Integrating Sentiment with LSTM for Enhanced Stock Market Prediction Accuracy

Siththarthan Arunthavabalan
Computer Science and Engineering
Graduate Student

***Abstract*—Stock market prediction is a complex tasks that involves analyzing and interpreting extensive amount of historical data to predict the future prices of stocks. Although traditional statistical model perform relatively well in predicting future stock prices, they lack in capturing non-linear relationships and complex patterns present within the data. Due to the recent advancements in the deep learning and development of more sophisticated models for sequential data, application of these models in stock market prediction has increased. In recent years, deep learning models such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks shown promising results in predicting future stock prices as they handle the sequential depencies effectively. This project aims to explore the application of LSTM models in predicting stock prices by utilizing comprehensive set of stock price indicators. This study consider crucial stock factors such as momentum and volatility indicators, and basic stock features.In addition to LSTM, this project seek to incorporate sentiment of the stocks and tries to improve the limitations of the traditional statistical models by utilizing deep learning architecture thus provide more reliable prediction results.**

***Index Terms*—Stock market prediction, Deep learning, Long Short-Term Memory (LSTM), Momentum indicators, Sentiment analysis**

I. Introduction

Stock market refers to several exchanges in which shares of publicly traded companies are bought and sold [1]. It serves as a critical barometer for the health of an economy, reflecting the collective valuation of public companies, which is based on the current performance and anticipation of their future performance. Stock price refers to the current price that a share of a specific company is trading for on the stock market. It reflects the public willingness to pay for a piece of the company. The stock price fluctuates based on a variety of factors, such as macroeconomic factors, market anticipation, confidence in the company's management and operation, political announcements, industry developments, etc.

Although stock prices fluctuate due to various factors, it is possible to predict stock prices with a certain level of confidence by analyzing patterns present in the stock price history and trends in historical stock market data using statistical and machine learning models. These models are capable of recognizing recurring behaviors and correlations within the data, thus enabling the prediction of future price movements based on past and present information. Traditional statistical models typically rely on predefined equations and assumptions about the data they analyze, which can limit the model's ability to handle complex and non-linear relationships within vast amounts of structured or unstructured data.

Deep learning, on the other hand, uses multi-layer neural networks that can automatically learn patterns and relationships directly from data without explicit rules. This capability allows deep learning models the ability to analyze data that are complex and nuanced making it superior to traditional statistical models. Traditional models often require manual feature selection and are limited in their ability to process and learn from colossal amounts of unstructured data. However, deep learning algorithms can automatically detect and learn features directly from data. Furthermore, Deep learning models such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks are specially designed to handle sequential data like stock prices, allowing them to predict future trends based on the temporal dynamics of the market.Since LSTM was developed to overcome the shortcomings of the RNN, the stock price prediction is performed using only the LSTM architecture.

II. Related Works

A. Recurrent Neural Networks (RNN) in Stock Price Prediction

RNN is a neural network that excels at modeling and processing sequential data. It has a unique capability to memorize the previous state which can be used to predict the current state. Unlike typical hidden layers in neural networks, which operate independently and receive input solely from the preceding layer, RNN hidden layers also incorporate output from the previous layer in addition to their input layer. This distinctive architecture enables RNN to capture temporal dynamics and perform better for time series data. The inherent correlation of stock price fluctuations with previous trends makes RNN well-suited for analyzing and predicting the movements of stock prices. Several papers explored the use of RNN in stock price prediction and it shows promising results in predicting the stock prices.

