Stock Market Prediction System Using LSTM with Technical Indicators as Voters

Vincent
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
vincent064@binus.ac.id

Benecia Jean Saputra
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
benecia.saputra@binus.ac.id

Salsabila Yuka Izzatunnisa
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
salsabila.izzatunnisa@binus.ac.id

Henry Lucky
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
henry.lucky@binus.ac.id

Irene Anindaputri Iswanto
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
irene.iswanto@binus.ac.id

2023 4th International Conference on Artificial Intelligence and Data Sciences (AiDAS) | 979-8-3503-1843-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/AIDAS60501.2023.10284633

***Abstract*—The stock market is a current significant trend in modern life who seek to improve their wealth through investing, thus accurate stock market predictions are essential to maximize gains and minimize losses. This research focuses on enhancing stock market prediction performance using a combination of several technical indicators and artificial neural network technology, specifically Long Short-Term Memory (LSTM). Technical indicators used include Moving Average Convergence Divergence (MACD), Bollinger Bands, Relative Volume (RVOL), Williams %R, Chaikin Money Flow (CMF), Detrended Price Oscillator, Stochastic Oscillator, Volume Weighted Average Price (VWAP), Relative Strength Index (RSI), Moving Average (MA), and TRIX. Each technical indicator generates buy or sell signals that collectively determine whether a stock is worth buying, selling, or holding. Historical stock price data is collected and preprocessed, followed by training and testing the LSTM model with accuracy metrics to evaluate the performance. Using appl dataset from Kaggle, the training and testing generated RMSE values of 0.004 and 0.04 respectively, whereas the standard deviation is 0.04 and 0.16. These results demonstrate high accuracy which corresponds to our expectations given that a threshold of 5 or below represents the minority and a threshold of 6 or above enters the majority voting region. In the future, we plan to incorporate more variables into the LSTM model to achieve greater accuracy and use different stocks to test our findings.**

Keywords—stock prediction, machine learning, technical indicator, LSTM, financial market

I. INTRODUCTION

It is impossible to separate the stock market from contemporary living. Investing in the stock market is a common practice among individuals all around the world. They set aside some of their salaries to try their luck at making more money by investing in the stock market. Traders will purchase stocks with rising predicted values and ignore those with declining values. Thus, there is a pressing need for a reliable forecast of price movements on the stock market in order to maximize capital gain and reduce loss. In addition, stock market prediction is still a challenge as there are factors that might influence it, including economic conditions, corporate news and performance, industry performance, investor sentiment, and sentiment on social media [1]. In this research, we apply artificial neural network technology methods based on artificial intelligence and machine learning algorithms to enhance our performance results for stock market prediction. The main approach used is LSTM with various technical indicators.

Many academic researchers are fascinated by the impressive execution of the Long Short-Term Memory network (LSTM) advantage in predicting the stock market. As a result, Li et al. [2] developed a deep learning technique combined with sentiment analysis, including RNNs, LSTM, and GRU to increase prediction accuracy, and the result achieved up to 67% in MDA.

Another method that is usually used in predicting stock market prices is linear regression. When linear regression is compared to LSTM, the linear regression model turns out to be far better than LSTM showing shallow error in the prediction in technical analysis. In terms of root mean squared error (RMSE), linear regression comes out with 1.82 RMSE while LSTM comes out with 3.42 RMSE [3].

In technical analysis, the analyst may forecast the price of future stocks by examining trends in the past and present stock prices. This is done by regression in a machine learning algorithm to anticipate the stock price trend at the end of a business day based on past price data. However, it only examines the stock's price history and trading choices are made based on mathematical indicators that are derived from the stock price. These indicators include the money flow index (MFI), the moving average convergence/divergence (MACD), and the relative strength index (RSI) [3].

