Enhancing Stock Price Prediction with Bi-Directional LSTM Networks: A Comparative Analysis with Traditional LSTM Models

Ali Yiğit Başaran
Department of Data Informatics
Middle East Technical University
Ankara, TURKEY
yigit.basaran@metu.edu.tr

***Abstract*—Stock price prediction is a pivotal challenge due to the dynamic and volatile nature of financial markets. Accurate predictions can yield significant financial gains and strategic advantages. This paper presents a comparative analysis of stock price prediction models, focusing on Bi-Directional Long Short-Term Memory (Bi-LSTM) networks and traditional Long Short-Term Memory (LSTM) models. Using a dataset comprising daily trading information for THYAO stock from 2020 to 2024, we aim to identify which architecture better captures the complexities of stock market data. The dataset includes comprehensive financial metrics, such as closing prices, volume, capitalization, exchange rates, and market indices. We preprocess the data using Min-Max normalization and create input sequences using a sliding window approach. The baseline LSTM model and the Bi-LSTM model are developed and trained on historical data, with performance evaluated using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). Our methodology includes hyperparameter optimization using Optuna, feature engineering with technical indicators, and advanced regularization techniques. The results demonstrate that the Bi-LSTM model, with its ability to process data in both forward and backward directions, provides superior performance in capturing temporal patterns and predicting stock prices.This study contributes to the field by offering insights into the effectiveness of Bi-LSTM networks for financial time-series forecasting, highlighting their potential to improve prediction accuracy and support strategic decision-making in financial markets.** **Keywords—Stock Price Prediction, Bi-Directional Long Short-Term Memory (Bi-LSTM), Long Short-Term Memory (LSTM), Deep Learning, Financial Time-Series Forecasting.**

I. INTRODUCTION

A. Problem Statement

Stock price prediction remains one of the most complex and challenging tasks in financial markets due to their dynamic and highly volatile nature. Accurate stock price predictions can yield significant financial gains and provide strategic advantages for investors, traders, and financial analysts. However, the inherent complexity of financial markets, which are influenced by a multitude of factors such as economic indicators, market sentiment, and geopolitical events, poses a substantial challenge to achieving precise forecasting.

Traditional methods, including statistical and econometric models, often fall short in capturing the non-linear relationships and temporal dependencies inherent in financial time-series data. This inadequacy necessitates the exploration of more sophisticated approaches that can model these complex patterns and dependencies effectively.

B. Aim of the Project

The primary aim of this project is to enhance the accuracy of stock price predictions by leveraging advanced deep learning techniques. Specifically, the project will conduct a comparative analysis of Bi-Directional Long Short-Term Memory (Bi-LSTM) networks and traditional Long Short-Term Memory (LSTM) models to determine which architecture best captures the complexities of stock market data.

C. Objectives

To achieve this aim, the project will focus on the following objectives:

1. **Data Collection and Preprocessing:** Gather and preprocess historical stock price data and relevant financial metrics from the Istanbul Stock Exchange for THYAO stock from 2020 to 2024.
2. **Model Development:** Develop and configure both Bi-LSTM and traditional LSTM models for stock price prediction.
3. **Hyperparameter Optimization:** Employ automated hyperparameter tuning techniques to optimize model performance.
4. **Comparative Analysis:** Evaluate and compare the performance of Bi-LSTM and traditional LSTM models using relevant metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE).
5. **Feature Engineering:** Enhance model inputs by incorporating additional financial indicators and technical analysis metrics.
6. **Model Validation and Testing:** Validate and test the models on unseen data to assess their generalizability and robustness.

D. Research Question

The core research question guiding this project is:

"Which deep learning architecture, Bi-Directional LSTM or traditional LSTM, provides more accurate stock price predictions for THYAO stock?"

By addressing this research question, the project aims to provide valuable insights into the effectiveness of advanced deep learning models for financial time-series forecasting, potentially leading to improved predictive accuracy and better-informed investment strategies.

II. LITERATURE REVIEW

Jain et al. (2018) explored the application of deep neural networks to predict stock prices using daily stock data for Tata Consultancy Services (TCS) and Madras Rubber Factory Limited (MRF). The dataset used includes the Open, High, Low, and Close values, similar to the THYAO dataset, though lacking additional financial indicators such as exchange rates and market capitalization. The preprocessing technique involved applying z-score normalization to scale the values for neural network processing [1].

The study implemented various deep learning models, including Convolutional Neural Networks (CNN), Long Short-Term Memory Networks (LSTM), and a hybrid Conv1D-LSTM model. These models were trained using sliding windows of past data to predict the next day's closing prices. The performance of these models was evaluated using metrics like Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). The results demonstrated that the hybrid Conv1D-LSTM model achieved lower prediction errors compared to individual CNN and LSTM models, indicating superior prediction accuracy [1].

Weaknesses:

Limited Financial Indicators: The dataset used only includes the Open, High, Low, and Close values, but lacks additional financial indicators such as exchange rates, market capitalization, and other economic variables that could improve prediction accuracy.

