



TÜRKİYE CUMHURİYETİ
ADANA ALPARSLAN TÜRKER SCIENCE AND TECHNOLOGY
UNIVERSITY

GRADUATE SCHOOL OF ENGINEERING
DEPARTMENT OF INDUSTRIAL ENGINEERING

FORECASTING STOCK PRICES USING DEEP LEARNING MODELS
WITH TECHNICAL INDICATORS

Mine KONUR

MASTER OF SCIENCE

ADANA 2024



TÜRKİYE CUMHURİYETİ
ADANA ALPARSLAN TÜRKEŞ SCIENCE AND TECHNOLOGY
UNIVERSITY

GRADUATE SCHOOL OF ENGINEERING
DEPARTMENT OF INDUSTRIAL ENGINEERING

FORECASTING STOCK PRICES USING DEEP LEARNING MODELS
WITH TECHNICAL INDICATORS

MİNE KONUR
MASTER OF SCIENCE

THESIS ADVISOR
PROF. DR. Mustafa GÖÇKEN

ADANA, 2024

ÖZET

TEKNİK GÖSTERGELERLE DERİN ÖĞRENME MODELLERİ KULLANARAK HİSSE SENEDİ FİYATI TAHMİNİ

Mine KONUR

Endüstri Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Mustafa GÖÇKEN

Ağustos 2024, 107 sayfa

Hisse senedi fiyatlarının doğru tahmini, yatırımcıların ne zaman ve nereye yatırım yapacaklarına dair kararlarını destekler. Ancak, hisse senedi piyasasının dinamik, doğrusal olmayan, karmaşık ve kaotik doğası, fiyat tahminini zorlaştırır. Piyasa hareketleri, siyasi olaylar, firma politikaları, ekonomik koşullar, emtia fiyatları ve banka kurları gibi birçok makroekonomik faktörden etkilenir. Ayrıca gelişen teknoloji ve iletişim sistemleri, bu olayların hızla işlenmesine olanak tanıyarak hisse senedi fiyatlarının hızlı dalgalanmasına neden olur. Dolayısıyla bankalar, finans kurumları ve büyük yatırımcılar, hızlı alım satım yapmak zorunda kalır ve piyasanın karmaşıklığı kesin tahmin yapmayı daha da zorlaştırır. Hisse senedi fiyatlarının doğru tahmini için yeni ve etkili yöntemlerin geliştirilmesi gerekmektedir. Bu çalışmada, Yahoo Finance gibi internet sitelerinden elde edilen veriler teknik göstergeler kullanılarak hisse senedi fiyatlarının tahmin edilmesi amaçlanmıştır. Ham verilerdeki gürültüyü azaltmak, anlamlı sonuçlar elde etmek ve tahmin doğruluğunu artırmak amacıyla teknik indikatörler için korelasyon katsayısı yöntemi kullanılmıştır. Yapay Sinir Ağları (YSA) modelleri, finansal zaman serisi verilerinin analizinde büyük bir etkinlik sağlamaktadır. YSA ile yapılacak tahminler için en uygun parametreler seçilmiştir. Veri ön işleme sürecinden sonra, Tek Katmanlı Uzun Kısa Süreli Hafıza (LSTM), 3 Katmanlı LSTM, 3 Katmanlı Çift Yönlü Uzun Kısa Süreli Hafıza (BiLSTM) ve hibrit Evrişimli Sinir Ağı- Uzun Kısa Süreli Hafıza (CNN-LSTM); yöntemleri uygulanmıştır. Bu model ile, gerçeğe yakın fiyat tahminlerinin bulunmasının yanı sıra, teknik indikatörler aracılığıyla hisse senedi fiyatı tahminini etkileyen özellikleri azaltmak amaçlanmıştır. Sonuçlar, derin öğrenme (DL) teknikleri arasında Tek Katmanlı LSTM yönteminin diğer yöntemlere göre daha gerçekçi tahminler sunduğunu ortaya koymuştur.

Anahtar Kelimeler: Hisse senedi fiyatı tahmini, yapay sinir ağları, teknik indikatörler, Hisse senedi tahmini.