Fundamental analysis is a type of analysis where various fundamentals of a company are considered, such as its balance sheet, micro-economic indicators, consumer behavior, price-earnings ratio, and price-to-book value. It offers a long-term view of the value and prospects of a company. However, in Indonesia, there is an argument that fundamental analysis may not be relevant to use due to the large number of brokers who can manipulate stock prices as they please [4]. Therefore, we focus on implementing technical analysis as a basis for investment decisions making.

Our research aims to fill the gap in the literature where there is almost no research that uses LSTM as a voting system, whether the brokers should buy the increasing stock market values. This is done by implementing various technical indicators such as MA, RSI, MACD, etc. into the LSTM model. Each indicator would be used to generate buy or sell signals for the stocks under consideration. The goal is to evaluate the performance of the merged technical indicators compared to using individual indicators separately. This would involve analyzing metrics such as accuracy, precision, or recall, to determine the effectiveness. This study considers the volume of shares traded data at a certain moment, which might be daily, weekly, monthly, or even yearly. By establishing our niche in this area, we can contribute to the broader conversation on the application of machine learning in finance and provide a new angle for informed decision-making in the stock market.

Our approach involves collecting historical stock price data for a selected time frame as well as calculating the technical indicators. We will then use the preprocessed dataset to train and test the LSTM model which is used to make predictions on both the training and testing data. The predictions are inverse transformed to obtain the actual stock price values and will be evaluated by the evaluation metrics. The data needed in this experiment are stock price datasets to calculate and for the machine to learn and predict the next trend and price the data are gathered from finance websites such as Kaggle, Yahoo Finance, or Stockbit.

II. Literature Review

Technical analysis models have their own limitations, the models are based on stock price history and can not consider non-technical factors such as world events, news, or change in a company's structure. Machine learning and deep learning algorithms have gained significant attention recently in the finance industry, notably for stock market forecasting. That being said, Gustav Gerholm and Adam Lindberg [5] did an experiment to compare the accuracy of different machine learning models such as Support Vector Machine (SVM), LSTM, Random Forest (RF). They found that prediction capability increases as the time horizon increases. Also, they concluded that SVM got the highest accuracy and F-score on every prediction period. These results contrasted with Nelson, Pereira, and Oliveira's research where they found LSTM has the best accuracy. But, it is worth noting that they did the research on a 15-minute time step, this suggests that LSTM models perform better with more information.

Y. Kara et al. [6] utilized Artificial Neural Networks (ANN) and SVM to forecast stock price movements in the Istanbul Stock Exchange, employing two prediction models and comparing their effectiveness. The ANN model delivered an average prediction performance of 75.74%, while the SVM model was 71.52%. A unique comparison of machine learning techniques was done by Dash and Dash [7], they compared the performance of clustering with extended fuzzy logic and artificial neural networks (CEFLANN) with KNN, SVM, and DT, and found that CEFLANN provided the highest percentage of profit. The authors used a financial dataset and applied the algorithm for stock price prediction which aims to maximize profit. Referring to the study, CEFLANN achieved the highest profit percentage of 33.64%, followed by SVM with 23.28%, KNN with 17.35%, and DT with 12.18%. CEFLANN also proved that it outperformed the other algorithms in terms of accuracy and convergence speed.

Henrique et al. [8] studied Support Vector Regression (SVR). The results of the study indicate that the SVR algorithm has a better predictive capability for stock price prediction when compared to other methods. In particular, the authors found that using up to the minute data with an SVR algorithm led to better prediction accuracy compared to using daily data. Ayala et al. tested the performance of the Linear Model, ANN, RF, and SVR to select the most suitable machine learning technique. The LM and ANN algorithms fared the best when used to examine stocks using a method that combines technical indicators with machine learning.

On the other hand, other authors claim that the current popular model method, LSTM, performed better in predicting the stock market. For instance, in 2017, Nelson et al. [9] argued to use the LSTM model with some technical analysis indicators for stock prediction compared to SVM, RF, and multi-layer perceptron (MLP). The results demonstrated that LSTM provides fewer risks and high accuracy. Taking a closer look at a multi-layer perceptron and a bidirectional LSTM neural network, Chen et al. [10] proposed a unique hybrid deep learning model. The metrics such as MAE, MSE, MSLE, MedAE, and R2 of this model were the best, and the AM was particularly strong around stock price forecasting.

