**REPUBLIC OF TURKEY**

**ADANA ALPARSLAN TÜRKEŞ SCIENCE AND TECHNOLOGY UNIVERSITY**

**FACULTY OF ENGINEERING**

**DEPARTMENT OF COMPUTER ENGINEERING**

# AUTONOMOUS STOCK TRADING USING LSTM MODELS

**AHMET BURAK BİÇER**

**BACHELOR DEGREE**

**ADANA 2025**

**REPUBLIC OF TURKEY**

**ADANA ALPARSLAN TÜRKEŞ SCIENCE AND TECHNOLOGY UNIVERSITY**

**FACULTY OF ENGINEERING**

**DEPARTMENT OF COMPUTER ENGINEERING**

# AUTONOMOUS STOCK TRADING USING LSTM MODELS

**AHMET BURAK BİÇER**

**BACHELOR DEGREE**

**SUPERVISOR**

**ASSOC. PROF. DR ALİ İNAN**

**ADANA 2025**

# ABSTRACT


**AUTONOMOUS STOCK TRADING USING LSTM MODELS**



Ahmet Burak BİÇER
Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Ali İNAN


01 2025, 35 pages



This project presents the development of an autonomous stock trading application that leverages LSTM (Long Short-Term Memory) models for predictive modeling in the stock market. Focusing on selected stocks (AEFES, ASELS, THYAO, AFYON) from Borsa Istanbul, the application uses historical stock data and is strengthened with technical indicators such as Average True Range (ATR), Exponential Moving Average (EMA) and various lag features to estimate the future closing prices. The main idea of the application is the implementation of LSTM models trained to capture temporal dependencies within time-series data and provide a robust prediction for the stock price movements. This trading strategy uses predictions to update stop-loss and take-profit levels based on market volatility, measured by ATR. It can be tuned to most market conditions. You can modify the settings of the strategy: starting balance, capital allocation, transaction costs, trade sizes, and risk limits. That is very flexible for testing and fine-tuning.

It includes an interactive user interface developed using Dash and Plotly for visualizing simulation results on portfolio performance, balance history, portfolio composition, and comparative analyses between the predicted and actual stock prices. This backtesting simulation will help to show the potential effectiveness of using LSTM-based predictions in automated trading systems, based on key performance metrics such as return on investment (ROI), Sharpe Ratio, and maximum drawdown.

This project tests the feasibility of combining advanced machine learning techniques with algorithmic trading strategies in emerging markets.


**Keywords:** Borsa Istanbul, LSTM, ATR, EMA, Dash, ROI, Sharpe Ratio

# ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to everyone who has supported and contributed to the successful completion of this project.

First and foremost, I am profoundly thankful to my supervisor, Ali İnan for his exceptional guidance and insightful feedback throughout the development of this application.

Finally, I would like to express my deep appreciation to my family for their constant encouragement, understanding, and support. Their belief in me has been a source of strength and motivation, driving me to overcome challenges and strive for excellence.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# NOMENCLATURE

AEFES: Stock symbol for Anadolu Efes Biracılık ve Malt Sanayii A.Ş., a major brewery company listed on the Turkish stock exchange.

AFYON: Stock symbol for Afyon Çimento Sanayi T.A.Ş., a cement manufacturing company listed on the Turkish stock exchange.

ASELS: Stock symbol for Aselsan Elektronik Sanayi ve Ticaret A.Ş., a leading defense electronics company in Turkey.

ATR (Average True Range): A technical analysis indicator that measures market volatility by decomposing the entire range of an asset price for a given period.

Backtesting: The process of testing a trading hypothesis or model using historical data to estimate how accurately the model or strategy would perform in real trading.

Dash: An open-source Python framework for building analytical web applications, particularly useful for data visualization.

Drawdown: A measure of decline from a historical peak in some variable (typically the cumulative profit or total open equity of a financial trading strategy).

EMA (Exponential Moving Average): A type of moving average that places a greater weight and significance on the most recent data points.

Lag Features: Features that use previous time steps (lags) in time series data to predict future values.

LSTM (Long Short-Term Memory): A type of recurrent neural network architecture that widely used in time series forecasting. capable of learning order dependence in sequence prediction problems.

ROI (Return on Investment): A performance measure used to evaluate the efficiency or profitability of an investment, calculated as net profit divided by the initial cost of the investment.

Sharpe Ratio: A measure for calculating risk-adjusted return, indicating the average return earned in excess of the risk-free rate per unit of volatility or total risk.

THYAO: Stock symbol for Türk Hava Yolları A.O., the national flag carrier airline of Turkey, listed on the Turkish stock exchange.

Time Series Data: A sequence of data points collected or recorded at time-ordered intervals.

Transaction Cost: Expenses incurred when buying or selling securities, including broker commissions and fees, which affect net investment returns.

VWAP (Volume Weighted Average Price): A trading benchmark that calculates the average price at which a security has traded throughout the day, based on both volume and price.

Window Size: In time series analysis, the number of time steps used as input features for forecasting the next time step.

Plotly: An interactive graphing library for Python that enables the creation of dynamic visualizations.

# 1. INTRODUCTION

The proliferation of data and computational powers deeply changed financial markets, and the new era of complex algorithmic trading systems was born. These systems use mathematical models in combination with computational algorithms to execute trades at speeds and frequencies that far exceed human capability, often capitalizing on market inefficiencies and short-term fluctuations. In this domain, machine learning has become more prominent, especially deep learning techniques, because of their capability to model complex patterns and temporal dependencies inherent in financial time series data.

Stock price prediction is inherently a very challenging task due to the stochastic and chaotic nature of financial markets. Traditional statistical models lack the capability to capture nonlinear relationships and dynamic temporal patterns that are usually present in stock market data. Recurrent Neural Networks (RNN), especially the Long Short-Term Memory networks, have been quite efficient in modeling sequential data by learning long-term dependencies without the vanishing gradient problem as seen in standard RNNs. This model would be particularly applicable to financial time series forecasting because, in such a series, the historical trend and pattern are key to making an educated forecast.

This project focuses on developing an autonomous stock trading application that integrates LSTM neural networks for forecasting future stock prices. The application is designed for a specific sample portfolio comprising ASELS, THYAO, AEFES, and AFYON, all listed on Borsa Istanbul (the Turkish stock exchange). These stocks were deliberately selected to evaluate the application's performance across companies from various sectors with differing success levels. The dataset utilized covers the period from November 30, 2015, to December 21, 2021, and was sourced from Kaggle. It is an aggregated daily numerical dataset of opening and closing prices, high and low prices, trade volumes, and other financial metrics stored in CSV format.

Another important part of the project was feature engineering and selection. A set of technical indicators and lag features was created, reflecting both the short-term and long-term trading activities to capture more information and strengthen the predictive capability of the model. Some techniques involved in this were moving averages over 7, 14, and 21 days, Exponential Moving Average (EMA), Average True Range (ATR), and differences between prices. Feature selection was performed using methods like SelectKBest with F-regression and Recursive Feature Elimination, RFE, coupled with a Random Forest Regressor.

The LSTM model architecture was carefully designed to capture the temporal dynamics of stock prices. The model includes convolutional, LSTM and dense layers to extract meaningful patterns from the sequential data. The model was trained using the Huber loss function and the SGD optimizer with momentum. A learning rate scheduler was implemented to adjust the learning rate during training, enhancing convergence. The autonomous trading strategy utilizes the predictions from the LSTM model to make buy and sell decisions. The strategy incorporates dynamic thresholds based on the ATR to adjust for market volatility, and risk management parameters such as position size percentage, stop-loss percentage, and take-profit percentage are configurable. To evaluate the performance of the trading strategy some metrics are calculated such as total profit, return on investment (ROI), final balance, maximum profit achieved during simulation, sharpe ratio, maximum drawdown.

### 1.1. Challenges and Limitations

One major issue with this project is that the model will not be able to explain unexpected changes because of news or external factors affecting the market. Everything-big and small-sets the financial market running: political events, economic indicators, and even sudden company announcements. Since the model was trained only on historical numerical data and doesn't include real-time news sentiment analysis, it can't predict scenarios caused by the release of unexpected news. This limits the approach of integrating other sources of data with the model in an effort to achieve a wide catch of market influencers.