Researchers utilized RNN as a foundation model and supplemented with various techniques to predict the stock prices.Patel et al [2]. used RNN and LSTM models to predict the next day's closing stock prices with 89% accuracy.Ni and Li [3] combined Convolutional Neural Networks (CNNs) with

RNNs to create a C-RNN model for Forex price prediction, which outperformed CNN and LSTM models alone.Li, C. [4] used a Multi-Task RNN capable of handling multiple tasks simultaneously, incorporating trend and volatility tasks as lower-level tasks and price movement prediction as a higher-level task. The model also incorporated Markov Random Fields (MRFs) to capture complex dependencies in the data.Chen, W. [5] developed the RNN-Boost model, which combines AdaBoost with RNN and uses Gated Recurrent Units (GRUs) to address the vanishing gradient problem. The model incorporates technical features, sentiment features, and Latent Dirichlet Allocation (LDA) features, leading to improved accuracy and stability. Zheng, Z. [5] explored a hybrid model that combines an attention-based RNN (ARNN), wavelet denoising, and Autoregressive Integrated Moving Average (ARIMA) for Forex price prediction. The model using denoised data and the combination of ARIMA with ARNN outperformed other variations.

B. Long Short-Term Memory (LSTM) in Stock Price Prediction

LSTM is the variation of RNN that is capable of maintaining the information in memory for a long period of time, making it more effective at learning long-term dependencies compared to typical RNN making LSTM more suitable for handling sequential data such as stock prices. LSTM achieves this by having a special unit called memory cells and various gates (input gate, forget gate, output gate) that control the flow of information and mitigate vanishing gradient and exploding gradient problems. Many research papers have investigated the potential of using LSTM and LSTM with some variation for predicting stock prices and have found it to be a promising approach with positive results.

Nelson, D.M. [6] used LSTM with historical price data and technical analysis indicators to predict stock prices. Data preprocessing techniques such as log-return transformation and exponentially weighted moving averages (EWMA) were applied to stabilize the data and reduce noise. The LSTM model outperformed random forest (RF) and multi-level perceptron (MLP) models in terms of accuracy and yielded positive returns on all tested stocks. Li, H. [7] proposed an enhanced LSTM called the multi-input LSTM (MI-LSTM) model, which extracts valuable information from low-correlated features and discards detrimental noise by deploying additional input gates. MI-LSTM utilized prices of related stocks (auxiliary series) to improve the model's performance, achieving the lowest error compared to the original LSTM and LSTM-C models. Skehin, T. [8] investigated the combination of LSTM with ARIMA and wavelet techniques, using Maximal Overlap Discrete Wavelet Transform (MODWT) to decompose time series into different frequency components. Surprisingly, ARIMA outperformed LSTM across most stocks, and the implementation of MODWT did not significantly increase gains when combined with ARIMA or LSTM. Jin, Zhigang et al. [9]explored the integration of sentiment analysis with LSTM models for predicting stock closing prices. They incorporated investor sentiment using binary indices based on bullish or bearish comments, and employed empirical modal decomposition (EMD) to tackle non-stationary stock time series. The revised LSTM with attention model (S_EMDAM_LSTM) predicted prices closest to the actual prices compared to other models.

III. Methodology

A. Architecture

Initially, Long Short-Term Memory (LSTM) [10] architecture, along with varying hyper parameters in PyTorch, was used for the prediction of stock prices based on various stock market indicators, establishing this approach as the baseline performance. In addition to the baseline models, supplemental techniques previously employed by other researchers, as mentioned in the related work section, will also be applied to evaluate the differences in performance. For the achievement of this project's objectives, sentiment analysis and content capturing was conducted using the EODHD API [11]. Sentiment analysis was then be applied as supplements to the base model, and improvements in prediction performance were evaluated. The initial plan was to extract historical data from a website and apply the OpenAI API for sentiment analysis. However, due to the paywall for historical financial news related stock prices, the EODHD [11]API was chosen instead. The EODHD API provides normalized sentiment scores and sentiment scores calculated based on the number of news articles. Therefore, their sentiment scores from EODHD [11] were used directly in the model.