Enhancing the LSTM with another hybrid model, Li et al. [11] proposed a novel deep neural network (DP-LSTM) for stock price prediction, which integrates the news articles as hidden information and combines many news sources using the differential privacy (DP) mechanism, as well as ARMA sentiment, LSTM, and VADER model. Results of experiments using the S&P 500 stocks downloaded from Yahoo Finance demonstrate the proposed model is accurate and robust in performance, particularly for S&P 500 index which captures the market's overall trend. The S&P prediction results indicate that the DP method can significantly improve robustness and accuracy. The dataset used for experiments of this proposed model was a combination of historical data stocks and news articles. The news articles are usually taken from official financial domain websites such as CNBC, Reuters, WSJ, and Fortune.

In 2021, Shujia Liu [12] proposed an end-to-end LSTM model and fully connected layers with two data augmentation techniques: adding random noise to data fields and erasing partial information from training examples. It aims to predict long-term stock based on historical financial statements, and as a result, it shows that the techniques are powerful enough in training models on structured data.

Based on our research, we propose an LSTM model to forecast price movements utilizing an input that is not text-based, instead, we intend to use a wide range of technical indicators. With these indicators that are commonly used in investment strategies, we plan to make it a layer that needs to be overcome to determine the decision-making of buying or selling stocks. Therefore, the end results would be a signal to buy or sell the stocks.

III. Methodology

In this paper, refining the stock prices includes major steps as shown in Fig. 1. such as gathering the datasets, pre-processing the dataset, training and testing the proposed models with technical indicators, and comparing the result.

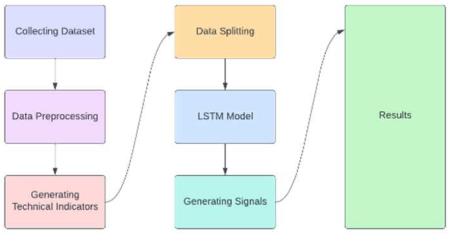


Fig. 1. Architecture for The Proposed Method

A. Dataset

Collecting data is crucial in machine learning as it is the key to making a good learning system. The data used in this research paper was obtained from a finance website, Kaggle. It specifically is an online community platform where data scientist and machine learning enthusiasts may connect to participate in a contest, or access and gives out resources supported with advanced tools. These resources may be code or datasets that can be written and shared. We chose the dataset of Apple with dates ranging from September 1984 up to November 2017. Table I below displayed an illustration of how the dataset appeared in spreadsheets. The 'High, 'Low' refers to the highest and lowest price that a stock reached within a particular period, whereas 'Open', 'Close' refers to the opening and closing price of the stock at the beginning and end of the trading day, respectively. The 'Volume' is the total number of shares traded.

TABLE I. EXAMPLE OF STOCK MARKET DATASET

Date	Open ($)	High ($)	Low ($)	Close ($)	Volume (Shares/Unit)
2017-11-07	173.29	174.51	173.29	174.18	24424877
2017-11-08	174.03	175.61	173.71	175.61	24451166
2017-11-09	174.48	175.46	172.52	175.25	2953086
2017-11-10	175.11	175.38	174.27	174.67	25130494

B. Data Preprocessing

During pre-processing, the data is scaled using the Min-Max scaler technique with the aim to transform the values into a specific featured range which in this case is between 0 and 1. The Min-Max scalar technique was instantiated from the scikit-learn library, notably a widely used Python machine learning library or toolkit. This normalization process guarantees that all data points are on a similar scale and eliminates any potential bias that could develop due to the magnitude of the values.

The data being scaled is focused on the 'Close' column of the appl dataset, which represents the closing prices of the financial instrument under examination. As the range set to 0 to 1 previously, the 'Close' column will have a minimum value mapped to 0 and the maximum value mapped to 1. Between these two extremes, all other values will be scaled correspondingly.