## 1.2. Objectives and Contributions

- Developing Predictive Models: Construct LSTM models capable of forecasting future closing prices using historical data and engineered features.
- Implementing an Autonomous Trading Strategy: Design a trading system that utilizes model predictions to execute trades with risk management protocols.
- Evaluating Performance: Evaluate the effectiveness of predicting models with error functions such as, Mean Absolute Error (MAE), Mean Square Root Error (MSE), Root Mean Square Root Error (RMSE) and evaluate the trading strategy through backtesting and analyzing key performance metrics.

This project contributes to the demonstrations of deep learning model applications in the context of an emerging market context using and automated trading approach. Currently, the focus is on the Turkish stock exchange, and it includes detailed feature engineering and model optimization, therefore providing insight into some of the practical considerations and potential associated with machine learning integrated into algorithmic trading strategies.

## 1.3. Structure of the Report

The remainder of this report is organized as follows:

- Literature Review: An examination of existing research on stock price prediction and algorithmic trading using machine learning.
- Methodology: A detailed explanation of the data collection, preprocessing, feature engineering, model development, and trading strategy implementation.
- Results: Presentation of the simulation outcomes, performance metrics, and visualizations.
- Conclusions: Summary of findings, contributions, and suggestions for future research directions.

# 2. LITERATURE REVIEW

## 2.1. Traditional vs Machine Learning Approaches

Traditional models are ARIMA for capturing the linear pattern and GARCH for volatility. Both perform poorly when it comes to capturing nonlinear complexities in financial data. To handle this, Support Vector Machines, Random Forest, and Gradient Boosting Machines have been used as some machine learning methods; they tend to outperform traditional methods under specific conditions. All these methods lack the capability to model sequential data, which has made the deep learning approach more suitable.

## 2.2. Neural Networks and LSTM Models

Recurrent Neural Networks, while designed for sequential data, have several flaws, such as the vanishing gradient problem, which makes them not quite efficient for long-term dependencies. This is fixed by LSTM networks through memory cells and gating mechanisms. Researchers point out that this is one reason behind the superior capture of temporal pattern variations in LSTMs, therefore guaranteeing better predictive performance, especially in combination with appropriate technical indicators like moving averages, such as Fischer and Krauss, 2018 or Zhang et al., 2017.

## 2.3. Hybrid Models: CNN and LSTM

While combining CNNs with LSTM networks leverage both spatial and temporal features of financial data, a combination of such in a hybrid model-which, for example, can be instantiated with the CNN-LSTM framework proposed by Adebiyi et al. (2020)-outperforms any standalone model by extracting salient features with CNNs and modeling temporal dependencies with LSTMs. That is what has been done in this project.

## 2.4. Algorithmic Trading and Automated Strategies

Algorithmic trading automates the decision-making process with predictive models and risk management mechanisms, such as stop-loss and take-profit orders. Research by Van Horne (2018) and Ortega and Kansa (2015) highlights that strategy development should consider transaction costs, volatility indicators, and realistic backtesting environments to enhance reliability. These elements are integral part of the trading algorithms in the project.

# 3. METHODOLOGY

The steps in developing the autonomous stock trading application will be explained here, starting with data acquisition, preprocessing, and feature engineering to model development and training procedures and, finally, how the trading strategy has been implemented. This methodology will be performed to ensure the robustness of the predictive models and the realism of the trading simulation.

## 3.1. Data

### 3.1.1. Data Acquisition

The dataset utilized in this project encompasses historical stock data for four prominent companies listed on Borsa Istanbul (the Turkish stock exchange):

- Aselsan Elektronik Sanayi ve Ticaret A.Ş. (ASELS): A leading Turkish defense electronics company specializing in advanced technology systems. It provides solutions for military and civil industries worldwide.
- Türk Hava Yolları A.O. (THYAO): Turkish Airlines is Turkey's national flag carrier and one of the world's largest airlines by passenger numbers. It operates flights to over 300 destinations globally.
- Anadolu Efes Biracılık ve Malt Sanayii A.Ş. (AEFES): Anadolu Efes is a major beverage producer in Turkey, primarily known for its beer and malt production. It operates in multiple international markets across Europe and Central Asia.
- Afyon Çimento Sanayi T.A.Ş. (AFYON): A prominent cement manufacturer in Turkey, providing materials for the construction industry. It plays a key role in infrastructure and urban development projects.

Note: Companies from different sectors were selected to better understand the success of the model. While ASELS, THYAO, AEFES can be classified as successful and profitable companies for their investors, AFYON company has been determined to cause losses to its investors in the long run and was deliberately selected.

Data was obtained from publicly available sources, specifically from Kaggle, covering the period from November 30, 2015, to December 21, 2021.

**Table 1** Data card

| # | Column | Description |
|---|---|---|
| 1 | TRADE DATE | Date of the trading day |
| 2 | INSTRUMENT SERIES CODE | Serial code contained in the instrument code |
| 3 | INSTRUMENT NAME | Long ID |
| 4 | MARKET SEGMENT | Market segment categories (e.g., BIST Stars, Emerging Companies, etc.) |
| 5 | MARKET | Market categories (e.g., Main Spot, Primary Market, etc.) |
| 6 | INSTRUMENT TYPE | Type of related instrument (e.g., BUYINEQT, MSPOTECW, etc.) |
| 7 | INSTRUMENT CLASS | Class of related instrument (e.g., BUYINEQTGARAN, MSPOTEQTGARAN, etc.) |
| 8 | MARKET MAKER | Availability of market maker (0: Not available, 1: Available) |
| 9 | BIST 100 INDEX | Inclusion in BIST 100 Index (0: Not available, 1: Available) |
| 10 | BIST 30 INDEX | Inclusion in BIST 30 Index (0: Not available, 1: Available) |
| 11 | GROSS SETTLEMENT | Gross Settlement (1: Applicable) |
| 12 | SUSPENDED | Suspension status (0: Not Suspended, 1: Suspended) |
| 13 | OPENING PRICE | First price of the day |
| 14 | OPENING SESSION PRICE | Price executed in opening session |
| 15 | LOWEST PRICE | Lowest price of the day |
| 16 | HIGHEST PRICE | Highest price of the day |
| 17 | CLOSING PRICE | Last price of the day |
| 18 | CLOSING SESSION PRICE | Price executed in closing session |
| 19 | CHANGE TO PREVIOUS CLOSING (%) | % change relative to the previous day's closing price |
| 20 | REMAINING BID | Best bid price |
| 21 | REMAINING ASK | Best offer price |
| 22 | VWAP | Daily volume-weighted average price |
| 23 | TOTAL TRADED VALUE | Total traded value of related equity |
| 24 | TOTAL TRADED VOLUME | Total traded volume of related equity |
| 25 | TOTAL NUMBER OF CONTRACTS | Total number of contracts traded |
| 26 | REFERENCE PRICE | Exchange-announced price for market insight |
| 27 | TRADED VALUE AT OPENING SESSION | Traded value in opening session |
| 28 | TRADED VOLUME AT OPENING SESSION | Traded volume in opening session |
| 29 | NUMBER OF CONTRACTS AT OPENING SESSION | Number of contracts in opening session |
| 30 | TRADED VALUE AT CLOSING SESSION | Traded value in closing session |
| 31 | TRADED VOLUME AT CLOSING SESSION | Traded volume in closing session |
| 32 | NUMBER OF CONTRACTS AT CLOSING SESSION | Number of contracts in closing session |
| 33 | TRADED VALUE OF TRADES AT CLOSING PRICE | Traded value at closing price |
| 34 | TRADED VOLUME OF TRADES AT CLOSING PRICE | Traded volume at closing price |
| 35 | NUMBER OF CONTRACTS OF TRADES AT CLOSING PRICE | Contracts at closing price |
| 36 | LOWEST SHORT SALE PRICE | Lowest price of short sale trades |
| 37 | HIGHEST SHORT SALE PRICE | Highest price of short sale trades |
| 38 | SHORT SALE VWAP | VWAP of short sale trades |
| 39 | TRADED VALUE OF SHORT SALE TRADES | Traded value of short sale trades |
| 40 | TRADED VOLUME OF SHORT SALE TRADES | Traded volume of short sale trades |
| 41 | NUMBER OF CONTRACTS OF SHORT SALE TRADES | Number of contracts of short sale trades |
| 42 | LOWEST TRADE REPORT PRICE | Lowest trade report price |

| | | |
|---|---|---|
| 43 | HIGHEST TRADE REPORT PRICE | Highest trade report price |
| 44 | TRADE REPORT VWAP | VWAP of trade reports |
| 45 | TRADE REPORT TRADED VALUE | Total traded value of trade reports |
| 46 | TRADE REPORT TRADED VOLUME | Total traded volume of trade reports |
| 47 | NUMBER OF TRADE REPORTS | Total number of trade reports |

The data card above demonstrates every feature that included in the raw data and their explanations. Each stock's data was stored in separate CSV files and imported using the Pandas library for analysis and preprocessing.