B. Experimental Methods

For the scope of this project, base model and enhanced model with sentiment analysis was utilized and evaluated for their performance on Apple(AAPL) stock with entire historical data. Stock prediction started with application of the base LSTM model to Apple stock to predict the closing prices based on the historical prices of the stock. Initially, the accuracy was assessed by comparing the actual and closing prices (base prediction accuracy). Denoising was applied to the stock's closing price (time series data) using wavelet techniques to explore the impact of prominent underlying patterns on price prediction effectiveness. The model's predictions with and without denoising were compared to determine whether to proceed with denoised or non-denoised data. After selecting the best data, medium and short term indicators such as Simple Moving Average (SMA) with 20 days(SMA20), Exponential Moving Average (EMA) with a window of 20 periods (EMA 20), volume traded at time step (day), and On-Balance Volume (OBV) were applied to check the model performance. Following the financial indicators, daily sentiment score, count (instances used to calculte the sentiment) were Incorporated to evaluate the improvement in accuracy.

1) LSTM Architecture

Recurrent Neural Networks (RNNs) were introduced to handle sequential data such as time series and natural language processing due to the limitations that traditional neural

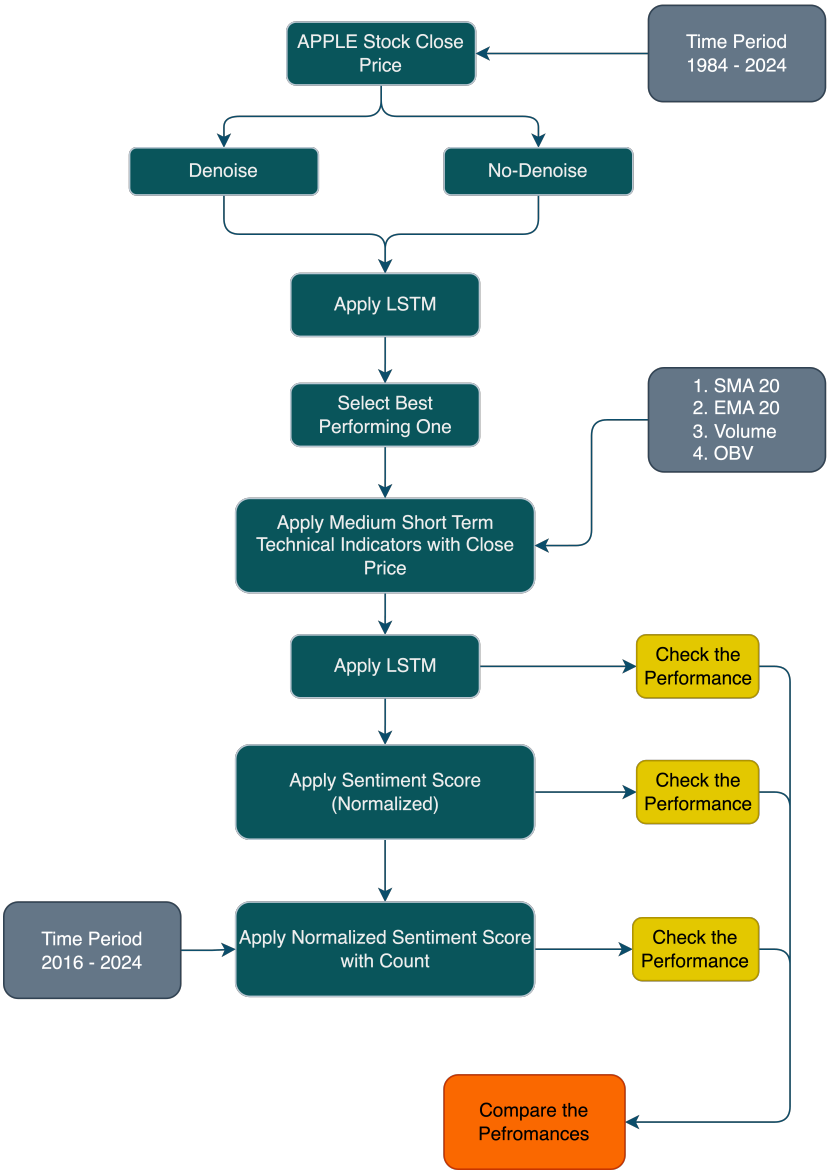

Fig. 1. Flow Chart for Experimental Plan

networks had in managing such data. sequential data requires the memory of previous input to perform prediction. Although, RNN performed better in handling sequential data, it suffered from exploding and vanishing gradient problems which prevent the RNN learn long-term dependencies of the data. In order to address this issues, LSTM was developed. LSTM introduced gating mechnisms that controls the flow of information, decides which and how much information to hold and forget. LSTM consists of three gates: input gate, output gate, forget gate. These gates work together to control the cell states of the model.This architecture enables LSTMs to capture long-term dependencies and complex patterns in sequential data, making them widely used in applications such as stock price prediction, speech recognition, and language modeling. The equations [1 2 3 4 5 6] show how the these gates works and control the data flow. The figure 9 shows the LSTM cell architecture.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{1}$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{3}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{4}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{5}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{6}$$

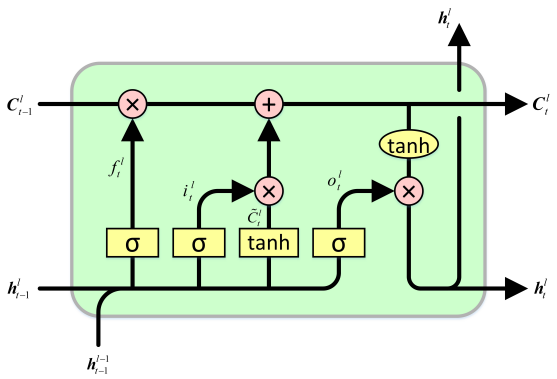

Fig. 2. LSTM Architecture

