Assignment-1: Stock Price Forecasting

Eeshan Beohar
DAASE, IIT-Indore

Aim

This research presents a comprehensive analysis of intraday stock price prediction using machine learning and deep learning techniques. We used historical data from the National Stock Exchange (NSE) with 1-minute intervals to develop and compare two distinct predictive modelling approaches: Long Short-Term Memory (LSTM) neural networks and linear regression. The models were designed to forecast stock closing prices at both 3-minute and 15-minute future intervals from any given input time. Our methodology incorporated flexible data preparation techniques that enable prediction for any stock token, date, and time combination within the dataset.

Results demonstrate that both LSTM and linear regression approaches show declining accuracy with 15 minute window. LSTM performs marginally better than linear regression, particularly for shorter prediction horizons. This study establishes a foundation for algorithmic trading strategies based on short-term price predictions while identifying several promising directions for future enhancement.

Introduction

Although the aim of this work or any market research is to predict stock closing prices at future time points (specifically t+3 and t+15 minutes) based on available data at time t. Since, stock market is inherently a random variable, literature has shown that current machine learning models like LSTM are notorious for providing false or baseless highly accurate predictions, which are merely target values offset by some timesteps. Therefore in this work, we instead forecast the percentage change in closing price for a stock. We explore two distinct modeling approaches: a deep learning approach using Long Short-Term Memory (LSTM) networks specifically designed to capture patterns in sequential data, and a statistical approach using linear regression as a baseline comparative model. The LSTM approach allows us to incorporate the temporal dependencies inherent in stock price movements, while linear regression provides a simpler, more interpretable benchmark.

The primary objectives of this study are to: (1) develop a flexible methodology for data preparation that can handle any input stock, date, and time; (2) implement LSTM models for 3-minute and 15-minute prediction horizons; (3) create comparative linear regression models; (4) evaluate and compare the performance of both approaches; (5) identify potential areas for improvement and future research.

Through this comprehensive analysis, we aim to contribute to the growing body of knowledge on machine learning applications in financial forecasting while providing practical insights for intraday trading strategies.

Methodology

0.1 Data Preparation and Processing

The dataset used in this study consisted of intraday stock data from the National Stock Exchange (NSE), organized in pickle (.pkl) files across five folders representing different stocks. Each file contained 1-minute interval data with fields including token identifier, timestamp, price information (open, high, low, close), volume metrics, and order aggressiveness indicators. Additional information linking token identifiers to stock names was provided in a separate mapping file.

Our data preparation process involved several key steps. First, we combined data from multiple .pkl files within each stock folder to create comprehensive datasets for each stock across the available dates. For each date, we also inserted a new 'datetime' pandas column to filter out specific time slices effectively. This is crucial for preparing our training and testing data sets. The data was temporally aligned to ensure that predictions at time t+3 and t+15 minutes corresponded correctly with the input features at datetime t. Steps were taken to ensure no missing values or NaN entered our analysis.

0.2 LSTM Model Implementation

Our deep learning approach is centred on Long Short-Term Memory (LSTM) neural networks, which are specialized recurrent neural networks designed to capture patterns in sequential data. LSTM's ability to selectively remember or 'forget' information through its unique cell architecture makes it particularly suitable for financial time series prediction, where both recent and more distant historical patterns can influence future prices.

The LSTM model architecture consisted of an input layer accepting sequences of historical data, followed by LSTM layers with dropout regularization to prevent overfitting. For both the 3-minute and 15-minute prediction models, we used a sequence length of 10 time-steps (minutes), which provided sufficient historical context without introducing excessive noise or computational burden. Each sequence included all original and engineered features, creating a rich representation of market conditions leading up to the prediction point.

The LSTM layers were configured as follows:

Layer	Output Shape
$LSTM_1$	(None,15,64)
$LSTM_2$	(None,128)
$Dense_1$	(None,128)
$Dense_2$	(None,1)

Table 1: LSTM network structure

We use 'ADAM' optimiser with learning rate = 0.0001 and mean squared error as our metric for minimising loss.