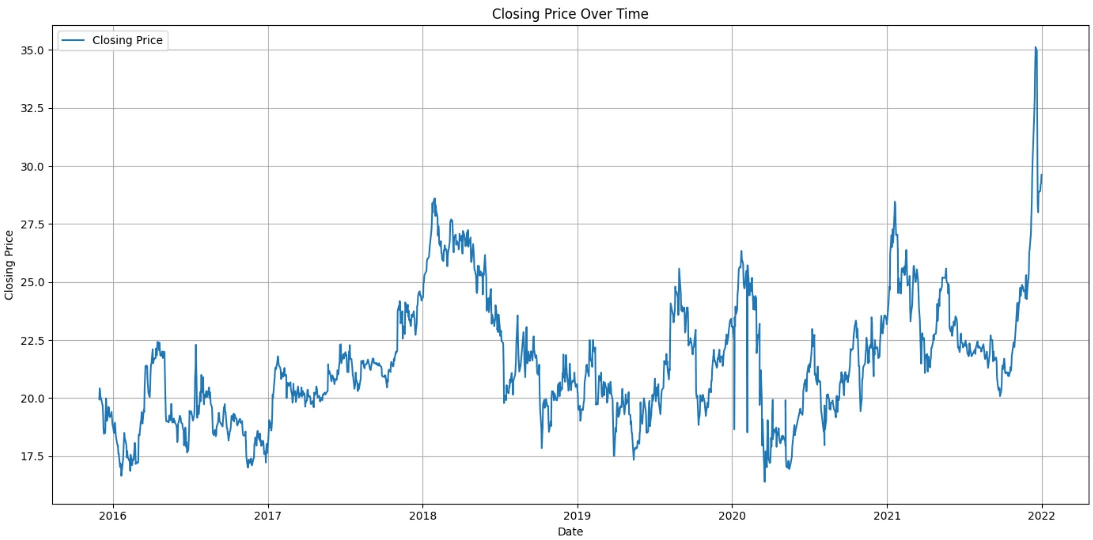


**Figure 1** Data visualization for AEFES

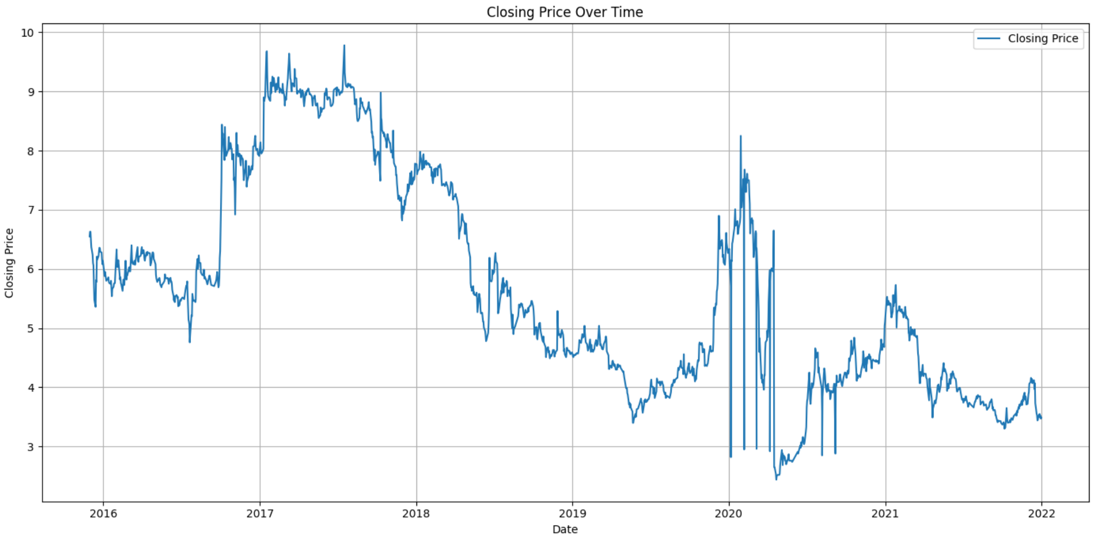


**Figure 2** Data visualization for AFYON

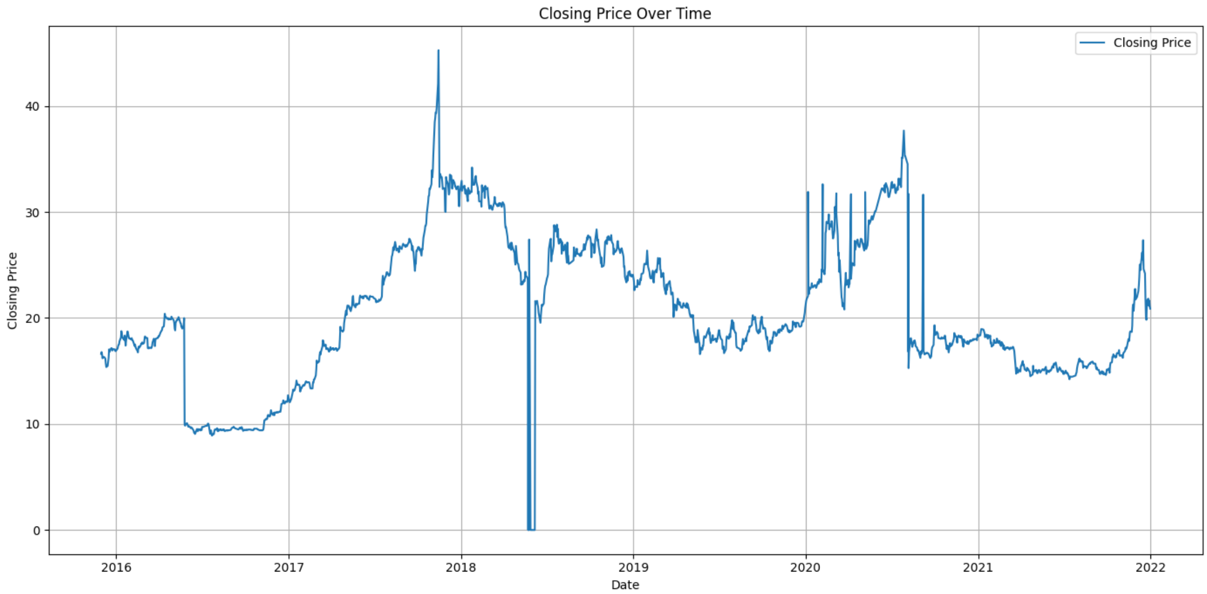


**Figure 3** Data visualization for ASELS

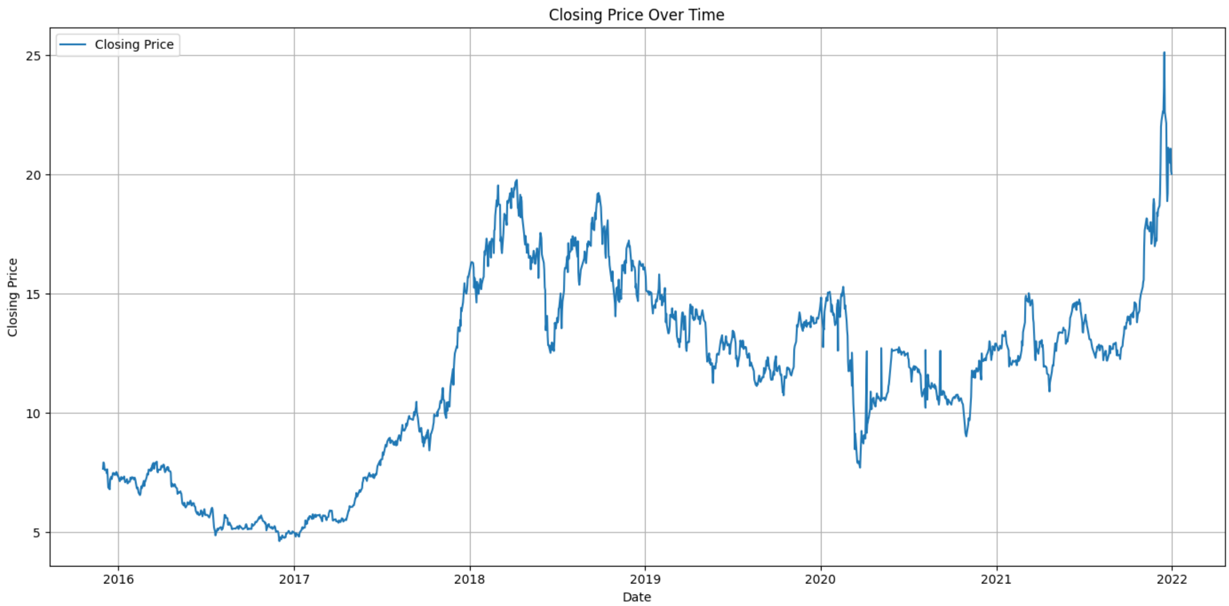


**Figure 4** Data visualization for THYAO

### 3.1.2. Data Preprocessings

First of all, doing some effective preprocessing on the data to prepare the dataset for modeling is important. This includes loading each individual dataset for a stock, adding a new column named 'Stock' representing the stock symbol, and concatenating these DataFrames to get a combined dataset that includes all four stocks. First, the dataset of each stock was checked for missing or null values, and all such rows were removed to retain integrity. The 'TRADE DATE' columns were converted to datetime format to enable time-series analysis, and the data was sorted by 'TRADE DATE' and 'Stock' to maintain the sequence of events for each stock. This first rescaled features using MinMaxScaler to limit the feature values from 0 to 1, part of sklearn library, and facilitated smooth convergence in models through standard scaling.

### 3.1.3. Feature Engineering

Feature engineering is the next step to better the model on the predictability of multiple stocks. Much emphasis will, therefore, be laid on the creation of technical indicators and lag features.

Firstly, the moving averages are computed: the 7-day, 14-day, and 21-day simple moving averages of the closing price for each stock that capture the short and medium-term trends.

$$Moving\ Average = \frac{1}{n} \times \sum_{i=1}^{n} (n_i) \tag{1}$$

Also, the exponential moving average of 7 days has been calculated to give more importance to recent price data.

Formula for the exponential moving average:

$$\begin{aligned} &\alpha = 2/(span + 1), \qquad for\ span \geq 1 \\ &y_0 = x_0 \\ &y_t = (1 - \alpha)y_{t-1} + \alpha x_t \end{aligned} \tag{2}$$

ATR indicator for each stock in a 14-day window to measure the volatility of the market, which is one of the key variables for setting dynamic thresholds for trading. It also calculates the 7-day moving average of the Volume Weighted Average (VWAP) Price to include both aspects of price and volume.