ABSTRACT

FORECASTING STOCK PRICES USING DEEP LEARNING MODELS WITH TECHNICAL INDICATORS

Mine KONUR

Department of Industrial Engineering

Supervisor: Prof. Dr. Mustafa GÖÇKEN

August 2024, 107 pages

Accurate stock price prediction is crucial for supporting investors' decisions regarding the timing and allocation of investments. However, the dynamic, non-linear, complex, and chaotic nature of the stock market makes price prediction a challenging task. Market movements are influenced by various macroeconomic factors, such as political events, corporate policies, economic conditions, commodity prices, and bank rates. Furthermore, advancements in technology and communication systems enable these events to be processed rapidly, leading to swift fluctuations in stock prices. Consequently, banks, financial institutions, and large investors are compelled to make quick buy and sell decisions, further complicating precise predictions. Therefore, there is a pressing need for the development of new and effective methods for accurate stock price predictions. This study aims to predict stock prices using technical indicators obtained from sources like Yahoo Finance. To reduce the noise in raw data, obtain meaningful results and increase the estimation accuracy, correlation coefficient method was used for technical indicators. Artificial Neural Networks (ANN) models provide great efficiency in the analysis of financial time series data. The most appropriate parameters were selected for the estimations to be made with ANN. After the data preprocessing process, Single Layer Long Short Term Memory (LSTM), 3 Layer LSTM, 3 Layer Bidirectional Long Short Term Memory (BiLSTM) and Hybrid Convolutional Neural Network-Long Short Term Memory (CNN-LSTM); methods were applied. With this model, in addition to finding realistic price estimations, it was aimed to reduce the features affecting stock price estimation through technical indicators. The results showed that the Single Layer LSTM method provides more realistic estimations compared to other methods among deep learning (DL) techniques.

Keywords: Artificial neural networks, Stock price prediction, Technical Indicators, Stock price prediction.

Dedication

I dedicate this thesis to my beloved family who has supported me throughout my life and has been with me every step of the way. I would like to thank my mother Emine KONUR and my father Ali KONUR for believing in me, encouraging me and never giving up their love. I would like to thank my sister Hüsne KONUR for always being there for me and giving me strength. I would also like to thank my comrade Kaan BİLGİN for always being there for me and supporting me.

This study was made possible with all your support. Thank you very much.

ACKNOWLEDGEMENTS

First of all, I would like to express my deep gratitude to my advisor Prof. Dr. Mustafa GÖÇKEN. He has guided me throughout my research, given valuable suggestions and ensured the completion of this study with his encouraging approach. I would also like to express my deep gratitude to my respected professor Assoc. Prof. Dr. Ayşe Tuğba DOSDOĞRU, who guided me through this challenging journey and generously shared her knowledge with me. This success would not have been possible without her support. I would like to express my infinite gratitude to Assoc. Prof. Dr. Muhammet DEVECİ, from whose knowledge and experience I benefited at every stage of this study.

Finally, I dedicate this study to all my teachers and academicians who have inspired me throughout my educational life. Without your efforts and hard work, I would not have achieved this success today.

My infinite gratitude to all of you.

TABLE OF CONTENTS

CERTIFICATION OF APPROVAL	I
DECLARATION OF CONFORMITY	II
ÖZET	III
ABSTRACT	IV
<i>Dedication</i>	V
ACKNOWLEDGEMENTS	VI
TABLE OF CONTENTS	VII
LIST OF FIGURES	IX
LIST OF TABLES	XI
LIST OF ABBREVIATIONS	XII
1. INTRODUCTION	1
2. LITERATURE REVIEW	7
3. MATERIAL AND METHOD	21
3.1. Machine Learning	21
3.2. Deep Learning (DL)	22
3.3. Artificial Neural Networks	23
3.3.1. Introduction and History of ANN	23
3.3.2. Biological Infrastructure and Architecture	25
3.3.3. ANN Structure	28
3.3.4. Training of ANN	30
3.3.5. Types of Learning	32
3.3.5.1. Supervised Learning	32
3.3.5.2. Semi-Supervised Learning	33
3.3.5.3. Unsupervised Learning	33
3.3.6. Gradient Descent and the Delta Rule	34
3.3.6.1. Derivation of The Gradient Descent Rule	36
3.3.7. Activation Function Types	38
3.4. Convolution Neural Network (CNN)	43
3.4.1. Convolution Layer	43
3.4.2. Pooling layer	44
3.4.3. Fully connected layer	45