Single Directional Models: The models, including CNN, LSTM, and Conv1D-LSTM, only consider past data without leveraging future context, which could be beneficial in time-series forecasting.

Hybrid Complexity: The Conv1D-LSTM model's complexity might lead to longer training times and higher computational costs without a proportional increase in accuracy.

Mootha et al. (2020) focused on predicting stock prices using Bi-Directional Long Short-Term Memory (Bi-LSTM) networks, which can capture both past and future contexts effectively. The dataset used in this study includes time-series data from various stock markets, featuring open, high, low, and close prices, but excludes volume and capital data. Preprocessing involved feature scaling, data smoothing, and anomaly detection [2].

The Bi-LSTM model leveraged sequence-to-sequence learning to enhance prediction accuracy, particularly for short-term forecasting. The study compared the performance of the Bi-LSTM model against standard LSTM models, using metrics such as accuracy, RMSE, and financial metrics comparing daily profits from buy-sell actions dictated by the model versus no buy-sell actions. The results indicated that the Bi-LSTM model outperformed standard LSTM models in handling complex temporal patterns, providing more robust and profitable predictions [2].

Weaknesses:

Absence of Volume and Capital Data: The dataset excludes crucial financial metrics such as trading volume and market capitalization, which can be significant predictors of stock price movements.

Short-term Focus: The model is primarily focused on short-term forecasting, potentially limiting its utility for longer-term investment strategies.

Complexity in Implementation: The sequence-to-sequence and multitask learning approaches introduce additional complexity, making the model harder to implement and tune.

Selvin et al. (2017) examined the use of different deep learning architectures, including LSTM, Recurrent Neural Networks (RNN), and CNN, for stock price prediction of National Stock Exchange (NSE) listed companies. The dataset focused on daily trading figures, such as volume and price fluctuations, which align well with the characteristics of the THYAO dataset. The preprocessing involved a sliding window approach to create sequences for training, essential for time-series prediction [3].

The study implemented a sliding window approach with overlapping windows for short-term future prediction. The performance of these models was quantified using percentage error metrics. The results showed that CNN provided more accurate predictions compared to RNN and LSTM models, likely due to CNN's ability to focus on the current window for prediction without relying on previous information. The study concluded that CNN was better suited for capturing dynamical changes in stock prices, while RNN and LSTM struggled with the highly dynamic nature of stock market data [3].

Weaknesses:

Overfitting to Short-term Data: The sliding window approach might lead to overfitting on short-term data patterns, making the model less effective for predicting longer-term trends.

To address the weaknesses in the literature, the following improvements are proposed:

A. *Incorporating Additional Financial Indicators*

Including a comprehensive set of financial indicators such as exchange rates, trading volume, market capitalization, and relevant economic variables. This additional data can improve the model's ability to capture complex market dynamics.

B. *Utilizing Bi-Directional LSTM (BiLSTM)*

Implementing BiLSTM allows the model to consider both past and future data points, improving its ability to understand temporal dependencies and patterns that single-directional models might miss.

C. *Feature Engineering and Regularization*

- Integrating technical indicators like Exponential Moving Average (EMA) and Relative Strength Index (RSI) into the feature set to provide the model with additional insights into market trends and momentum.
- Applying advanced regularization techniques such as dropout and L2 regularization to prevent

overfitting and improve the generalization of the model.

D. Hyperparameter Optimization

Utilizing Optuna to systematically explore and optimize the hyperparameters of both LSTM and BiLSTM models, ensuring the best possible performance with minimal validation loss.

E. Early Stopping and Learning Rate Scheduling

Implementing early stopping and learning rate reduction on plateau to avoid overfitting and ensure efficient training. These callbacks help in dynamically adjusting the learning process based on the model's performance on validation data.

III. Dataset Description

The dataset utilized in this study consists of daily trading information for the THYAO stock, spanning from January 2, 2020, to April 15, 2024. The dataset, sourced from the Istanbul Stock Exchange, comprises various financial metrics that provide a comprehensive view of the stock's performance and relevant economic indicators. The features included in the dataset are as follows:

- **Date**: The date of the stock price record.
- **Closing Price in TL**: The closing price of the stock in Turkish Lira (TL) on the given day.
- **Minimum Price in TL**: The lowest price of the stock in TL on the given day.
- **Maximum Price in TL**: The highest price of the stock in TL on the given day.
- **Weighted Average Price in TL**: The average price of the stock in TL, weighted by volume.
- **Volume in TL**: The trading volume in Turkish Lira.
- **Capital in million TL**: The capitalization of the THY Company in million Turkish Lira.
- **USDTRY**: The exchange rate from US Dollar to Turkish Lira for the given day.
- **BIST 100**: The BIST 100 index value on the given day.
- **Market Value in million TL**: The market value of the stock in million Turkish Lira.
- **Market Value in million USD**: The market value of the stock in million US Dollars.
- **Public Market Value in million TL**: The public trading value of the stock in million Turkish Lira.
- **Public Market Value in million USD**: The public trading value of the stock in million US Dollars.