To model temporal dependencies and momentum, lag features are also created. For each stock, it creates lag versions of the closing price up to 7 days and builds lagged features of total traded volume and total traded value to include historical trading activities. It included price differentials: daily differences of closing and opening prices, the difference between the current day's opening price and the previous day's opening price. The daily price range was also included in the dataset, computed as a difference between the highest and lowest prices for each stock. Lastly, lagging was applied to the technical indicators, enabling the model to make better predictions by including some historical context.

### 3.1.4. Feature Selection

Feature selection was crucial, considering that the feature columns was increased with many additional features, in order to understand the most influential features across all stocks.

The idea here is using Scikit Learn's feature selection module, SelectKBest with the F-test for regression; this method selects some of the best features, in terms of which had the best correlation with the target variable that was the closing price for every stock. Then, Recursive Feature Elimination has also been done with Scikit Learn's ensemble module's RandomForestRegressor. This does eliminate less important features recursively until the most influencing features for each of the stocks are found.

Finally a common feature set obtained by intersecting the best features chosen for each stock allowed this model to learn the features which are important consistently for several stocks, hence increasing the generalization and predictive performance of the model.

While picking the most important features, I detected features that were very related to each other using a heatmap and did not use these features to avoid biasing the model.

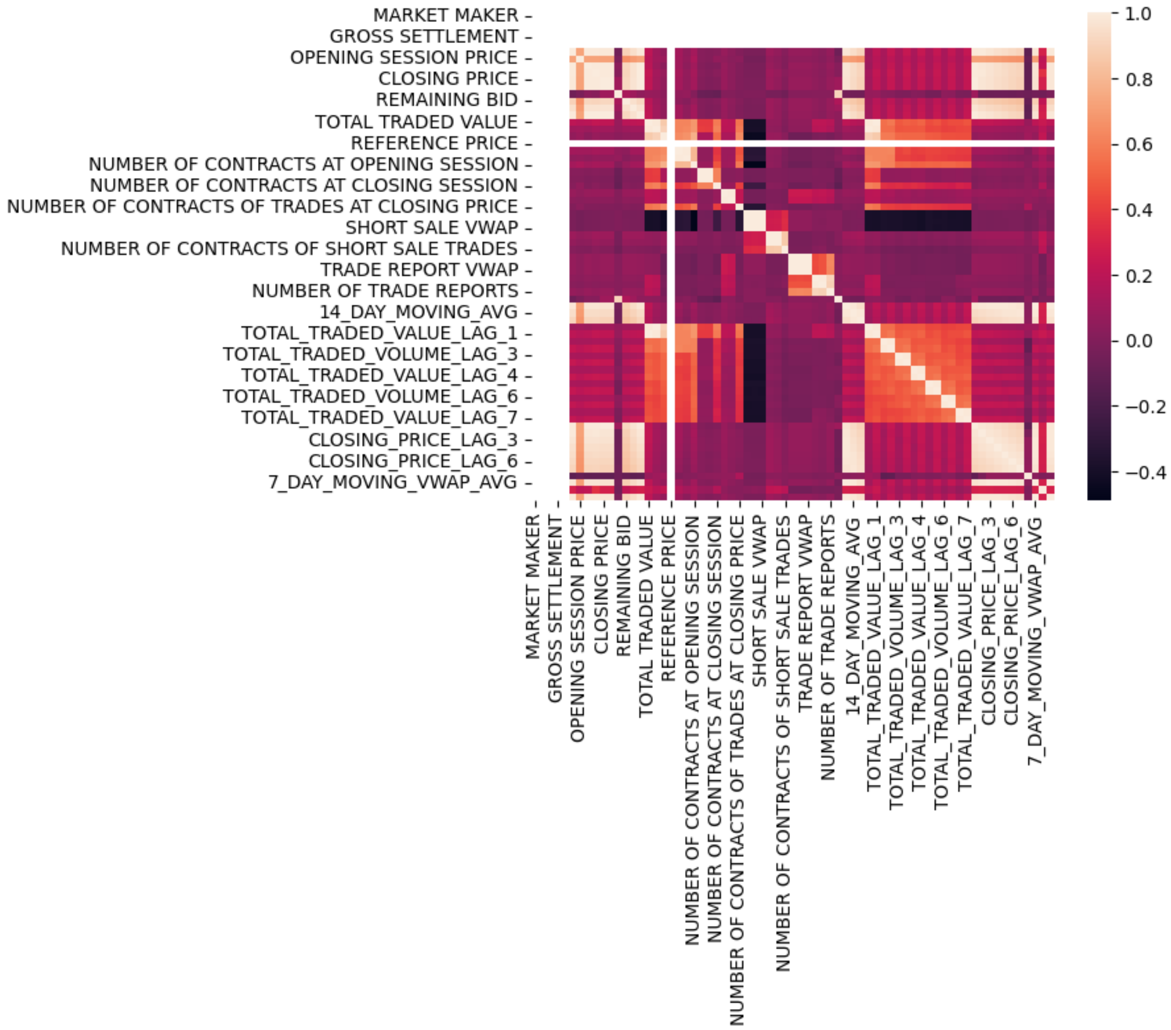


**Figure 5** Correlation Heatmap

### 3.2. Model

Separate LSTM models were developed for each stock to capture stock-specific patterns and dynamics while maintaining a consistent architecture across models.