3.5.	Recurrent Neural Networks (RNN)	46
3.6.	Long Short Term Memory Networks-LSTM	47
3.7.	Bidirectional LSTM	51
3.7.1.	Bidirectional LSTM Structure	51
3.7.2.	Bidirectional LSTM Principle	52
3.8.	Hybrid Model CNN-LSTM	53
3.9.	10 Commonly Used Indicators in Technical Analysis	54
3.9.1.	Cumulative Log Returns	55
3.9.3.	Relative Strength Index (RSI)	56
3.9.4.	Moving Average Convergence Divergence (MACD)	57
3.9.5.	Bollinger Band (BB)	58
3.9.6.	Stochastic Indicator	60
3.9.7.	Moving Averages	61
3.9.7.1.	Simple Moving Average	62
3.9.7.2.	Weighted Moving Average	62
3.9.7.3.	Exponential Moving Average	63
3.9.8.	KDJ indicator	63
3.9.9.	Fibonacci Retracements	65
3.9.10.	Ichimoku	66
3.10.	Data Preprocessing	69
3.10.1.	Min-Max Scaler	71
3.10.2.	Data Information	71
3.10.3.	Preparing Data for the Model	72
3.11.	Performance Evaluation	74
3.12.	Programming Languages and Libraries Used in Deep Learning	77
4.	RESULTS AND DISCUSSION	78
4.1.	Computational Results	91
5.	CONCLUSIONS	105
	REFERENCES	107
	CURRICULUM VITAE	119

LIST OF FIGURES

Figure 3.1. The single-layer network structure (Melli, 2017)	24
Figure 3.2. The neural network (Rakesh, 2018)	26
Figure 3.3. Adjust weights and biases (Tekyaygil, 2020)	27
Figure 3.4. The number of hidden layers and the number of neurons (Vui et al., 2013)	28
Figure 3.5. Functionality of neuronal models (Kılıç, 2023)	29
Figure 3.6. A function of the space of possible weight vectors (Mitchell, 1997)	35
Figure 3.7. Feature map (Pratiwi, 2020)	44
Figure 3.8. Maximum Pooling (Haq, 2023)	45
Figure 3.9. RNN Architecture Explained (Feng, 2017)	47
Figure 3.10. The LSTM architecture (Nyugen et al., 2019)	49
Figure 3.11. Bi-LSTM Structure (Park, 2020)	52
Figure 3.12. The CNN-LSTM model (Lu, 2020)	53
Figure 3.13. The KDJ indicator explained (Muhammad, 2022)	64
Figure 3.14. The Ichimoku Kinko Hyo system (Fidelity Investments. (n.d))	69
Figure 3.15. Data Flow Chart (Briceno-Mena, 2022)	70
Figure 4.1. Close price for selected firms	81
Figure 4.2. Feature Selection Criteria for KCHOL.IS	84
Figure 4.3. Histogram for KCHOL.IS	84
Figure 4.4. Feature Selection Criteria for THYAO.IS	85
Figure 4.5. Histogram for THYAO.IS	86
Figure 4.6. Feature Selection Criteria for FROTO.IS	87
Figure 4.7. Histogram for FROTO.IS	87
Figure 4.8. Feature Selection Criteria for TUPRS.IS	88
Figure 4.9. Histogram for TUPRS.IS	89
Figure 4.10. Feature Selection Criteria for ISCTR.IS	90
Figure 4.11. Histogram for ISCTR.IS	90
Figure 4.12. Plot for Real value vs Predicted value for KCHOL.IS using BiLSTM	94
Figure 4.13. Plot for Real value vs Predicted value for THYAO.IS using BiLSTM	95
Figure 4.14. Plot for Real value vs Predicted value for FROTO.IS using BiLSTM	95
Figure 4.15. Plot for Real value vs Predicted value for TUPRS.IS using BiLSTM	95
Figure 4.16. Plot for Real value vs Predicted value for ISCTR.IS using BiLSTM	96

