itmconf/20257003015
2. W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS ONE*, vol. 12, no. 7, p. e0180944, 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0180944
3. C. Zhang, N. N. A. Sjarif, and R. Ibrahim, "Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020–2022," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 14, no. 1, p. e1519, 2023. [Online]. Available: https://doi.org/10.1002/widm.1519
4. J. Li and S. J. Qin, "Applying and dissecting LSTM neural networks and regularized learning for dynamic inferential modeling," *Comput. Chem. Eng.*, vol. 175, p. 108264, 2023. [Online]. Available: https://doi.org/10.1016/j.compchemeng.2023.108264
5. Yahoo Finance API, "yfinance," [Online]. Available: https://pypi.org/project/yfinance/. [Accessed: Jun. 26, 2025].
6. F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://scikit-learn.org/stable/
7. TensorFlow Developers, "TensorFlow: End-to-end open-source platform for machine learning," TensorFlow, 2023. [Online]. Available: https://www.tensorflow.org/
8. Zerodha, "Technical analysis," *Varsity by Zerodha*, 2015–2025. [Online]. Available: https://zerodha.com/varsity/module/technical-analysis/

**.**