In addition to these features, two technical indicators have been calculated and added to the dataset to enhance the predictive capabilities of the models:

Exponential Moving Average (EMA): A technical indicator that smooths price data to create a trend-following indicator by applying a weighted factor to the current price. Formula of EMA is provided in (1), where *n* is window length.

$$EMA_t = Closing\ Price_t \times (2/(n+1)) + EMA_{t-1} \times (1 - 2\ (n + 1)) \quad (1)$$

Relative Strength Index (RSI): A momentum oscillator that measures the speed and change of price movements, oscillating between zero and 100. Equations (2) and (3) show the calculation of RSI. Gain and loss refer ro daily profit or loss from stock price. The average gain and loss typically calculated over a 14-day period.

$$RSI = 100 - (\ 100 / (\ 1 + RS\ )\ ) \quad (2)$$

$$RS = Average\ Gain / Average\ Loss \quad (3)$$

Only RSI indicator has missing values in entire dataset for first 4 data, which are imputed with first consequtive non-zero value, due to its computation given in (2) and (3).

A. Descriptive Statistics

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

Table I: Descriptive Statistics of Dataset

	count	mean	std	min	%25	%50	%75	max
Closing	1073	83,22	91,91	7,71	12,68	28,08	138,7	310
MIN Price	1073	81,83	90,53	7,41	12,56	27,64	135,5	308,5
MAX Price	1073	84,73	93,48	7,95	12,86	28,5	142,3	312,75
WAP	1073	83,35	92,05	7,75	12,71	28,05	139,26	309,81
Volume	1073	4,51E+09	3,97E+09	1,81E+08	1,1E+09	3,34E+09	7,17E+09	2,36E+10
Capital	1073	1380	0	1380	1380	1380	1380	1380
USDTRY	1073	15,24	8,11	5,86	7,82	14,06	19,04	32,42
BIST 100	1073	3390,91	2604,63	842	1386	2043	5115	9814
MV TL	1073	114845,9	126840,3	10640	17498	38750	191406	427800
MV USD	1073	5549,61	4040,09	1496	2187	3221	10033	13635
PMV TL	1073	57890,44	63958,45	5251	8812	19592	96488	215825
PMV USD	1073	2795,95	2037,7	753	1104	1586	5048	6870
EMA	1073	81,51	90,31	9	12,66	27,29	138,55	296,89
RSI	1073	57,01	19,35	1,45	42,78	57,47	71,45	99,32

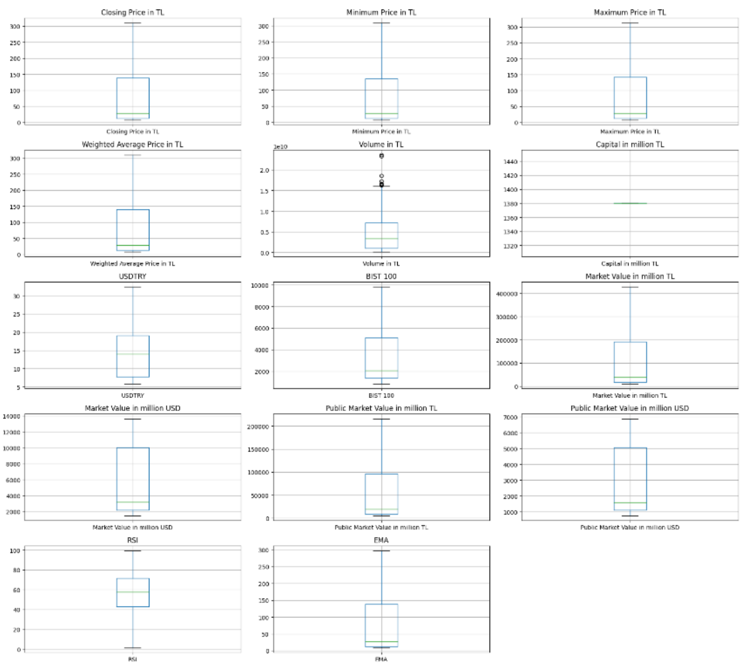

Fig. 1. Boxplots of Features and Target Variable in Dataset

Descriptive statistics of entire dataset and boxplots of the features and target variable (Closing Price in TL) are provided in Table I and Fig. 1, relatively.

According to Table I and Fig. 1 features and target variable are analyzed below.

- Most of the columns (Closing Price, Minimum Price, Maximum Price, Weighted Average Price, Volume, USDTRY, BIST 100, Market Value in TL, Market Value in USD, Public Market Value in TL, Public Market Value in USD, EMA) show a **right-skewed** distribution due to high means and standard deviations with long tails on the right.
- The **RSI** appears closer to a **normal distribution**.
- The **Capital in million TL** column shows no variability, thus does not fit a normal distribution or any skew. It resembles an uniform distribution. Nevertheless this observation may be caused by the relatively high values of the data.

The descriptive statistics and boxplots indicate several key characteristics of the dataset that make min-max normalization a suitable choice in preprocessing steps:

1. **Skewed Distributions**:

   Many features, such as closing price, minimum price, maximum price, volume in TL, and various market values, exhibit right-skewed distributions. Min-max normalization is effective in scaling these skewed distributions to a common range, allowing models to learn more effectively from the data without being biased by extreme values.