Figure 4.17. Plot for Real value vs Predicted value for KCHOL.IS using CNN-LSTM.....	96
Figure 4.18. Plot for Real value vs Predicted value for THYAO.IS using CNN-LSTM.....	97
Figure 4.19. Plot for Real value vs Predicted value for FROTO.IS using CNN-LSTM.....	97
Figure 4.20. Plot for Real value vs Predicted value for TUPRS.IS using CNN-LSTM	97
Figure 4.21. Plot for Real value vs Predicted value for ISCTR.IS using CNN-LSTM	98
Figure 4.22. Plot for Real value vs Predicted value for KCHOL.IS using Three-Layer-LSTM	98
Figure 4.23. Plot for Real value vs Predicted value for THYAO.IS using Three-Layer-LSTM	99
Figure 4.24. Plot for Real value vs Predicted value for FROTO.IS using Three-Layer-LSTM	99
Figure 4.25. Plot for Real value vs Predicted value for TUPRS.IS using Three-Layer-LSTM	99
Figure 4.26. Plot for Real value vs Predicted value for ISCTR.IS using Three-Layer-LSTM	100
Figure 4.27. Plot for Real value vs Predicted value for KCHOL.IS using SingleLayer-LSTM	100
Figure 4.28. Plot for Real value vs Predicted value for THYAO.IS using SingleLayer-LSTM	101
Figure 4.29. Plot for Real value vs Predicted value for FROTO.IS using SingleLayer-LSTM	101
Figure 4.30. Plot for Real value vs Predicted value for TUPRS.IS using SingleLayer-LSTM	101
Figure 4.31. Plot for Real value vs Predicted value for ISCTR.IS using SingleLayer-LSTM	102

LIST OF TABLES

Table 3.1. Gradient descent pseudocode	38
Table 3.2. Activation functions	39
Table 3.3. Backpropagation algorithm pseudocode	42
Table 4.1. The market capitalization values of stock prices.....	79
Table 4.2. The stock is related to data information	80
Table 4.3. Technical Indicators	82
Table 4.4. First hidden Layer LSTM.....	93
Table 4.5. The three-layer long short-term memory	93
Table 4.6. The three-layer BiLSTM	94
Table 4.7. Comparison of the errors of the prediction results of the traditional stock market indices by various models.	103

LIST OF ABBREVIATIONS

ANN	: Artificial Neural Network
LSTM	: Long Short-Term Memory
BiLSTM	: Bidirectional Long Short-Term Memory
CNN	: Convolutional Neural Network
RNN	: Recurrent Neural Network
DL	: Deep Learning
ARIMA	: AutoRegressive Integrated Moving Average
MLP	: Multi-Layer Perceptron
SVM	: Support Vector Machine
NSE	: National Stock Exchange of India
RF	: Random Forest
SVR	: Support Vector Regression
MSCI UK	: Morgan Stanley Capital International United Kingdom
GDP	: Gross Domestic Product
ML	: Machine Learning
SMA	: Simple Moving Average
HS	: Harman Search
GA	: Genetic Algorithm
RMSE	: Root Mean Square Error
MAPE	: Mean Absolute Percentage Error
AM	: Attention Mechanism
PCA	: Principal Component Analysis
MAE	: Mean Absolute Error
TA	: Turkish Airlines
BİST	: Borsa İstanbul
US	: United States
NN	: Neural Network
S&P 500	: Standard & Poor's 500
GBM	: Geometric Brownian Motion
KSE	: Karachi Stock Exchange
FX	: Foreign Exchange

SLP	: Single Layer Perceptron
RBF	: Radial Basis Function
DTRSI	: Deep Transfer with Relevant Stock Information
DNN	: Deep Neural Network
TEML	: Tech Mahindra
EURJPY	: Euro / Japanese Yen
BSI	: Boursa Kuwait Stock Index
GRNN	: General Regression Neural Network
RACAP	: Generalized Regression Neural Network
FL	: Fuzzy Logic
ARO	: Artificial Rabbit Optimization
DJIA	: Dow Jones Industrial Average
MSE	: Mean Squared Error
R2	: Coefficient Of Determination
DCT	: Discrete Cosine Transform
MFRFNN	: Multi-Functional Recurrent Fuzzy Neural Network
VDM	: Variable Mode Decomposition
IMFs	: Intrinsic Mode Functions
HSI	: Hang Seng Index
SSE	: Shanghai Stock Exchange
GRU	: Gated Recurrent Unit
LWDTW	: Logistic Weighted Dynamic Time Deviation
WDTW	: Weighted Dynamic Time Deviation
NYSE	: New York Stock Exchange
MLP	: Multilayer Perceptrons
BiCuDNNLSTM	: Bidirectional Cuda Deep Neural Network Long Short-Term Memory
DSE	: Dhaka Stock Exchange
VAR	: Vector Autoregression
GFNN	: Gaussian Feedforward Neural Network
TCN	: Temporal Convolutional Network
Evar	: Explained Variance
Acc	: Accuracy
FIVMD	: Feedback Interval Variational Modal Decomposition