By using the 'fit_transform' method of the scaler object, we applied the scaling transformation in accordance with the set range. Then, the 'Close' values were extracted from the dataset and reshaped into a two-dimensional array with a single column. It is necessary to reshape the extracted values because the scaler in scikit-learn expects a specific input shape for the data it processes. It requires a two-dimensional input where the number of rows represents the samples or instances, and the number of columns represents the features. In this case, each price value can be viewed as a separate sample. Therefore, extracted values must be reshaped into a single column, which denotes a single feature. The process of reshaping helps maintain consistency when working with various datasets that might have multiple features. Following all the processes described, the transformed data was then obtained and stored in a variable called 'scaled_data'.

C. Calculating Technical Indicators

Technical indicators that we have used for our research are discussed in this section. Each of the technical indicators would be used to decide the buy or sell signals for the data.

1) Moving Average Convergence Divergence (MACD)

$$EMA = p_i \frac{2}{n+1} + EMA_{i-1} \cdot 1 - \frac{2}{n+1} \tag{1}$$

Where p equals to asset price for the period, i equals to the current period, and n is number of data considered for the calculation of the moving average [13].

2) Bollinger Bands

Bollinger Bands are volatility bands that surround a moving average. They adjust automatically based on changes in market volatility. The bands widen during high volatility and narrow during low volatility. Traders use Bollinger Bands to identify trend strength and potential reversal patterns like M-Tops and W-Bottoms. The formula is,

$$\textit{Middle Band} = \textit{20-day SMA} \tag{2}$$

$$\textit{Upper Band} = \textit{20-day SMA} + (\textit{20 SD of price} * 2) \tag{3}$$

$$\textit{Lower Band} = \textit{20-day SMA} - (\textit{20 SD of price} * 2) \tag{4}$$

Simple Moving Average or *SMA* is found by calculating the average price of a security over a distinct number of periods. *SD* or Standard Deviation purpose is to measure how spread out the data is, how far for each value is from the mean.

3) Relative Volume (RVOL)

The Relative Volume (RVOL) indicator contrasts each price bar's volume data with the volume average over a predetermined number of prior bars. The formula for Relative Volume (RVOL) is,

$$RVOL = \frac{current\ volume}{average\ volume\ under\ the\ look{-}back\ period} \tag{5}$$

4) Williams %R

$$\%R = \frac{(H14-C)}{(H14-L14)} * -100 \tag{6}$$

Where C equals the current closing price, $H14$ equals to highest high over the past 14 days, and $L14$ where lowest low over the past 14 days.

5) Chaikin Money Flow (CMF)

Chaikin volume measures the amount of money flow volume over a specific look back period. Money flow volume forms the basis for the accumulation distribution line.

$$MFM = \frac{[(C-L)-(H-C)]}{H-L} \tag{7}$$

$$MFV = MFM * V \tag{8}$$

$$CMF_i = \frac{MFV_i}{\sum V} \tag{9}$$

Where *MFM* stands for money flow multiplier, *C* is the close price, *L* is the low price, *H* is the high price, *V* is volume, *MFV* stands for money flow volume, and *CMF* stands for the Chaikin money flow.

6) Detrended Price Oscillator (DPO)

$$DPO_x = \left(\frac{x}{2} + 1\right) - SMA_x \tag{10}$$

Where the *x* refers to the number of periods used, *SMA* refers to simple moving average. The DPO is equal to price of *x*/2 +1 ago minus the *x*-day simple moving average.

7) Stochastic Oscillator

$$\%K = 100 * \frac{(C-L14)}{(H14-L14)} \tag{11}$$

Stochastic Oscillator tracks the momentum of stock price. Before the price changes, the momentum changes first as a rule. *H14* and *L14* are the highest high and lowest low over the past 14 days. *C* is the current closing price.