2. **Different Value Ranges**:

   The dataset includes features with vastly different value ranges. For example, the closing price in TL ranges from 7.71 to 310, while the volume in TL ranges from 181 million to 23.59 billion. Min-max normalization scales all features to a consistent range, typically [0, 1], ensuring that features with larger ranges do not dominate the learning process.

3. **High Variability**:

   Features such as market value in TL and USD, and public market value in TL and USD, show high variability with large standard deviations. Min-max normalization compresses these large ranges into a uniform scale, facilitating better model performance and convergence during training.

B. Approach to Date: Cyclic Nature

The process begins by converting the Date column to a datetime format to allow for easier manipulation and extraction of date-related features. Once the dates are in the correct format, the data is sorted chronologically to maintain the time-series order, which is crucial for any temporal analysis or modeling.

From the datetime data, several features are extracted: the day of the month, the month, the year, and the day of the week. These features provide granular insights into temporal patterns and can enhance the model's ability to learn seasonality and trends within the data.

To capture cyclical patterns inherent in the calendar, cyclic encoding is applied to the day of the week and month features. This encoding transforms these features into sine and cosine components, which allows the model to understand that the time elements are periodic (e.g., Monday is next to Sunday in a cyclic manner).

Formulas of converting day and month features to sine and cosine counterparts are given (4), (5), (6), and (7).

$$Day\_Sin = sin(2 * \pi * Day / 7) \tag{4}$$

$$Day\_Cos = cos(2 * \pi * Day / 7) \tag{5}$$

$$Month\_Sin = sin(2 * \pi * Month / 12) \tag{6}$$

$$Month\_Cos = cos(2 * \pi * Month / 12) \tag{7}$$

C. Correlation Matrix

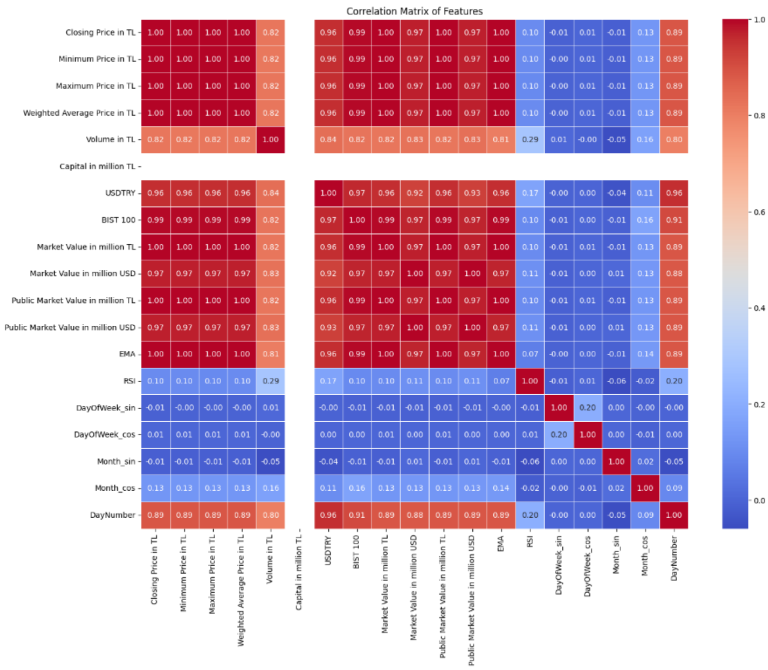

Fig. 2. Correlation Matrix of Features and Target Variable

Fig. 2 presents the correlation matrix between the target variable (Closing Price in TL) and various features. It shows that features such as Minimum Price in TL, Maximum Price in TL, and Weighted Average Price in TL exhibit extremely high correlations with the closing price. These findings suggest that the stock's daily minimum and maximum prices, as well as its weighted average price, are closely tied to the closing price, indicating minimal deviations. Additionally, the Market Value in million TL and Public Market Value in million TL both demonstrate perfect correlations with the closing price, highlighting their direct derivation from the closing price. The Market Value in million USD and Public Market Value in million USD also show very high correlations, albeit with slight influences from the exchange rate. The USDTRY (exchange rate) and BIST 100 index values have strong correlations with the closing price, suggesting significant impacts from exchange rate fluctuations and overall market performance. The DayNumber feature also has a notable correlation, reflecting the importance of temporal progression in predicting stock prices.

Conversely, features such as RSI (Relative Strength Index) and Volume in TL exhibit very low correlations with the closing price, indicating limited direct impact on stock price prediction. Similarly, cyclical features like DayOfWeek_sin, DayOfWeek_cos, Month_sin, and Month_cos show very low correlations, suggesting that weekly and monthly patterns do not play significant roles in the stock price movement.