For each stock, the model architecture included:

- Conv1D Layer:
  - One convolution layer with 64 filters, 3 kernel size, 1 strides, casual padding and relu activation function. Extracts local features from the input sequences, aiding in pattern recognition.
- LSTM Layers:
  - Two stacked LSTM layers with 128 units each to capture long-term dependencies in the time series data for each stock.
- Dense Layers:
  - One dense layer with 32 units and relu activation function
  - One dense layer with 16 units and relu activation function
  - One dense layer with 1 unit without activation function (linear). Maps the extracted features to the output (predicted closing price).

For each stock, the data was transformed into sequences suitable for LSTM input. Each stock's data was split into training and testing sets with an 80-20 ratio. So, the test data covers 28 October 2020 to 31 May 2021. Trading in the simulation is happening during this time period.

For each stock, hyperparameter tuning was conducted to optimize model performance. A learning rate scheduler was employed to determine the initial learning rate, with the value corresponding to the lowest loss selected as the starting point.

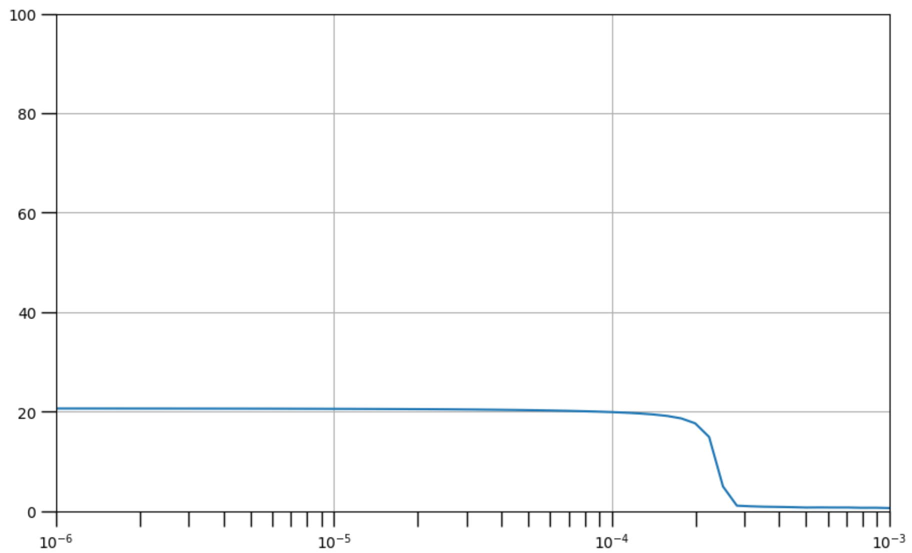


**Figure 6** Finding optimal learning rate

The y axis of the above graph represents the loss value and the x axis represents the learning rate. According to the this graph the optimal learning rate is approximately 0.0071.

Once the initial learning rate was established, a secondary learning rate scheduler was configured for the main model training process.

During training, the Huber loss function was utilized to mitigate the influence of outlier values.

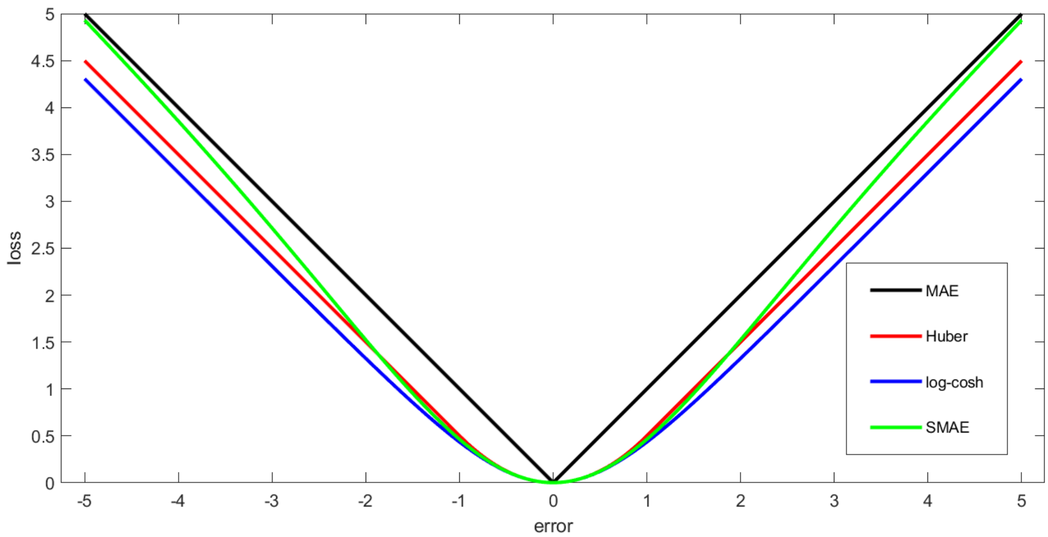


**Figure 7** Comparison of loss functions

The optimization process relied on Stochastic Gradient Descent (SGD) with a momentum value of 0.9. To prevent overfitting and achieve the optimal model, early stopping techniques were incorporated. Specifically, training was halted if no improvement in the validation MAE (Mean Absolute Error) was observed over 40 consecutive epochs.

Performance metrics were evaluated using Mean Absolute Error (MAE) for each stock's model to assess its accuracy. Validation Compared predicted closing prices with actual values. Additionally, learning curves were plotted to visualize and monitor performance throughout the training process.

### 3.3. Trading

Using the generated predictions of each stock, an autonomous trading strategy has been developed. The initial parameters used for this simulation include Initial Balance = 100,000 TL, transaction cost = 0.2%, position size percent = 10%, stop loss percent = 5%, and take profit percent = 10%. Dynamic thresholds are used with the Average True Range method to dynamically adjust the buying and selling thresholds for every share.

Trading logic was then based on well-defined entry and exit conditions: a buy signal occurred when the predicted price of a stock surpassed the buy threshold in case sufficient funds were available. A sell signal occurred when the predicted price fell below the sell threshold. The exit conditions included a stop-loss-a mechanism that was triggered when the current price fell to the level of stop-loss-and a take-profit condition once the price had risen to the take-profit level. The risk was managed by sizing the positions based on the position size percentage.

Trading simulation was done iteratively: it considered each trading day and analyzed every stock in them. With each evaluation, the portfolio would have updated dynamically. Also, detailed transaction logs were recorded down: a stock symbol, transaction type, quantity, price, total cost or revenue. The value of the total portfolio was calculated by summing up cash and positions' market values.

These performance metrics comprised the following financial indicators of the trading strategy: the total profit or loss derived by the difference between the final value of the portfolio and its initial balance, return on investment expressed in percent; the Sharpe ratio that had been calculated on a daily return basis to assess risk-adjusted performance; maximum drawdown illustrating peak-to-trough decline of the portfolio; and stock earnings per share.

### 3.4. Interface

A web application was developed using Dash and Plotly to provide users with a friendly interface where they can choose which stocks to include in the simulation, adjust parameters, and start the trading simulation. The application is specifically designed to include heavy visualization that makes it more interactive for its users and to understand the data on a full-scale basis. It incorporates a balance chart that maps portfolio value over time, hence clearly showing the performance thereof. Portfolio composition shows the user the current holdings. A price comparison tool helps the user see the predicted and actual prices for each stock, giving an idea about how good or bad the predictive models were doing. Furthermore, the application adds support for a transaction summary that tracks every transaction made within this simulated environment; that way, full details of all the financial outcomes or performance metrics resulting from a trading strategy can be appraised by its user.