SZI	: Shenzhen Stock Exchange Index
CSI100	: China Securities 100 Index
COI	: Company of Interest
CI	: Index on Day
CI+CS	: Cosine Similarity
CI+SF	: Similar Field
CI+HMC	: Highest Market Capitalization
CI+RND	: Randomly on Day
KNN	: K-nearest neighbors
TCS	: Tata Consultancy Services
NIFTY 50	: National Stock Exchange Fifty
Adam	: Adaptive Moment Estimation
IBM	: International Business Machines Corporation
DA	: Directional Accuracy
APE	: Absolute Percentage Error
AEE	: Average Absolute Error
ARPE	: Average Relative Percentage Error
TSRM	: Time Series Relational Model
GCN	: Graph Convolutional Network
MA	: Moving Averages
RSI	: Relative Strength Index
MACD	: Moving Average Convergence Divergence
IoT	: Internet of Things
AI	: Artificial Intelligence
MCP	: Multilayer Sensors
WMA	: Weighted Moving Average
EMA	: Exponential Moving Average
BB	: Bollinger Bands
BBL	: Bollinger Bands Lower
BBM	: Bollinger Bands Middle
BBU	: Bollinger Bands Upper
BBB	: Bollinger Bands Width
BBP	: Bollinger Bands Percent

ATR	: Average True Range
ADX	: Average Directional Index
ISA	: Tenkan-Sen
ISB	: Kijun-Sen
ITS A	: Senkou Span
IKS B	: Senkou Span
ICS	: Chikou Span
AME	: Absolute Maximum Error
MSLE	: Mean Squared Logarithmic Error
MDA	: Mean Directional Accuracy
EVS	: Explained Variance Score
KCHOL.IS	: Koç Holding
THYAO.IS	: Turkish Airlines
FROTO.IS	: Ford Otosan
TUPRS.IS	: Türkiye Petrol Rafinerileri
ISCTR.IS	: İş Bankası

1. INTRODUCTION

A stock exchange is a market where securities of the same type are traded in a specific order. In other words, the stock market is the venue for the daily purchase and sale of billions of dollars' worth of assets. The stock market plays a main role in the allocation of capital, the creation of wealth, and the advancement of the economy. Stock exchanges serve as venues for the trading of not only securities but also commodities. For example, there are cotton exchanges, precious metal exchanges, and commodity exchanges (Lu, 2020). The stock market plays a pivotal role in economic activity, fulfilling the capital requirements of businesses. Investors engage in both buying and selling activities to generate profits in the stock market. A trading strategy is an active strategy that is implemented by a set of defined rules (Lu, 2020).

The stock market, frequently referred to as the stock market or securities market, constitutes a pivotal component of the universal financial system. It functions as a marketplace where buyers and sellers converge to transact financial instruments, particularly stocks, which represent ownership in public companies. All financial investments are made with to make a profit. In the field of financial investments, investors can't foresee potential profit or loss scenarios in advance. As a result, they formulate their predictions based on a combination of past data, current market information, and their intuitive insights (Şakrak, 2022). Nevertheless, the inherent non-stationarity and non-linearity of financial series render accurate prediction a challenging problem. The stock price is influenced by a range of factors, including changes in national policies, the internal and external economic environments, and international situations. As explained by Lu (2021), this phenomenon is affected by numerous factors. Investors must monitor market developments, new information, and announcements, as these can directly impact their financial decisions (Şakrak, 2022).

The methods used in stock price prediction have been extensively documented in the literature. Research has been conducted on time series analysis and linear regression models, which are traditional statistical methods that have been used for many years. As technology advances, the methods used to work with time series data are also evolving. In recent years, ANN, recurrent neural networks (RNN), and LSTM networks have emerged as preferred approaches. In a study conducted by Adebiyi et al. (2014), the researchers attempted to make predictions using autoregressive integrated moving average (ARIMA) and ANN for New York stock data. The

findings indicated that ANN demonstrated superior performance compared to ARIMA. Anand (2021) employed CNN, LSTM, RNN, multilayer perceptron (MLP), and support vector machines (SVM) as DL architectures for stock price prediction of the world's leading stock exchange, the National Stock Exchange of India (NSE). Considering sudden changes in the system, the CNN model exhibited superior predictive accuracy compared to other techniques. In their study, Nikou et al. (2019) employed a range of machine learning (ML) techniques, including LSTM, random forest (RF), Support Vector Regression (SVR), and ANN, to predict the daily closing price of the iShares MSCI United Kingdom stock exchange. The LSTM method yielded superior prediction results compared to other techniques for forecasting the daily closing price of the iShares MSCI United Kingdom. In these studies, historical data values, technical indicators, and basic economic data are typically employed.