8) Volume Weighted Average Price (VWAP)

Volume Weighted Average Price (VWAP) is calculated by dividing the cumulative dollar value of all trading periods by the total trading volume for the current day. The cumulative total is obtained by summing the products of volume and the typical price (average of high, low, and close prices) for each trading period, and then dividing it by the cumulative volume. The formula for cumulative total is,

$$Cumulative\ Total = \frac{Cumulative\ (Volume \times Typical\ Price)}{Cumulative\ (Volume)} \tag{12}$$

9) Relative Strength Index (RSI)

$$RSI = 100 - \frac{100}{1+RS} \tag{13}$$

$$RS = Average\ Gain\ /\ Average\ Loss \tag{14}$$

The basic components of RSI are RSI-RS, average gain, and average loss. The calculation is based on 14 days; the default J. Welles Wilder proposes in his book [14].

10) Moving Average (MA)

All types of Moving Averages are calculated by taking an average of a distinct number of prior data points, but each of them weighs those data points differently. The simple moving average (SMA) and the exponential moving average (EMA) are the two most used moving averages. While exponential moving averages (EMAs) give more weight to current prices, simple moving averages (SMAs) simply average values across the given timeframe.

11) TRIX

TRIX is an EMA of an EMA of an EMA. It is the 1-period percentage rate-of-change for a triple-smoothed exponential moving average. For example, there needs to be a single-smoothed, double-smoothed, and triple-smoothed EMA first to break down the 15-period TRIX.

D. LSTM with Technical Indicators Voting System

```
Overall Signal:
    Set the number of required "Buy" signals.
    Count the number of "Buy" signals from the individual signal's columns (e.g., 'RSI_Signal',
    'MACD_Signal', etc.).
    If the count is greater than or equal to the required number of "Buy" signals, assign 'Buy' into'Overall'.
    Otherwise, assign 'Sell' to 'Overall'.
```

Fig. 2. Pseudocode of The System Model Overall Signal

LSTM is a type of recurrent neural network that can remember all the output from the previous node. Here, the LSTM model takes historical data and the generated technical indicators. Fig. 2. shows the pseudocode of our system model. The input data is preprocessed, which may involve scaling or normalization, to ensure compatibility with the LSTM model. The LSTM architecture consists of multiple LSTM layers followed by dropout layers to prevent overfitting. The model is then compiled and trained with 20 epochs and 32 batch sizes. Epoch refers to the complete pass through the dataset and batch size is a hyperparameter that determines how many samples are processed together before updating the model [15].

The predictions generated by the LSTM model are combined with the signals to make trading decisions using a voting system. The voting system considers both the direction of the LSTM prediction and the signals generated by the technical indicators. The specific rules and algorithms used in the voting system are that when most technical indicators signal to buy or sell and are supported by LSTM prediction, then it will have good confidence in buying or selling the stock.

For example, if the LSTM model predicts a Buy signal and the majority of the technical indicators also generate a Buy signal, the system would assign a higher confidence level to the Buy decision. On the other hand, conflicting signals might lead to a lower confidence level or a neutral decision.

After the model training is completed, we will then generate 11 signals based on the technical indicators to act as our voters. The signals are generated based on specific conditions for each indicator. For instance, the RSI signal is determined by comparing the RSI value to threshold values of 70 (indicating overbought) and 30 (indicating oversold). Similarly, the MACD signal is based on whether the MACD value is positive or negative. Another example is MA which will give a signal buy if the close price is higher than the MA values.

By generating these signals, we can then create an overall signal by analyzing if the total number of Buy signals generated by these 11 technical indicators exceed the threshold that we set. For example, if the threshold for the overall signal is set to 7, the overall signal will be generated as "Buy" if there are at least 7 buy signals among the technical indicators.