Fig. 2 also highlights features with high interdependence. Market Value in million TL and Public Market Value in million USD are perfectly correlated due to their derivation from the stock price, influenced by the USDTRY exchange rate. Similarly, Public Market Value in million USD and Public Market Value in million TL show high interdependence for the same reason. The Minimum, Maximum, and Weighted Average Prices in TL are highly correlated, moving together with the closing price and reflecting the day's price range. Additionally, USDTRY and Market Value in million USD exhibit strong interdependence, indicating the significant impact of currency fluctuations on the stock's value in USD.

D. Feature Importance

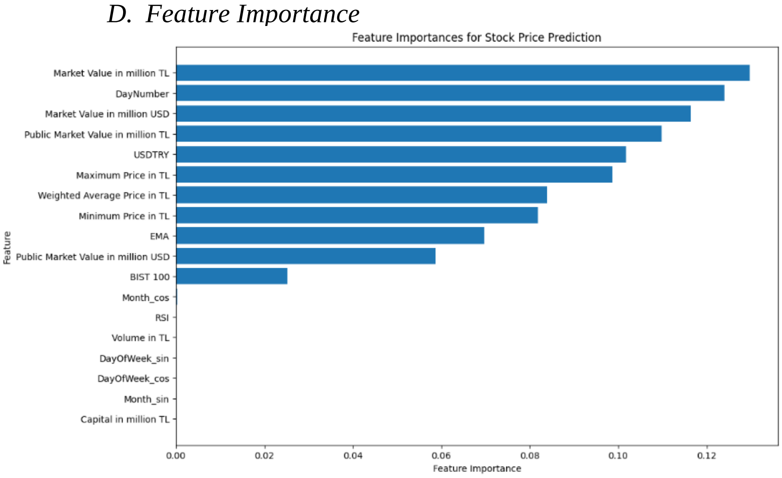

Fig. 3. Feature Importances

Fig. 3 presents the feature importance analysis, highlighting the significance of various features in predicting the target variable.

High Importance Features:

Market Value in million TL emerges as the most significant predictor, capturing overall market sentiment and investor valuation. The DayNumber feature is also highly important, reflecting the significance of temporal trends and patterns. Market Value in million USD is crucial for understanding the stock's valuation from a global perspective. Public Market Value in million TL indicates the influence of publicly traded portions on stock price movements, while the USDTRY exchange rate underscores the impact of currency fluctuations and economic conditions. Maximum Price in TL highlights the relevance of daily highs in predicting stock performance.

Moderate Importance Features:

Weighted Average Price in TL provides a comprehensive view of trading activity, making it moderately important. Minimum Price in TL and EMA (Exponential Moving Average) are also moderately significant, reflecting the relevance of daily lows and smoothed price trends, respectively. Public Market Value in million USD indicates the importance of international perceptions of the publicly traded portion. The BIST 100 index value shows some importance, reflecting broader market trends.

Low Importance Features:

Features such as Month_cos, RSI (Relative Strength Index), and Volume in TL have very low importance, suggesting limited impact on the model's predictions. DayOfWeek_sin and DayOfWeek_cos show negligible importance, indicating that weekly cyclical patterns are not critical. Similarly, Month_sin and Capital in million TL have minimal to no importance, suggesting that the cyclical pattern of months and capital value do not significantly influence the model.

IV. Methodology

A. Preprocessing

A.1. Dropping Features

Based on the correlation matrix in Fig. 2 and feature importance analysis in Fig 3, specific columns were dropped to improve the model's efficiency and performance. The features "Market Value in million USD" and "Public Market Value in million USD" were highly correlated with their TL counterparts, indicating that they provide redundant information. This redundancy can lead to multicollinearity, which negatively impacts the model's performance. Although these USD-based features are relevant, their importance is lower than the corresponding TL-based features, making them less critical for the model.

Similarly, "Minimum Price in TL" and "Maximum Price in TL" were removed due to their extremely high correlation with the closing price. These features add redundant information as their variations are almost directly proportional to the closing price, which is already well represented by the "Weighted Average Price in TL."

"Volume in TL" was dropped because it showed minimal importance in the feature importance analysis. This suggests that trading volume does not significantly influence the model's ability to predict the closing price for this particular dataset. The features "DayOfWeek_cos" and "DayOfWeek_sin" were also removed due to their negligible importance, indicating that the day of the week has little to no effect on the stock price prediction.

The feature "Capital in million TL" was constant across all records, providing no variability and therefore no useful information for the model. "Month_sin" was also excluded as it had very low importance, suggesting that the monthly cyclical pattern does not play a significant role in predicting the stock price.

A.2. PCA

To enhance the model further, Principal Component Analysis (PCA) was applied to reduce the dimensionality of the dataset. PCA transforms the original features into a new set of uncorrelated features called principal components. These components are linear combinations of the original features, ordered to retain the most significant variation present in the dataset. By selecting the top four principal components, the model captures the most critical aspects of the data while reducing redundancy and complexity.

PCA helps in simplifying the dataset, making the model more efficient by focusing on the most informative features. This process improves the model's performance, especially in cases where high-dimensional data might lead to overfitting or where multicollinearity is present. By emphasizing the principal components, the model can better learn and generalize from the data, leading to more accurate predictions. In PCA process, number of principal components is defined as 4. That is, at the end of the process, 4 different components will be held, namely PC1, PC2, PC3 and PC4.