The stock market is frequently characterized as a dynamic, unpredictable, and non-linear system. The prediction of stock prices is subject to several influencing factors, including political conditions, the global economic environment, and the financial reports and performance of a particular company or group of companies. As a result of the multitude of variables that can potentially influence stock market movements, forecasting stock prices remains a challenging undertaking (Vijh, 2020). It is therefore crucial for investors to gain an understanding of these factors to make informed stock predictions. Macroeconomic indicators that affect stock forecasts provide important information about market conditions, economic health and the general investment environment. These basic macroeconomic indicators can be listed as follows: gross domestic product (GDP), unemployment rate, inflation rate, interest rates, monetary policy, budget deficit and public debt, consumer confidence index, industrial production, foreign trade balance, and exchange rate changes. The GDP represents the total value of all goods and services produced within a specified period, providing a measure of the economic size and health of a country. The growth of GDP is an indicator of economic growth, which in turn can lead to increased company profits and a positive effect on stock prices. A rise in unemployment levels may result in a reduction in consumer spending. Consequently, this may have a damaging effect on company revenues. A reduction in the unemployment rate can lead to an improvement in a company's performance, which in turn can result in an increase in consumer spending. When the inflation rate is increased, the cost of goods and services rises, which in turn reduces profit margins. At the same time, the possibility of a rise in interest rates by the central bank arises. Therefore, it may have a detrimental impact on stock prices. An

increase in interest rates can result in higher borrowing costs for companies, which may subsequently lead to a reduction in consumer spending. Low interest rates can stimulate investment and spending by encouraging borrowing. A reduction in interest rates and an increase in the money supply can stimulate economic growth and lead to an increase in stock prices. Conversely, high interest rates and a low money supply exert a constraining influence on the economy, which may result in a decline in stock prices. A budget deficit and total public debt are the inevitable consequences of government expenditures exceeding revenues. The consequence of high budget deficits and public debt is that future tax increases or spending cuts are inevitable. Therefore, this could result in a deceleration of economic growth and a detrimental impact on stock prices. A rise in consumer confidence may result in an increase in spending and economic growth. Consequently, this may increase stock prices. Conversely, low confidence can result in a reduction in spending and a lowering of prices. Industrial production represents the total quantity of goods and services produced in a country's factories, mines, and service sectors over a specified period. Consequently, an increase in industrial production is indicative of economic growth and corporate profitability, and can have a beneficial impact on stock prices. The foreign trade balance represents the discrepancy between a country's exports and imports. A trade deficit indicates that a country's imports exceed its exports, resulting in a net outflow of foreign exchange. Such developments may have an impact on the exchange rate and inflation. Exchange rate fluctuations indicate the relative value of a country's currency in comparison to other currencies. An appreciation of the currency may lead to a decrease in the cost of imported goods, while simultaneously making the export of goods more challenging. While a depreciation of the currency encourages exports, it increases the cost of imported goods. These macroeconomic indicators have the potential to exert a direct or indirect influence on stock prices. It is crucial for investors to anticipate future market trends. It enables investors and analysts to gain insight into the economic situation and market expectations.

The objective of this study is to employ two analytical approaches, fundamental analysis and technical analysis, to reduce the inherent noise in raw data and thereby facilitate the prediction of stock prices. Consequently, the elimination of superfluous data will facilitate the attainment of precise predictive outcomes. Fundamental analysis and technical analysis represent two significant methodologies employed in stock market forecasting, with each approach focusing on distinct aspects of security evaluation.

Fundamental analysis is the process of determining the intrinsic value of a security by examining a company's or asset's economic, financial, and qualitative factors. The objective is to ascertain whether a security is overvalued or undervalued in the market. The objective of fundamental analysis, which focuses on current values, is to provide individuals and institutions with a long-term perspective on the company or stock in question. To estimate a company's stock price or return, several basic financial indicators must be taken into account. These include income-expense ratios, annual growth rates, market position and sector analysis, financial statements, price-earnings ratios (P/E), and market value/book value (P/B). The aforementioned fundamentals are of particular consequence for stock prices, as investors endeavor to anticipate future performance based on this data.