E. System Evaluation

```
Accuracy:
    Initialize variables:
        Set 'correct_predictions' to 0.
        Set 'total' to 0.
Set 'i' as the loop variable.
Within the loop:
    Get the 'current_prediction' from 'test_predictions' at index 'i'.
    Get the 'current_close' from the 'Close' column of 'data' at the corresponding index.
    Get the 'next_close' from the 'Close' column of 'data' at the next index after 'i'.
    Get the 'date' from the 'Date' column of 'data' at the corresponding index.
    Get the 'overall_signal' from the 'Overall' column of 'data' at the corresponding index.
    Check if 'overall_signal' is 'Buy' and 'current_prediction' is greater than 'current_close':
        If 'next_close' is greater than 'current_close', increment 'correct_predictions' by 1.
        Increment 'total' by 1.
    Check if 'overall_signal' is 'Sell' and 'current_prediction' is less than 'current_close':

        If 'next_close' is less than 'current_close', increment 'correct_predictions' by 1.
        Increment 'total' by 1.
Calculate the accuracy:
    Divide 'correct_predictions' by the length of 'test_predictions' and assign the result to 'accuracy'.
```

Fig. 3. Pseudocode of The System Model Accuracy

The accuracy, shown in Fig. 3. is calculated by analyzing the next close price with the LSTM prediction and overall signal, for example, if the overall signal says buy, the next predicted price is higher and the next close price is up, we will mark it as a correct prediction and vice versa. Accuracy is calculated by dividing the number of correct predictions by the total predictions made.

IV. RESULT

In this section, the results obtained from the performance of the LSTM model for time series prediction, as well as the accuracy of the generated trading signals, are evaluated and analyzed.

A. Evaluation

TABLE II. EVALUATION OF LSTM

	RMSE	Standard Deviation
Train	0.003	0.0845
Test	0.025	0.16154

From Table II, it can be observed that after running the model 10 times, the RMSE predictions in the training set differ from the actual values by approximately 0.003. The RMSE for the test set suggests that the predictions in the test set have an average difference of 0.025. The test set has a lower value of RMSE compared to the training set, indicating that the model's predictions are relatively accurate.

Standard deviation measures the variability of the values in a dataset. The standard deviation for the training and the test set is 0.0845 and 0.16154. This indicates that the values or data points are more spread out in the training set since it has a higher standard deviation.

B. Signal Performance

TABLE III. RESULTS OF THE EXPERIMENT

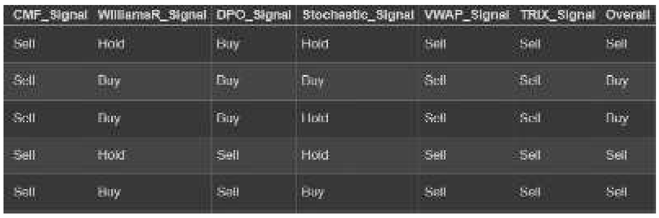
	Number of Triggered Data	Accuracy
Pure LSTM Model without Signals	1651 data	21.2%
LSTM with 4 Buy Signals	575 data	16.2%
LSTM with 5 Buy Signals	658 data	20.9%
LSTM with 6 Buy Signals	1061 data	33.7%
LSTM with 7 Buy Signals	1198 data	37.4%
LSTM with 8 Buy Signals	1192 data	37.7%

The results shown in TABLE III are the result of our 10 experiments with the threshold. It is found that the pure LSTM model without signals has a bigger accuracy compared to the LSTM model combined with only 4 or 5 signals. However, when the threshold is set to 6 or more buy signals, the accuracy is higher than in the pure LSTM model. This outcome aligns with our expectations as using a threshold of 5 or below represents the minority region, while a threshold of 6 and above enters the majority voting region.

By setting the threshold to capture a majority consensus, we can enhance the accuracy of our predictions. It's worth noting that incorporating more signals may introduce additional noise or conflicting indicators, potentially leading to a decrease in accuracy.