2) Wavelet Denoising

Wavelet denoising is used to remove noise from time series data, enhancing the underlying signal for more accurate analysis and prediction. Unlike traditional filtering methods, wavelet denoising leverages the wavelet transform, which decomposes the data into different frequency components at multiple resolutions. This multi-resolution analysis allows for the isolation of noise from the true signal based on their frequency characteristics. The process involves three main steps: decomposition, thresholding, and reconstruction. First, the time series data is decomposed into wavelet coefficients using a chosen wavelet function. Next, these coefficients are thresholded to suppress the noise components while preserving significant signal features. Finally, the denoised signal is reconstructed from the threshold coefficients. This technique is effective for financial time series data, such as stock prices, as it helps to understand prominent patterns and trends that may be obscured by market noise, thereby improving the performance of predictive models.

Daubechies 4 (db4) wavelet was used to decomposed the time series data due to its ability to capture the time and frequency characteristics effectively.Unlike other wavelet method such as Haar wavelet, which is too simplistic and complex Coiflet wavelet which might over complicate the analysis. The db4 strikes balance between smoothness and localization making it adept at detecting sharp changes and transient features which are typical in stock prices. The db4 is designed to handle the finer details without losing the broader trends make it more suitable for stock price data denoising.

3) Simple Moving Average (SMA) with 20 Days- SMA 20

The Simple Moving Average (SMA) with 20 days (SMA 20) is a technical indicator used in stock analysis to smooth out price data by creating a constantly updated average price over a specified period. In this case, the period is 20 days. It is calculated by summing the closing prices of a stock for the past 20 days and then dividing the sum by 20. This average usually plotted on a chart to provide a clear visual representation of the stock's price trend over time. SMA 20 is important because it helps traders and analysts identify the direction of the trend and potential buy or sell signals. When the stock price crosses above the SMA 20, it may indicate a buying opportunity, whereas crossing below the SMA 20 may suggest a selling opportunity. By smoothing out daily price fluctuations, the SMA 20 provides a clearer picture of the underlying trend. This is useful for the model to predict the prices based on the buying or selling opportunity. If the SMA 20 trend is buying opportunity, the stock prices tends to go higher and if SMA 20 indicates the selling opportunity, prices tends to go lower.