A.3. Splitting Training-Validation and Testing of Sequential Data

The window size, which determines the number of time steps included in each sequence used for prediction is defined as 5. This means that each sequence will contain data from 5 consecutive time steps.

A function is used to create sequences of data points along with their corresponding target values. For each position in the dataset, a sequence of 5 consecutive rows is extracted, and the target value is the closing price on the day following the last day in the sequence. This process continues until there are no more full sequences of the specified window size available in the dataset.

After creating the sequences, the data is split into training, validation, and test sets. The training set is used to train the model, the validation set is used to tune hyperparameters and prevent overfitting, and the test set is used to evaluate the final model performance. The split is done sequentially to maintain the temporal order of the data, which is crucial for time series analysis. Specifically, 70% of the data is allocated to training, 20% to validation, and the remaining 10% to testing.

Input and target variables are

Input Variables: These are the sequences created from the features of the dataset. Each sequence includes data points from the principal components (PC1, PC2, PC3, PC4) for the last 5 time steps.

Target Variable: This is the closing price in TL, which the model aims to predict.

A.4. Min-Max Scaling Data

Min-Max scaler is used to ensure the model can learn effectively, the input and target data are scaled. This scaler transforms the data to a range between 0 and 1, which helps in accelerating the convergence optimization process and improves model performance. The formula of Min-Max scaling is given in (8).

$$X_{scaled} = (X - X_{min}) / (X_{max} - X_{min}) \tag{8}$$

B. Models

B.1. LSTM (Baseline Model)

Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Network (RNN) designed to capture and learn long-term dependencies in sequential data. Traditional RNNs struggle with long-term dependencies due to the vanishing gradient problem, where gradients become very small, preventing the network from learning effectively. LSTMs address this issue by incorporating memory cells and three gating mechanisms: input gate, forget gate, and output gate.

How LSTM Works Theoretically?

Memory Cell: The core component of LSTM is the memory cell, which retains information over long periods. The memory cell state is controlled by the three gates.

Input Gate: This gate determines how much of the new input should be added to the memory cell state.

Forget Gate: This gate decides which parts of the memory cell state should be forgotten or retained.

Output Gate: This gate controls the output based on the memory cell state and the new input.

LSTMs are particularly useful for time series data because they can maintain and learn patterns over long sequences, making them ideal for predicting future values based on historical data.

B.1. BiLSTM

Bidirectional LSTM (BiLSTM) networks are an extension of LSTM networks that process data in both forward and backward directions. This means that BiLSTMs have two hidden states for each time step, one for processing the sequence from start to end and another for processing it from end to start. By doing this, BiLSTMs can capture dependencies from both past and future contexts, providing a more comprehensive understanding of the sequence.

How BiLSTM Works Theoretically

Forward Pass: One LSTM processes the sequence from the beginning to the end.

Backward Pass: Another LSTM processes the sequence from the end to the beginning.

Combining Outputs: The outputs of both passes are combined, usually by concatenation, providing richer context for each time step.

Difference Between LSTM and BiLSTM

LSTM: Capture information from the past to the current time step, useful for sequential data where future information is not available during prediction.

BiLSTM: Capture information from both past and future time steps, providing a more comprehensive understanding of the sequence, especially useful when the entire sequence is available and future context can enhance the prediction.

BiLSTMs are expected to perform better than traditional LSTMs because they consider the context from both directions. This is particularly beneficial in scenarios where future information provides valuable context for making predictions, leading to improved accuracy and robustness in time series forecasting.

C. Training, Validation and Test Structures

C.1. Hyperparameters

- **LSTM Model**:
  - **Number of LSTM Units:** Between 50 and 100 (determined by Optuna)
  - **Dropout Rate:** Between 0.3 and 0.5 (determined by Optuna)
  - **Learning Rate:** Between 1e-5 and 1e-3 (determined by Optuna)
  - **Number of Layers:** 1 or 2 (determined by Optuna)
  - **Batch Size:** Between 32 and 64 (determined by Optuna)
  - **L2 Regularization:** Between 1e-5 and 1e-3 (determined by Optuna)
- **BiLSTM Model**:
  - **Number of LSTM Units:** Between 50 and 100 (determined by Optuna)
  - **Dropout Rate:** Between 0.3 and 0.5 (determined by Optuna)
  - **Learning Rate:** Between 1e-5 and 1e-3 (determined by Optuna)
  - **Number of Layers:** 1 or 2 (determined by Optuna)
  - **Batch Size:** Between 32 and 64 (determined by Optuna)
  - **L2 Regularization:** Between 1e-5 and 1e-3 (determined by Optuna)

C.2. Building Models

- **LSTM:** The model is constructed with the specified number of LSTM units and layers, incorporating dropout for regularization and L2 regularization to prevent overfitting.
- **BiLSTM:** The model is similarly constructed but uses Bidirectional LSTM layers to process data in both forward and backward directions.
- **Compiling the Model**: Both models are compiled using the Adam optimizer with a specified learning rate, and the loss function used is Mean Squared Error (MSE). The metrics tracked are Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