Index	Date	Close	RSI_Signal	MACD_Signal	BB_Signal	MA_Signal	RVOL_Signal
43	1984-11-07	0.41111	Hold	Sell	Hold	Buy	Sell
44	1984-11-08	0.39443	Hold	Sell	Hold	Buy	Sell
45	1984-11-09	0.37138	Hold	Sell	Buy	Buy	Sell
46	1984-11-12	0.38419	Hold	Sell	Hold	Buy	Sell
47	1984-11-13	0.37522	Hold	Sell	Hold	Buy	Sell

Fig. 4. Example of Generated Signals Output 1



CMF_Signal	WilliamsR_Signal	DPO_Signal	Stochastic_Signal	VWAP_Signal	TRIX_Signal	Overall
Sell	Hold	Buy	Hold	Sell	Sell	Sell
Sell	Buy	Buy	Buy	Sell	Sell	Buy
Sell	Buy	Buy	Hold	Sell	Sell	Buy
Sell	Hold	Sell	Hold	Sell	Sell	Sell
Sell	Buy	Sell	Buy	Sell	Sell	Sell

Fig. 5. Example of Generated Signals Output 2

Fig. 4. and Fig. 5. displayed an example of generated signals output in which there are a total of 11 technical indicator signals along with the date, close prices, and overall signals. Taking index 45 as a case in point, 2 out of 11 generate Hold signals, 5 Sell signals, and 4 Buy signals. The overall signal of the index 45 would be Buy.

C. Technical Performance

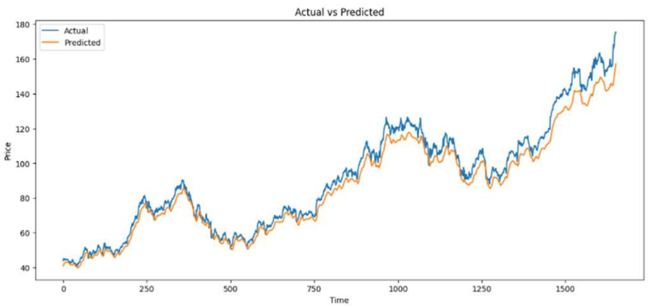


Fig. 6. Actual vs Predicted Price Visualization

In the technical aspects, the LSTM model was trained on the training dataset gathered from Kaggle. The dataset includes a bunch of indicators such as MA, RSI, and BB. After the training phase, the model is used to forecast the training and testing performance. To evaluate the model, we will use RMSE and standard deviation to be our metrics for the regression task. The obtained RMSE values for the training and testing datasets were 0.004 and 0.04 respectively. The train and test standard deviation obtained from the model are 0,.4 and 0.16 respectively. These metrics suggest that the model is performing with a high level of precision. Fig. 6. visualizes the line chart of actual and predicted prices in a particular period as proof of the preciseness. Both lines are close to each other without a huge significant difference which concluded that it is highly precise.

V. Discussion

The results of the tests highlight the influence of considering various signals and establishing a sufficient threshold to increase the precision of the LSTM model for time series forecasting and the improvement of trading signals. The conduct of the pure LSTM model alone beats that of the LSTM model when fill out with only 4 or 5 signals, highlighting the importance of including more signals to make more accurate prediction.

More signals should be used, but it's crucial to use caution because they may bring noise or contradicting data, which could reduce accuracy. To ensure the accuracy of the trading signals created, thorough review and signal selection is therefore required.

The low RMSE values for both the training and testing datasets indicate that the LSTM model performs with a high level of accuracy in terms of technical performance. The low standard deviations were found to support the model's consistency and predictability further.

Overall, the results indicate that the LSTM model can successfully predict time series data and make precise trading signals when set up properly with the right threshold and the right number of signals. Future studies can focus on improving the signal selection procedure, investigating complex thresholding approaches, and assessing the model's performance on actual trade data to determine its practical application.

VI. Conclusion

In conclusion, taking into account numerous signals and setting an appropriate threshold considerably improve the LSTM model's accuracy for generating trading signals and time series forecasting. Accuracy is improved by incorporating adequate indicators, but caution must be taken to prevent noise or contradictory data. Low RMSE values and tiny standard deviations show the LSTM model's great precision. Overall, the findings displayed how well-configured, the LSTM model can produce precise trading signals. Future studies can fixate on improving signal selection and assessing the model's effectiveness in the real trading settings.