4) Exponential Moving Average (EMA) with a Window of 20 Days - EMA 20

The Exponential Moving Average (EMA) with a window of 20 periods (EMA 20) is a type of moving average that gives more weight to recent prices, making it more responsive to new information. It is calculated using a formula that applies a weighting factor to the most recent data points, which decreases exponentially over time. The weighting factor for the EMA 20 is(2/21). EMA 20 is important because it reacts more quickly to price changes compared to the Simple Moving Average (SMA), making it useful for identifying short-term trends and potential reversal points. The reversal point is where the price is likely to change direction. Traders often use the EMA 20 to confirm trends and generate trading signals. For instance, a price crossing above the EMA 20 can be seen as a bullish signal, while a price crossing below it can be seen as a bearish signal.

5) Volume Traded at Time Step (Day)

Volume traded at a time step (day) refers to the total number of shares of a stock that were bought and sold during a specific trading day. This metric is calculated by summing all transactions for the stock within the given day. Volume is a crucial indicator because it reflects the level of interest and activity in a stock. High trading volume often indicates strong investor interest and can signal the strength of a price move, either upward or downward. Conversely, low trading volume might suggest a lack of interest or a weak price move. Analyzing volume trends alongside price movements helps traders and analysts confirm the validity of a trend or identify potential reversals. Thus, volume is included as a feature to enable the model to identify the underlying relationship between price and volume trends and improve the accuracy/

6) On-Balance Volume (OBV)

On-Balance Volume (OBV) is a momentum indicator that uses volume flow to predict changes in stock prices. It is calculated by adding the day's volume to a cumulative total when the stock's price closes higher than the previous day and subtracting the day's volume when the price closes lower. The OBV line is then plotted to show the cumulative volume flow. OBV is important because it helps traders identify potential buy and sell signals based on volume trends. When the OBV increases, it indicates that buying pressure is building up, which could lead to higher prices. Conversely, when the OBV decreases, it suggests selling pressure and potentially lower prices. By comparing OBV movements to price movements, traders can gain insights into the strength and direction of a trend.

C. Model Training

1) Model Architectures Grid

The model architecture grids illustrate the modifications made to a typical LSTM model and the parameter changes implemented to evaluate the accuracy of each architecture.

TABLE I
MODEL ARCHITECTURES GRID

Model	*Layers(Nos.)*	*Hidden Layers(Nos.)*	*Dropout*
LSTM-1L-10H-1D	1	10	1.0
LSTM-1L-50H-1D	1	50	1.0
LSTM-1L-10H-0.5D	1	10	0.5
LSTM-1L-50H-0.5D	1	10	0.5
LSTM-2L-10H-1D	1	10	0.5

2) Normalization of Data

All the data were normalized before training the model. Normalizing the data before training the LSTM has several advantages. Firstly, it ensures that all features contribute equally to the learning process, preventing features with larger numerical ranges from dominating the training process. Secondly, normalization helps accelerate the convergence of the optimization algorithm, leading to faster training times. Thirdly, it reduces the risk of gradient-related issues, such as vanishing or exploding gradients, by keeping the input data within a consistent range. A Min-MaxScaler as shown in Equation 7 was used to normalize the data, except for the sentiment score, which was already normalized. Equation 8 shows he de-normalization which is required to transform the scaled predicted values to actual closing price of the stock

$$X_t^{(n)} = \frac{X_t - \min(X_t)}{\max(X_t) - \min(X_t)} \tag{7}$$

$$X_t = X_t^{(n)} \cdot (\max(X_t) - \min(X_t)) + \min(X_t) \tag{8}$$

D. Results

TABLE II
RESULTS OF THE ARCHITECTURES STOCK CLOSE PRICES

Model	*Epoch*	*Validation Loss)*	*Test Loss*
LSTM-1L-10H-1D	0	0.00126	0.290
LSTM-1L-50H-1D	85	0.01871	0.315
LSTM-1L-10H-0.5D	7	0.00409	0.224
LSTM-1L-50H-0.5D	99	5.6185e-05	0.001
LSTM-2L-10H-0.5D	69	3.40377e-05	0.021