The models were implemented using TensorFlow with the Keras API. Our models are customisable to be trained and tested separately for the 3-minute and 15-minute prediction tasks, for any stock at any date and time. Preliminary experiments indicated that specialized models performed better than a single model attempting to predict multiple horizons. The models were trained for 15 epochs.

We did not implement any early stopping mechanism or dropout regularisation since we deem it necessary for inter-stock predictions, where there might be a significant chance or over/under fitting. Also, the market's highly volatile nature essentially renders all these techniques ineffective.

0.3 Linear Regression Implementation

We implemented linear regression models for both prediction horizons as a baseline statistical approach. While considerably simpler than the LSTM models, linear regression provides an interpretable benchmark and can sometimes perform surprisingly well on financial prediction tasks with well-engineered features.

The linear regression models used the same feature set as the LSTM models. This approach assumes a linear relationship between the input features and the target closing price, which, while a simplification of the complex dynamics of stock prices, can capture some meaningful patterns, particularly in short time horizons with relatively stable market conditions.

We implementedear regression models using scikit-learn, with standard scaling applied to the input features. The models were trained on the same data splits as the LSTM models to ensure fair comparison, although we do not have a validation split here.

0.4 Evaluation Framework

The primary evaluation metrics included Mean Absolute Error (MAE), which provides a measure of average prediction error in the original scale; and R-squared (R2), which indicates the proportion of variance in the target variable explained by the model.

We evaluated model performance across different stocks and market conditions to assess generalization capabilities. Special attention was paid to performance during periods of high volatility versus low volatility, as prediction accuracy often varies significantly based on market conditions.

Results

We present results for a dummy token number (see notebook attached) for the last date. The performance metrics for both LSTM and linear regression models across the two prediction horizons revealed several important insights.

For the 3-minute prediction horizon, the LSTM model achieved an MAE of 0.0011 and R2 of -0.0012 while the linear regression model for the same horizon produced an MAE of 0.0011 and an R2 of -0.0070. **Forecasted percentage Change in Close Price After t+3 Minutes:**
LSTM: [-0.00012178]
Linear Regression: [-1.33984699e-05]
Actual: [0.00014665]

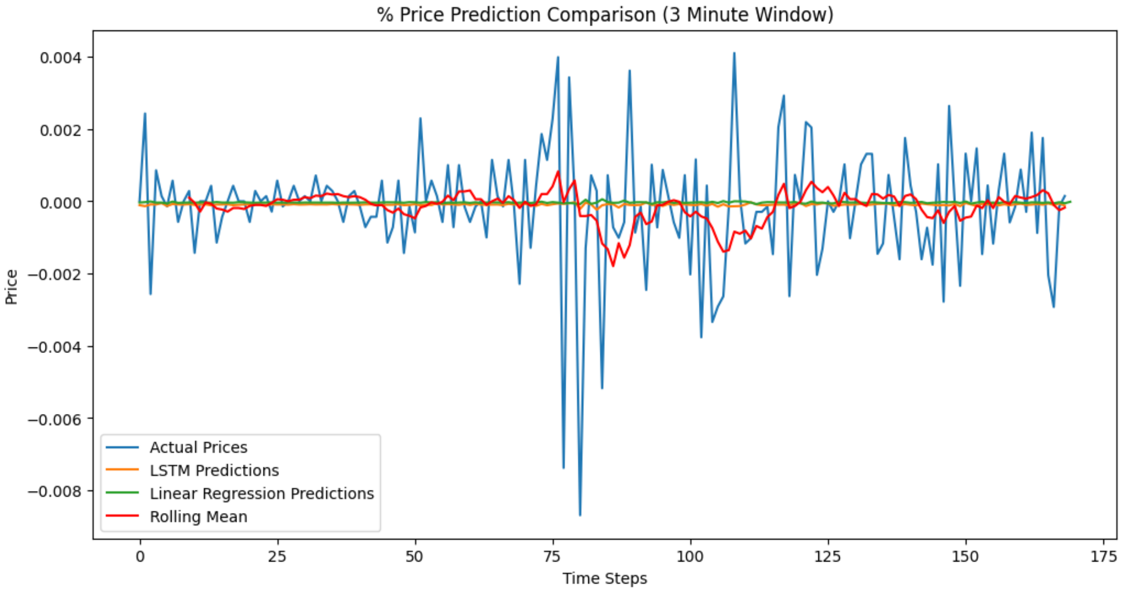

Figure 1: Percentage change prediction for LSTM, Linear regression, moving average (w=20) and actual values with 3-min window.