References

[1] R. M. I. Kusuma, T.-T. Ho, W.-C. Kao, Y.-Y. Ou, and K.-L. Hua, "Using Deep Learning Neural Networks and Candlestick Chart Representation to Predict Stock Market," Feb. 2019, [Online]. Available: http://arxiv.org/abs/1903.12258.

[2] Y. Li and Y. Pan, "A novel ensemble deep learning model for stock prediction based on stock prices and news," Int J Data Sci Anal, vol. 13, no. 2, 2022, doi: 10.1007/s41060-021-00279-9.

[3] S. Mokhtari, K. K. Yen, and J. Liu, "Effectiveness of Artificial Intelligence in Stock Market Prediction based on Machine Learning," Int J Comput Appl, vol. 183, no. 7, 2021, doi: 10.5120/ijca2021921347.

[4] A. Raditya and E. A. Sumirat, "Juli 2022 (406-417) Fundamental, Technical and Broker Summary Analysis in Investing in Indonesian Stock Market between," vol. 2, no. 3, p. 406, 2016, doi: 10.36418/comserva.v2i3.2.

[5] G. Gerholm and A. Lindberg, "Comparison of machine learning models for market predictions with different time horizons."

[6] Y. Kara, M. Acar Boyacioglu, and Ö. K. Baykan, "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, vol. 38, no. 5, 2011, doi: 10.1016/j.eswa.2010.10.027.

[7] R. Dash and P. K. Dash, "A hybrid stock trading framework integrating technical analysis with machine learning techniques," Journal of Finance and Data Science, vol. 2, no. 1, 2016, doi: 10.1016/j.jfds.2016.03.002.

[8] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Stock price prediction using support vector regression on daily and up to the minute prices," Journal of Finance and Data Science, vol. 4, no. 3, 2018, doi: 10.1016/j.jfds.2018.04.003.

[9] D. M. Q. Nelson, A. C. M. Pereira, and R. A. De Oliveira, "Stock market's price movement prediction with LSTM neural networks," in Proceedings of the International Joint Conference on Neural Networks, 2017. doi: 10.1109/IJCNN.2017.7966019.

[10] Q. Chen, W. Zhang, and Y. Lou, "Forecasting Stock Prices Using a Hybrid Deep Learning Model Integrating Attention Mechanism, Multi-Layer Perceptron, and Bidirectional Long-Short Term Memory Neural Network," IEEE Access, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3004284.

[11] X. Li, Y. Li, H. Yang, L. Yang, and X.-Y. Liu, "DP-LSTM: Differential Privacy-inspired LSTM for Stock Prediction Using Financial News," Dec. 2019, [Online]. Available: http://arxiv.org/abs/1912.10806

[12] S. Liu, "Long Term Stock Prediction Based On Financial Statements." [Online]. Available: github.com/sugia/tradeX/tree/main/data.

[13] A. A. A. Aguirre, R. A. R. Medina, and N. D. D. Méndez, "'Machine learning applied in the stock market through the Moving Average Convergence Divergence (MACD) indicator,'" Investment Management and Financial Innovations, vol. 17, no. 4, pp. 44–60, 2020, doi: 10.21511/imfi.17(4).2020.05.

[14] J. Welles Wilder, Jr., "The Relative Strength Index," New Concepts in Technical Trading Systems, vol. 5, no. 14, pp. 63–70, 1978.

[15] A. Julianto, A. Sunyoto, D. Ferry, and W. Wibowo, "Optimasi Hyperparameter Convolutional Neural Network Untuk Klasifikasi Penyakit Tanaman Padi (Optimization of Convolutional Neural Network Hyperparameters for Classification of Rice Plant Diseases)," TEKNIMEDIA, vol. 3, no. 2, 2022.