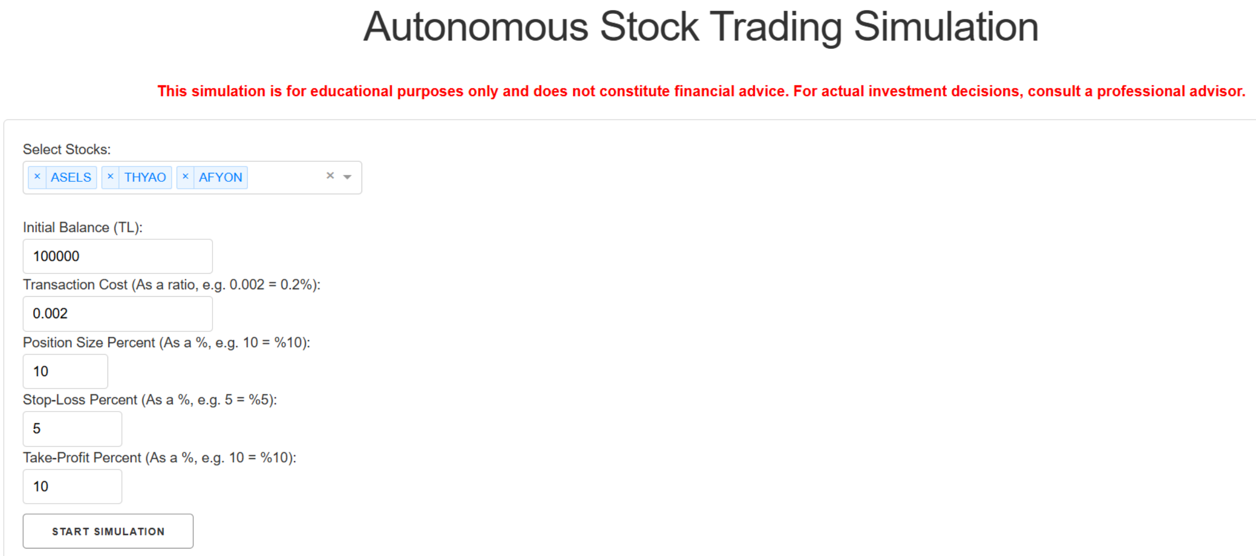


**Figure 8** Application interface

# 4. RESULTS

The following section presents the result of a simulated backtest regarding an independent stock trading application. Result interpretations cover financial performances of metrics, evolution of a portfolio, and also efficacy at prediction, along with necessary visualization of outcomes or an elaboration discussion in using such a strategy in light of ASELS, THYAO, AEFES, and AFYON stock securities.

## 4.1. Predictive Model Performance

Separate Long Short-Term Memory (LSTM) models were trained for each stock using historical data up to 80% of the dataset for training and the remaining 20% for testing. The models aimed to predict the closing prices based on the engineered features.

In all the stocks, the training and validation MAE decreased with the epochs, meaning that the models were learning well. Early stopping was employed to prevent overfitting, with patience set to ensure the models did not stop training too early.

The final MAE, MSE, and RMSE on the test set for each stock were as follows:

**Table 2** Model evaluation results

| Stock | MAE | MSE | RMSE |
|---|---|---|---|
| ASELS | 0.3169 | 0.2575 | 0.5074 |
| THYAO | 0.3064 | 0.2622 | 0.5120 |
| AEFES | 0.4566 | 0.5370 | 0.7328 |
| AFYON | 0.1074 | 0.0239 | 0.1547 |

The relatively low MAE values indicate that the models were able to predict the closing prices with reasonable accuracy.