TABLE III
RESULTS OF LSTM-1L-50H-0.5D WITH ADDITIONAL FEATURES

Features	*Epoch*	*Validation Loss)*	*Test Loss*
Denoised	98	3.27122e-05	0.002
Non-Denoised	99	5.6185e-05	0.001
Non-Denoised + Sentiment	99	9.5228e-05	0.001

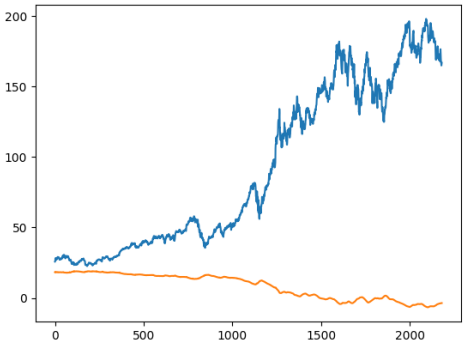

Fig. 3. LSTM-1L-10H-1D

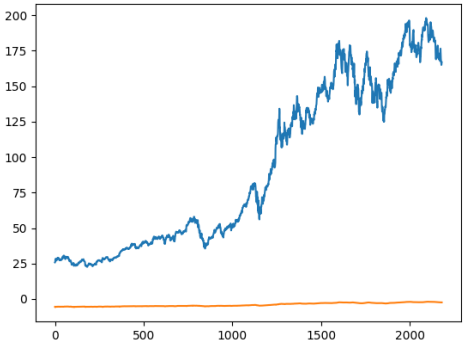

Fig. 4. LSTM-1L-50H-1D

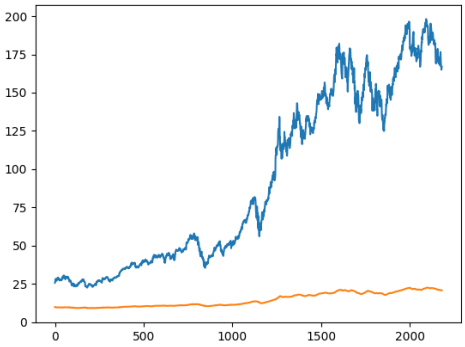

Fig. 5. LSTM-1L-50H-0.5D

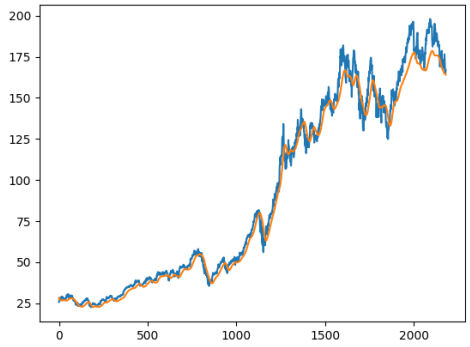

Fig. 6. LSTM-1L-50H-0.5D

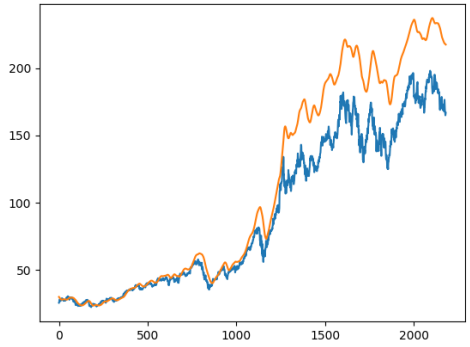

Fig. 7. LSTM-2L-50H-0.5D

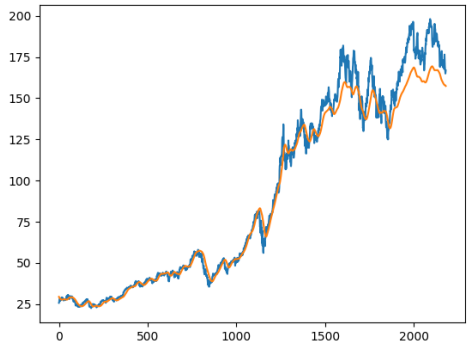

Fig. 8. LSTM-2L-50H-0.5D - Denoised Data

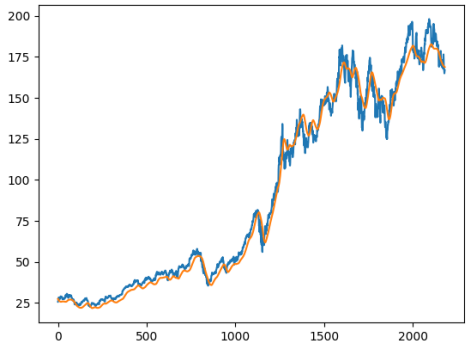

Fig. 9. LSTM-2L-50H-0.5D - Non-Denoised Data with Sentiment

E. Discussion

Multiple variations of LSTM architectures were used to predict the closing stock price (see Figure III). Among them, the LSTM with one layer and 50 hidden units, and a dropout rate of 0.5, performed the best. When the number of layers increased, the model's performance started to deteriorate. Therefore, the LSTM with one layer, 50 hidden units, and a dropout rate of 0.5 was selected to predict the stock price, incorporating additional features. When denoised data was used instead of non-denoised data, the test loss slightly increased. Therefore, further analysis was performed using the non-denoised data. When sentiment was introduced as a feature, the improvement was not significant, implying that sentiment did not affect the prediction of the stock's closing price.

F. Future Works

This project analyzed the closing price prediction for only one stock. Although it is challenging to predict stock prices with limited data, the model performed quite well. In the future, more features can be included to enhance the predictions, and conducting our own sentiment analysis instead of using pre-analyzed sentiment data will help fine-tune the model for greater accuracy. Additionally, integrating more advanced techniques, other than wavelet ,also could further improve the model's performance