For the 15-minute prediction horizon, the LSTM model achieved an MAE of 0.0011 and R2 of -0.0132, while the linear regression model for the same horizon produced an MAE of 0.0011 and

an R2 of -0.0073. **Forecasted percentage Change in Close Price After t+15 Minutes:**
LSTM: [-0.00039343]
Linear Regression: [-8.9509935e-06]
Actual: [-0.00088106]

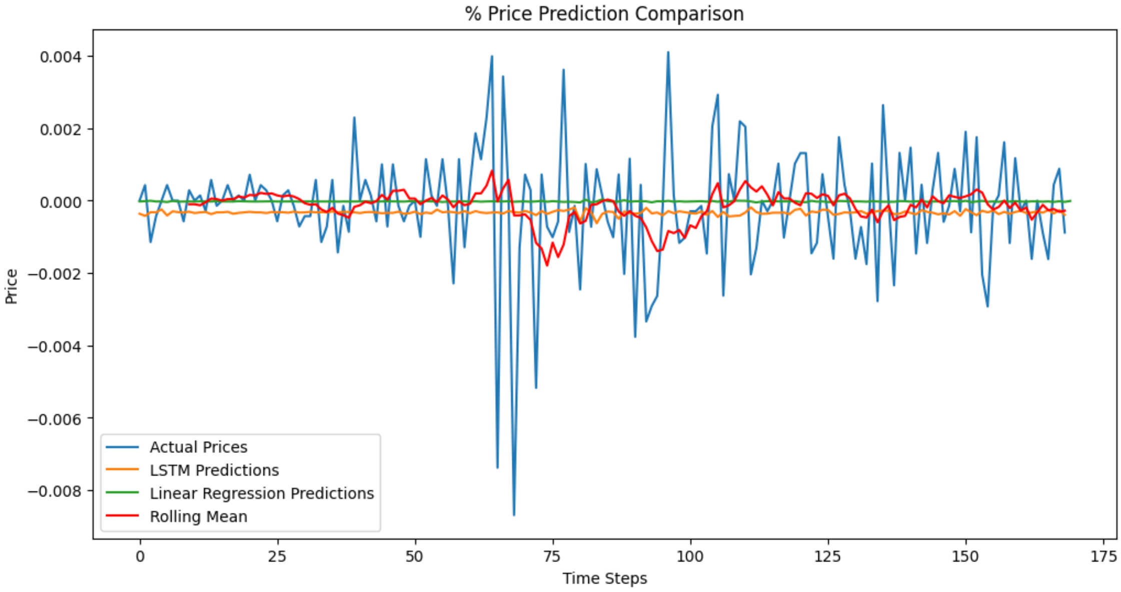

Figure 2: Percentage change prediction for LSTM, Linear regression, moving average (w=20) and actual values with 15-min window.

1 Discussion

Going by the absolute mean squared metric, the results do not favour one technique over another for this dataset. However, we see a significant improvement in LSTM's R2 score for a shorter prediction window, while the same is unchanging for linear regression. These rather underwhelming results may be attributed to a number of reasons:

- The LSTM hyperparameters not being optimised: since we have relatively less data to train or model, the specific layer network becomes a vital hyperparameter that needs to be optimised. An overtly complex structure may lead to overfitting while a simple one may lead to underfitting. Therefore, one should aim to strike a balance between the two. In addition, several regularisation techniques could be employed like dropout.
- Inadequate Feature Engineering: Since we only considered the close price as feature, we may have missed out on some peculiar market behaviours. For instance, large volume orders may regulate future close prices, especially during peak trading times.