C.3. Training, Validating and Testing Models

- **Training**: The models are trained for 50 epochs with the specified batch size. Early stopping is applied to halt training if the validation loss does not improve for 5 consecutive epochs. Additionally, the learning rate is reduced by a factor of 0.5 if the validation loss plateaus for 10 epochs, with a minimum learning rate threshold of 1e-5.
- **Validation**: The model performance is validated using the validation set, and the final evaluation metric is the validation loss (MSE).
- **Hyperparameter Optimization**: Optuna is used to optimize the hyperparameters by minimizing the validation loss over 5 trials. The best hyperparameters are selected for training the final model.
- **Final Model Training & Testing**: The final model is trained using the best hyperparameters identified by Optuna, ensuring optimal performance, then it predicts on the test data.

D. Results & Discussion

The performance of both LSTM and BiLSTM models was evaluated on the test set using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as metrics. The training and validation loss curves for both models were also analyzed to assess the training process and generalization performance.

D.1. Test MAE and Test RMSE

Hyperparameters of best models, test MAE and test RMSE results are provided in Table II.

Table II: Best Hyperparameters, Test MAE and Test RMSE of LSTM and BiLSTM

	LSTM	**BiLSTM**
Number of LSTM Units	76	68
Droput Rate	0.334	0.336
Learning Rate	0.000123	0.000137
Number of Layers	2	1
Batch Size	49	60
L2 Regularization Rate	0.000993	0.000065
Test MAE	0.1319	0.0581
Test RMSE	0.1481	0.0728

The baseline method in this experiment is the standard LSTM model. Compared to the baseline, the BiLSTM model shows significant improvements according to Table II:

MAE Improvement: The BiLSTM model reduces the MAE by more than half compared to the LSTM model. This reduction indicates that the average magnitude of prediction errors is considerably lower in the BiLSTM model.

RMSE Improvement: The BiLSTM model also shows a substantial decrease in RMSE, suggesting that the BiLSTM model is better at reducing larger errors, which is crucial for reliable stock price predictions.

The results demonstrate the superior performance of the BiLSTM model over the standard LSTM model in predicting stock prices. The BiLSTM model achieved lower MAE and RMSE values on the test set, indicating more accurate predictions. This can be attributed to the ability of BiLSTMs to capture dependencies from both past and future time steps, providing a richer context for each prediction.

D.2. Loss Curves

Training and validation loss curves of LSTM and BiLSTM models are provided in Fig. 4 and 5.

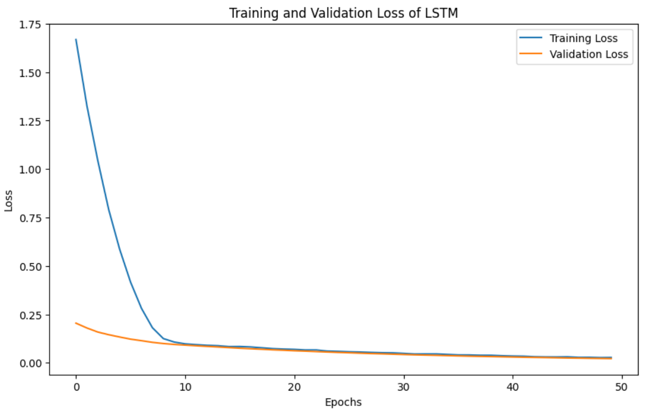

Fig. 4. Training and Validation Loss Curves of LSTM

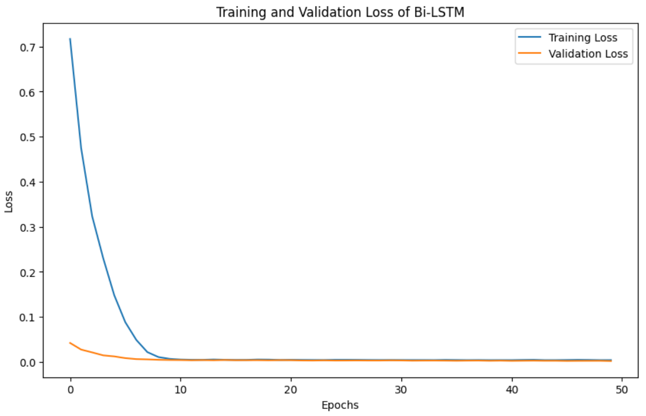

Fig. 5. Training and Validation Loss Curves of BiLSTM

Analyzing the training and validation loss curves provides further insights into the models' performance. The BiLSTM model's loss curves show a rapid decrease during the initial epochs, with both training and validation losses stabilizing at lower values. This close alignment between the training and validation loss curves suggests that the BiLSTM model generalizes well to unseen data, effectively capturing the underlying patterns in the stock price movements without overfitting.

In contrast, the LSTM model's loss curves reveal a different trend. Although the training loss decreases rapidly, indicating effective learning from the training data, the validation loss stabilizes at a higher value. This discrepancy between the training and validation losses suggests potential overfitting, where the model captures noise rather than generalizable patterns.