REFERENCES

[1] Investopedia. Stock market definition. https://www.investopedia.com/terms/s/stockmarket.asp, 2023. Accessed: 2023-03-31.

[2] J. Patel, M. Patel, and M. Darji. Journal of emerging technologies and innovative research. *JETIR*, 5, 2018. Issue 11.

[3] L. Ni, Y. Li, X. Wang, J. Zhang, J. Yu, and C. Qi. Forecasting of forex time series data based on deep learning. *Procedia Computer Science*, 147:647–652, 2019.

[4] C. Li, D. Song, and D. Tao. Multi-task recurrent neural networks and higher-order markov random fields for stock price movement prediction. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1141–1151, 2019.

[5] W. Chen, C. K. Yeo, C. T. Lau, and B. S. Lee. Leveraging social media news to predict stock index movement using rnn-boost. *Data and Knowledge Engineering*, 118:14–24, 2018.

[6] D.M. Nelson. Ijcnn 2017: the international joint conference on neural networks. IEEE Computational Intelligence Society, International Neural Network Society, Institute of Electrical and Electronics Engineers, n.d.

[7] H. Li, Y. Shen, and Y. Zhu. Stock price prediction using attention-based multi-input lstm. In *Proceedings of Machine Learning Research*, volume 95, 2018.

[8] T. Skehin, M. Crane, and M. Bezbradica. Day ahead forecasting of faang stocks using arima, lstm networks and wavelets, n.d.

[9] Z. Jin, Y. Yang, and Y. Liu. Stock closing price prediction based on sentiment analysis and lstm. *Neural Computing and Applications*, 32(13):9713–9729, 2020.

[10] torch.nn.LSTM - pytorch documentation. https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html, 2023. Accessed: 2023-03-31.

[11] EOD Historical Data. Stock market financial news api, 2024. Accessed: 2024-05-16.