The fundamental analysis of a given company is likely to exert an influence on its functioning and, consequently, on the functioning of the company in question's inclusion in the stock index. In other words, it guides investment decisions. Consequently, it is an investment analysis in which the share value of a company is estimated by means of an analysis of its sales, earnings, and profits. This method is most appropriate for long-term forecasting (Çetinyokuş, 2002).

Economic factors an evaluation of economic factors in fundamental analysis is crucial for comprehending the prevailing economic climate and its ramifications for a specific company or industry. Notable economic indicators encompass interest rates, inflation rates, and unemployment rates. An industry analysis is a method of examining the overall performance of a company within its respective industry. This analysis is employed to assess industry trends, competitive landscapes, market sizes, and regulatory factors. Regarding management and corporate strategy, how a company is managed and the strategy it employs can have a profound impact on its future performance. Investors assess a company's management team based on a range of criteria, including their experience, leadership style, strategic vision, and past performance. Regarding sectoral and company analyses, sectoral reports offer general information regarding a specific industry or sector. Such an analysis can be employed to facilitate a comparison of the performance of companies within this industry. A company analysis is conducted to evaluate the financial soundness, performance, and future potential of a specific company. The P/E, P/B ratio, and earnings per share are methods of company stock analysis (Birgili, 2013).

Technical analysis is a method of financial analysis that employs the examination of past price movements and transaction volumes of securities to predict future price trends (Birgili, 2013). It is widely acknowledged that Charles Dow is the individual who introduced the concept of technical analysis to the field of finance. Consequently, he is regarded as the founder of modern technical analysis. In 1884, Dow developed the first market index. The fundamental premise of Dow theory is that the returns on stocks over time are contingent upon the prevailing trends in stock prices. There is no consensus on the definition of technical analysis. Technical analysis is the process of identifying information that can be utilized in the formulation of trading and investment decisions, employing historical price and transaction volume data (Uyar, 2020). Technical analysis is founded upon the tenets of basic economic theory. The fundamental tenets of technical analysis are as follows:

- Stock prices are determined by supply and demand.
- Stock prices move with trends.
- Changes in supply and demand cause trends to reverse.
- Changes in supply and demand can be detected with the help of graphs.
- Patterns on charts tend to repeat themselves.

Technical analysis is the process of examining historical price movements of stocks to gain insight into future price trends. The resulting graphs offer a basis for forecasting future stock prices. For this reason, graphs are employed in the field of technical analysis.

The objective is to obtain predictive values through the application of diverse ML algorithms and DL models, utilizing data that becomes meaningful through the use of technical indicators. The objective is to utilize the most sought-after DL models for prediction, namely, ANN, CNN, LSTM, and BiLSTM methods. The findings of the studies indicate that the predictions yield results that are highly correlated with the actual values. Furthermore, the evaluation of the models will be based on the assessment of their accuracy, error rates, and other performance metrics. The data set to be utilized in the study will be procured from a dependable source, such as Yahoo Finance. The data set will comprise the daily stock closing price financial indicator for a specified period. The period selected for estimation was determined to encompass the previous two years. The data will be partitioned in a manner conducive to its utilisation in the training and testing phases of the models.

This study contributes to the existing literature on stock price prediction by comparing the performances of different ML models and determining the most effective methods. Additionally, the study aims to assist investors and financial analysts by offering more precise and dependable projections for financial markets.

The initial section of this thesis presents an overview of the subject matter. In the second part of the study, a review of the literature was conducted on studies on stock price forecasting. In the third section, more information about ML, DL, ANN, CNN, LSTM, BiLSTM, and performance criteria is given. The techniques used will also be explained. In the fourth section, the results obtained using ANN, LSTM, CNN-LSTM, and BiLSTM models were compared and analyzed, and the accuracy of the models was determined by evaluating the performance metrics. In the fifth section, the key findings are analyzed and discussed.