The superior performance of the BiLSTM model has several implications:

- **Richer Context**: By processing the sequence in both directions, BiLSTMs leverage future information that is particularly valuable in time series prediction tasks. This ability to capture bidirectional dependencies is a key factor in the improved performance.

- **Generalization**: The better generalization ability of the BiLSTM model, as evidenced by the validation loss, suggests that BiLSTMs are more robust to unseen data and less prone to overfitting compared to standard LSTMs.

In conclusion, the BiLSTM model outperforms the standard LSTM model in predicting stock prices, with lower MAE and RMSE values and better training dynamics. This demonstrates the value of using bidirectional architectures for time series prediction tasks, where understanding the context from both past and future time steps can significantly enhance model performance.

E. *Conclusion*

This study set out to address the complex challenge of stock price prediction in the dynamic and volatile financial markets. The primary aim was to enhance prediction accuracy by leveraging advanced deep learning techniques, specifically through a comparative analysis of Bi-Directional Long Short-Term Memory (Bi-LSTM) networks and traditional Long Short-Term Memory (LSTM) models. Using a comprehensive dataset of daily trading information for THYAO stock from the Istanbul Stock Exchange spanning from 2020 to 2024, we sought to determine which architecture better captures the intricacies of stock market data.

Our approach began with meticulous data collection and preprocessing, including the incorporation of key financial metrics and technical indicators like Exponential Moving Average (EMA) and Relative Strength Index (RSI). The data was normalized using Min-Max scaling to ensure uniformity across different features. We then structured the data into sequences using a sliding window approach, essential for time-series analysis, and split the data into training, validation, and test sets to maintain temporal integrity.

Both LSTM and BiLSTM models were developed, with their hyperparameters optimized using Optuna, a robust automated hyperparameter tuning framework. The models were trained and evaluated using well-known metrics, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), to assess their predictive accuracy. Our findings revealed that the BiLSTM model significantly outperformed the traditional LSTM model, achieving lower MAE and RMSE values on the test set. Specifically, the BiLSTM model demonstrated a MAE of 0.0582 and a RMSE of 0.0729, while the LSTM model showed a MAE of 0.1319 and a RMSE of 0.1481.

The superior performance of the BiLSTM model can be attributed to its ability to process data in both forward and backward directions, capturing bidirectional dependencies and providing a richer temporal context for each prediction. This capability is particularly beneficial in time-series forecasting, where understanding patterns from both past and future data points can significantly enhance prediction accuracy.

The results of this study directly address the core research question: "Which deep learning architecture, Bi-Directional LSTM or traditional LSTM, provides more accurate stock price predictions for THYAO stock?" The evidence clearly supports the superiority of the BiLSTM model, demonstrating its enhanced capability in capturing complex temporal patterns and providing more accurate stock price predictions.

In summary, this project not only showcases the potential of advanced deep learning models in financial time-series forecasting but also highlights the practical benefits of using BiLSTM networks over traditional LSTM models. The insights gained from this study can aid investors, traders, and financial analysts in making more informed and strategic decisions. Furthermore, the methodological framework and findings contribute to the growing body of knowledge on the application of deep learning in financial markets, paving the way for future research and development in this critical area.

F. *Future Works*

While this study has demonstrated the superior performance of Bi-Directional LSTM (BiLSTM) networks over traditional LSTM models in predicting stock prices, there are several avenues for future research and improvements that could further enhance the accuracy and robustness of these models.

F.1. Adaptive Window Size

One promising direction for future work is the implementation of adaptive window sizes. In this study, a fixed window size was used to create sequences for the LSTM and BiLSTM models. However, financial markets are inherently dynamic, with varying levels of volatility over different periods. An adaptive window size approach would allow the model to adjust the window length based on market conditions, capturing more relevant patterns during periods of high volatility and smoothing out noise during more stable periods. This adaptability could lead to improved model performance by providing a more tailored and responsive sequence input to the neural networks.

F.2. Model Ensemble Techniques

Future research could explore ensemble techniques, combining multiple models to leverage their individual strengths and mitigate their weaknesses. For instance, combining BiLSTM with other architectures like Convolutional Neural Networks (CNNs) for feature extraction or Attention mechanisms for better handling of long-term dependencies could lead to more robust and accurate predictions.

References

[1] S. L. Jain, R. Gupta, and A. A. Moghe, “Stock Price Prediction on Daily Stock Data using Deep Neural Networks,” *2018 International Conference on Advanced Computation and Telecommunication (ICACAT)*, Dec. 2018, doi: https://doi.org/10.1109/icacat.2018.8933791.

[2] S. Mootha, S. Sridhar, R. Seetharaman, and S. Chitrakala, “Stock Price Prediction using Bi-Directional LSTM based Sequence to Sequence Modeling and Multitask Learning,” *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, Oct. 2020, doi: https://doi.org/10.1109/uemcon51285.2020.9298066.

[3] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, “Stock price prediction using LSTM, RNN and CNN-sliding window model,” *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Sep. 2017, doi: https://doi.org/10.1109/icacci.2017.8126078.