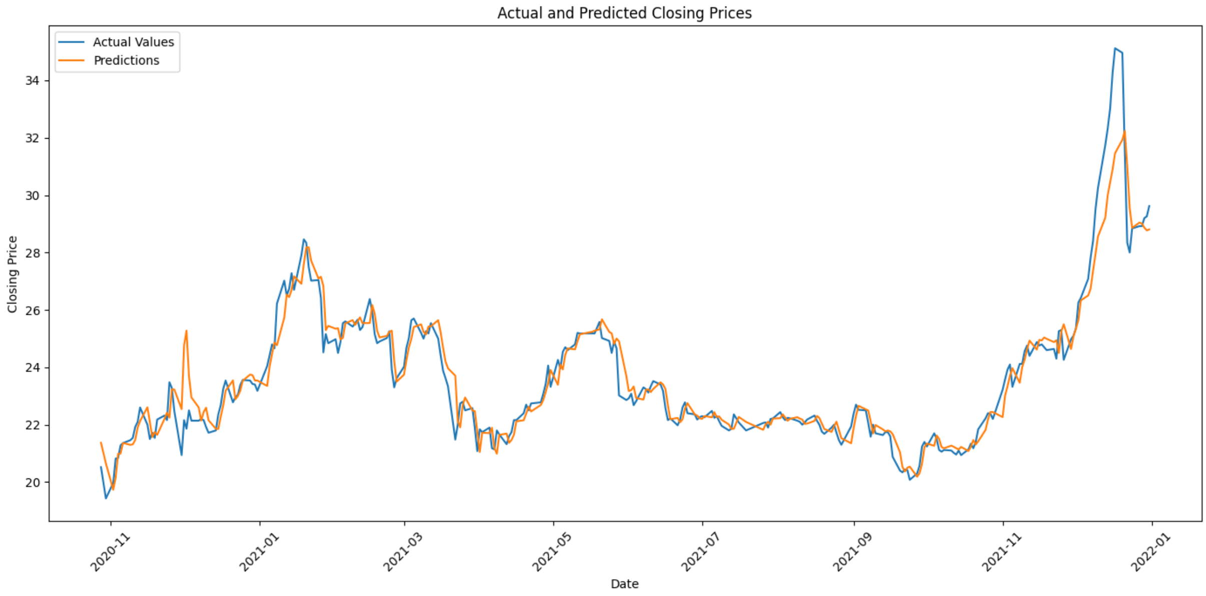


**Figure 9** Prediction for AEFES

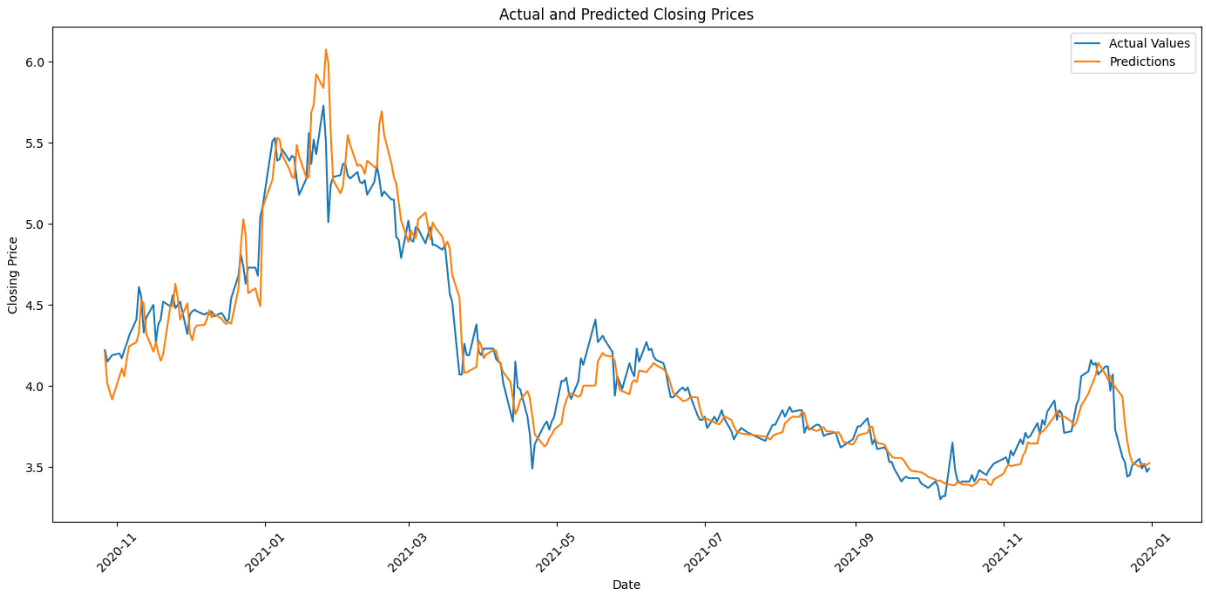


**Figure 10** Prediction for AFYON

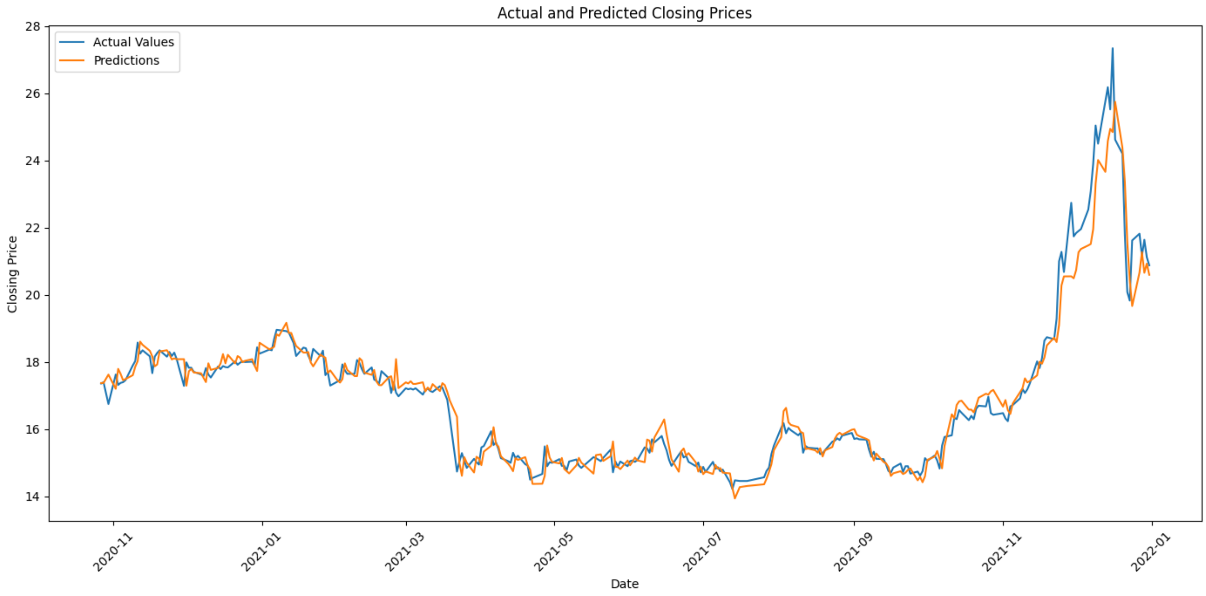


**Figure 11** Prediction for ASELS

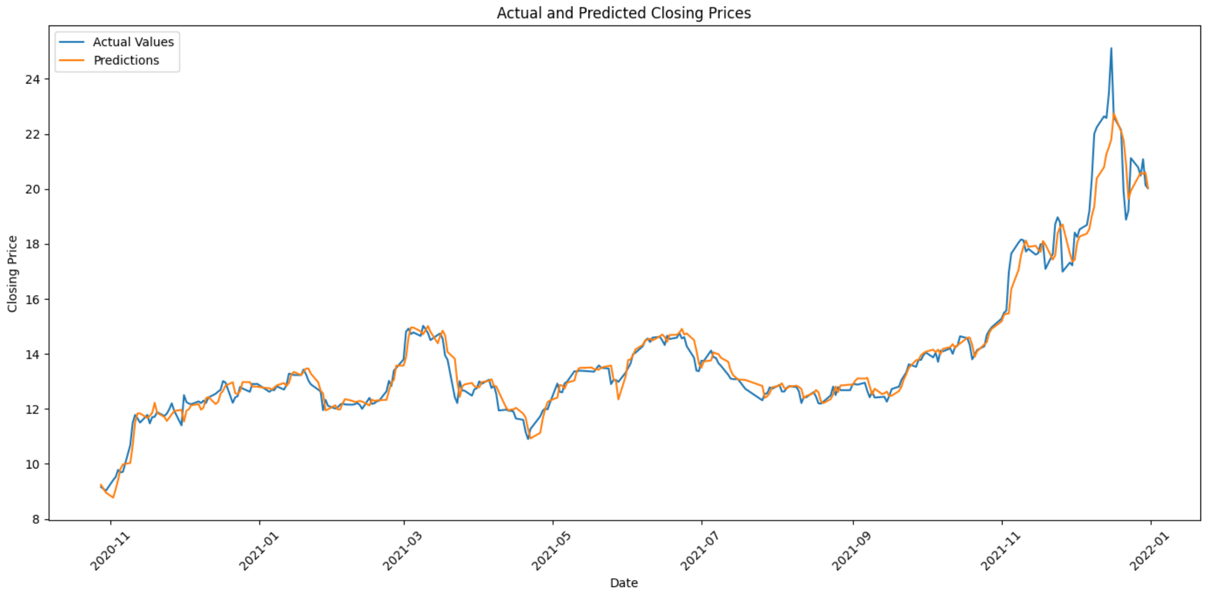


**Figure 12** Prediction for THYAO

## 4.2. Trading Simulation Outcomes

The following parameters will be used as the default for the simulation:

- Initial Balance: 100,000 TL
- Transaction Cost: 0.2% - to simulate the cost of brokerage and market impact
- Position Size Percentage: 10% - amount of total capital to allocate per trade
- Stop-Loss Percentage: 5% - to limit the potential loss per position
- Take-Profit Percentage: 10% - to secure profit once targets are reached
- Window Size for LSTM Models: 7 days - using the last 7 days to predict the next day

Using the forecasted prices of the LSTM models, a trading simulation was performed for various combinations of selected stocks. Three scenarios are discussed further in the following sections

1. Selected Stocks: ASELS, THYAO

2. Selected Stocks: ASELS, THYAO, AFYON

3. Selected Stocks: ASELS, THYAO, AFYON, AEFES

Each case shows a different aspect with regard to the influence of the added stocks on the overall performance of the trading strategy.

### 4.2.1. Scenario 1: ASELS and THYAO

**Overall Financial Performance**

- Total Profit/Loss: 4,742.42 TL
- Return on Investment (ROI): 4.74%
- Final Cash Balance: 104,742.42 TL
- Positions Held: None (All positions were closed by the end of the simulation)
- Total Value of the Portfolio: 104,742.42 TL
- Maximum Profit Achieved During Simulation: 5,909.78 TL
- Sharpe Ratio: 1.12
- Maximum Drawdown: -3.19%

The simulation using ASELS and THYAO resulted in a positive total profit of 4,742.42 TL, yielding an ROI of 4.74%. The Sharpe Ratio of 1.12 indicates a favorable risk-adjusted return, suggesting that the strategy effectively balanced risk and return.

**Earnings per Share and Trading Activity by Stock**

- ASELS:
    - Total Earnings: 543.06 TL
    - Total Shares Traded: 19,658

- Earnings per Share: 0.03 TL/share
- THYAO:
  - Total Earnings: 4,199.36 TL
  - Total Shares Traded: 21,977
  - Earnings per Share: 0.19 TL/share

**Analysis**

- Both stocks contributed positively to the total earnings, with THYAO accounting for the majority of the profit.
- The higher earnings per share for THYAO (0.19 TL/share) indicate that trades in THYAO were more profitable on a per-share basis compared to ASELS.
- The absence of open positions at the end of the simulation reflects the strategy's effectiveness in closing positions based on take-profit and stop-loss thresholds.

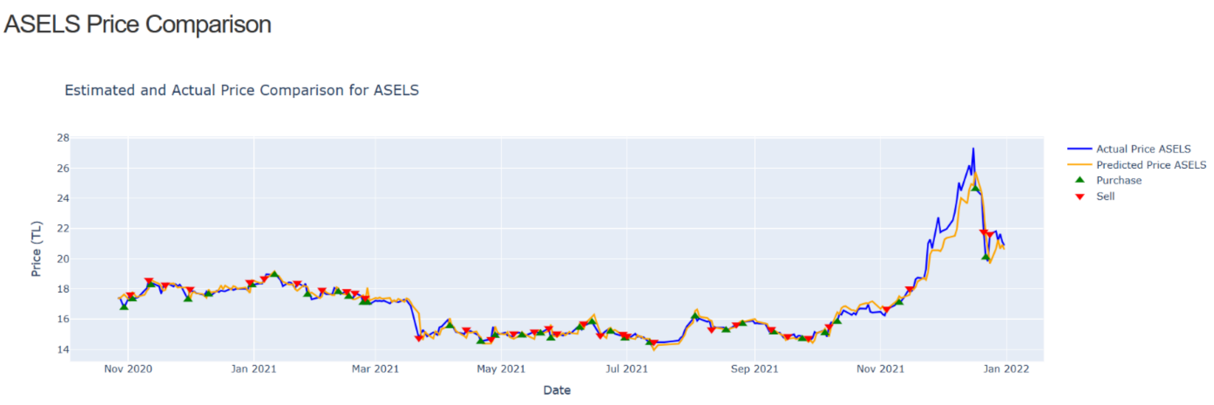


**Figure 13** Trading of ASELS


**Figure 14** Trading of THYAO


**Figure 15** Daily portfolio value while trading of ASELS and THYAO

### 4.2.2. Scenario 2: ASELS, THYAO and AFYON

**Overall Financial Performance**

- Total Profit/Loss: 4,010.37 TL
- Return on Investment (ROI): 4.01%
- Final Cash Balance: 93,561.31 TL
- Positions Held:
    - AFYON: 2,994 shares
- Total Value of the Portfolio: 104,010.37 TL
- Maximum Profit Achieved During Simulation: 5,671.83 TL
- Sharpe Ratio: 0.73
- Maximum Drawdown: -5.38%

With the addition of AFYON, the total profit decreased to 4,010.37 TL, resulting in an ROI of 4.01%. The Sharpe Ratio decreased to 0.73, indicating a lower risk-adjusted return compared to Scenario 1. The maximum drawdown increased to -5.38%, suggesting higher volatility and risk.

**Earnings per Share and Trading Activity by Stock**

- ASELS:
    - Total Earnings: 708.40 TL
    - Total Shares Traded: 18,652
    - Earnings per Share: 0.04 TL/share
- THYAO:
    - Total Earnings: 3,973.21 TL
    - Total Shares Traded: 20,797
    - Earnings per Share: 0.19 TL/share
- AFYON:
    - Total Earnings: -800.34 TL
    - Total Shares Traded: 61,405

    - Earnings per Share: -0.01 TL/share

**Analysis**

- While ASELS and THYAO continued to contribute positively, the inclusion of AFYON resulted in a loss of 800.34 TL.
- AFYON's negative earnings per share (-0.01 TL/share) indicate that trades in this stock were unprofitable.
- The holding of 2,994 shares of AFYON at the end of the simulation suggests that positions in AFYON were not closed, possibly due to the stock not reaching the take-profit or stop-loss thresholds.
- The increased maximum drawdown reflects the added volatility introduced by AFYON's performance.

### 4.2.3. Scenario 3: ASELS, THYAO, AFYON and AEFES

**Overall Financial Performance**

- Total Profit/Loss: 2,050.32 TL
- Return on Investment (ROI): 2.05%
- Final Cash Balance: 93,716.20 TL
- Positions Held:
    - AFYON: 2,388 shares
- Total Value of the Portfolio: 102,050.32 TL
- Maximum Profit Achieved During Simulation: 5,062.59 TL
- Sharpe Ratio: 0.34
- Maximum Drawdown: -7.70%

Including AEFES along with the previous stocks further reduced the total profit to 2,050.32 TL and the ROI to 2.05%. The Sharpe Ratio decreased significantly to 0.34, indicating a substantially lower risk-adjusted return. The maximum drawdown increased to -7.70%, pointing to greater volatility and risk exposure.

**Earnings per Share and Trading Activity by Stock**

- ASELS:
  - Total Earnings: 300.65 TL
  - Total Shares Traded: 17,801
  - Earnings per Share: 0.02 TL/share
- THYAO:
  - Total Earnings: 3,581.69 TL
  - Total Shares Traded: 19,823
  - Earnings per Share: 0.18 TL/share
- AEFES:
  - Total Earnings: -1,211.78 TL
  - Total Shares Traded: 11,448
  - Earnings per Share: -0.11 TL/share
- AFYON:
  - Total Earnings: -723.22 TL
  - Total Shares Traded: 57,223
  - Earnings per Share: -0.01 TL/share

**Analysis**

- The negative earnings from both AEFES (-1,211.78 TL) and AFYON (-723.22 TL) significantly impacted overall profitability.
- AEFES had the highest negative earnings per share (-0.11 TL/share), indicating poor performance and unprofitable trades.
- The positions held in AFYON at the end of the simulation suggest unresolved trades that did not meet exit criteria.
- The substantial decrease in the Sharpe Ratio and increase in maximum drawdown highlight the increased risk and volatility due to the inclusion of underperforming stocks.

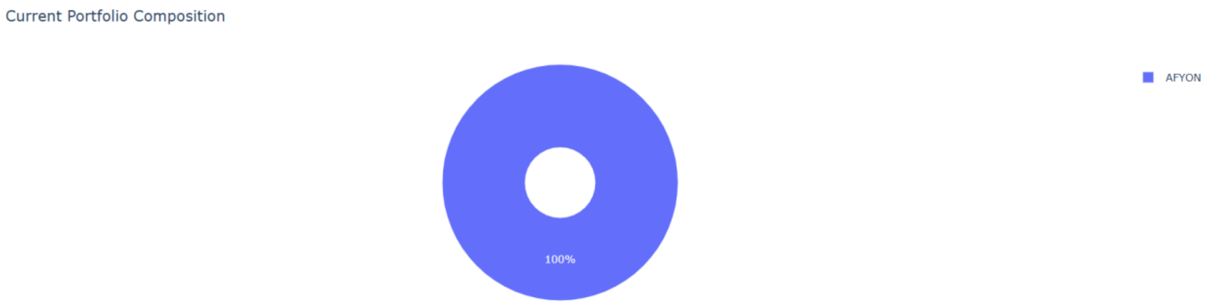


**Figure 16** Position hold for AFYON

# 5. CONCLUSIONS

The project aimed to use LSTM networks for predicting stock prices and informing trading strategies in an autonomous stock trading application. Focusing on selected stocks, ASELS, THYAO, AFYON, and AEFES from the Borsa Istanbul, Turkish stock exchange, this application tries to find ways to integrate machine learning models into algorithmic trading in an emerging market context. The methodology will involve extensive preprocessing of data, and feature engineering for creating a robust dataset enriched with technical indicators and lag features. Developing separate LSTM models on each stock for capturing stock-specific patterns and temporal dependencies; training the model on historical data of over six years with careful hyperparameter tuning to avoid overfitting. The trading strategy implemented dynamic thresholds based on ATR, together with some user-configurable parameters for risk management, such as stop-loss and take-profit in percent. Backtesting simulation was performed for different scenarios of stock selections to gauge the performance of the strategy.

## 5.1. Recommendations for Future Work

**Different Data Sources Based on Sentiment**: The future model could also incorporate alternative data sources, like news sentiment analysis and social media trends to further give a full view of factors influencing stock prices.

**Real-Time Deployment**: Progressing from backtesting simulations to live real-world trading environments would provide realistic insight and also expose the strategy to real-life market dynamics, hence making iterative improvements.

## 5.2. Final Remarks

This project contributed to the field of algorithmic trading by demonstrating the integration of LSTM-based predictive models within an autonomous trading application in an emerging market context. The positive results achieved with selected stocks indicate the potential of machine learning techniques to inform and enhance trading strategies. However, the limitations identified highlight the necessity for cautious application, continuous model refinement, and comprehensive risk management.

By addressing these aspects in future work, the application could evolve into a more robust and versatile tool, capable of adapting to a wider array of market conditions and providing valuable insights for traders and investors in the financial industry.

# REFERENCES


Adebiyi, A. A., Ayo, C. K., Adebiyi, M. O., & Otokiti, S. O. (2020). Stock price prediction using LSTM, CNN and convolutional-LSTM models. *International Journal of Engineering and Technology*, 9(2), 29–37.

Borsa İstanbul. (2023, October 12). *In Wikipedia*. Retrieved January 12, 2025, from https://tr.wikipedia.org/wiki/Borsa_%C4%B0stanbul

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654–669.

Kesler, G. (2020). *Borsa Istanbul: Turkish Stock Exchange Dataset*. Kaggle. https://www.kaggle.com/datasets/gokhankesler/borsa-istanbul-turkish-stock-exchange-dataset

Ortega, P. A., & Kansa, E. J. (2015). How good is your algorithmic trading strategy? *Journal of Portfolio Management*, 42(1), 116–127.