2. LITERATURE REVIEW

Stock price prediction is of critical importance in the context of financial markets. A substantial body of academic research has been conducted over time on the prediction of stock prices. The objective of the literature on stock price prediction is to predict future price movements using various methods and approaches. The efficacy of traditional methods, such as technical and fundamental analysis, has been evaluated in the context of stock price prediction. In recent years, the development of Artificial Intelligence (AI) and ML techniques has given rise to novel approaches in stock prediction. These methods have demonstrated considerable success in the domain of stock prediction. However, it should be noted that each method has its own set of advantages and disadvantages. Fundamental analysis is better suited to long-term investment decisions, whereas technical analysis is more effective for short-term trading decisions. The application of ML and DL techniques has the potential to facilitate more accurate predictions by processing large datasets.

Göçken et al. (2015) aimed to determine the effectiveness of using Simple Moving Average (SMA) and momentum closing price technical indicators to forecast the closing prices of stocks in the Turkish stock market. The study used a hybrid ANN model that combines harmony search (HS) and genetic algorithms (GA) to capture the relationship between technical indicators and stock market performance. The proposed models were evaluated using various criteria, including the loss function, return on investment analysis, buy-and-hold analysis, and graphical analysis. GA was used to train the network, while harmony search algorithms were employed to enhance diversity. For the GA-ANN model, 26 input values were selected, while 23 input values were chosen for the HS-ANN model. The GA used two neurons, while the hidden layer of the harmony search consisted of 17 neurons. Among the trained models, the HS showed superior performance on the test set, although it produced higher error rates compared to the GA. The findings showed that the HS-based ANN model exhibited superior overall performance.

Vijh et al. (2020) used both ANN and RF methodologies to predict the daily closing prices of five companies from various sectors. The companies studied were Nike, Goldman Sachs, Johnson & Johnson, Pfizer, and JP Morgan Chase and Co. The models were evaluated using Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) metrics. The

findings showed that ANN demonstrated superior performance in predicting stock prices compared to RF.

Lu et al. (2021) proposed a method based on CNN and BiLSTM units with an attention mechanism (AM) for forecasting the closing price of the Shanghai Composite Index on the following day over a period exceeding 1,000 trading days. In this approach, a CNN is employed to extract pertinent features from the input data, which are then utilized by a BiLSTM network to predict the stock's subsequent closing price. The AM is used to capture the impact of feature states at different past times on the stock closing price, thereby enhancing prediction accuracy. To demonstrate the efficacy of this method, it was compared with seven other methods for predicting stock closing prices. Compared to the other methods, the CNN-BiLSTM-AM approach offers a dependable means for investors to anticipate stock prices and make informed stock investment decisions.

Selvin et al. (2017) employed three distinct DL architectures—namely, RNN, LSTM, and CNN—to forecast the prices of Tata Consultancy Services (TCS) and CIPLA companies traded on the NSE. Using these DL architectures, they identified hidden dynamics in the data. A sliding window approach was employed to evaluate the predictive performance of these models in the short term. The efficacy of the models was assessed through the use of ARIMA error percentages. The results of the stock price prediction demonstrated that the CNN network exhibited a high degree of accuracy in capturing the trends and providing precise forecasts, outperforming the other two models. It was determined that the CNN model is the optimal choice for the proposed methodology.

Zahedi et al. (2015) evaluated the predictability of prices on the Tehran Stock Exchange by applying ANN models and the principal component analysis (PCA) method using 20 accounting variables. The goodness of fit for PCA was determined based on actual values, and the factors influencing Tehran Stock Exchange prices were accurately predicted. The results of the statistical analysis demonstrated that the ANN method exhibited superior performance compared to both the PCA and MLP methods.

Laboissiere et al. (2015) put forth a methodology for forecasting the maximum and minimum daily stock prices of three Brazilian energy distribution companies listed on the São Paulo Stock

Exchange BM&FBovespa. In comparison to existing literature, this paper makes a notable contribution to the field by predicting the closing price range of stocks of Brazilian electric distribution companies. The proposed methodology is based on the calculation of different features to be analyzed through feature selection, to identify the most suitable features for predicting the maximum and minimum daily stock prices for each company. Subsequently, the actual prediction was achieved through the use of ANNs, with performance evaluated through the calculation of mean absolute error (MAE), MAPE, and RMSE.

In the Istanbul Stock Exchange (BIST), the stock prices of seven companies comprising the insurance sector index—Aksigorta, Anadolu Hayat Emeklilik, Anadolu Sigorta, Aviva Sigorta, Güneş Sigorta, Ray Sigorta, and Yapı Kredi Sigorta—have been predicted using the ANN model, which is commonly utilized in contemporary financial markets (Akcan & Kartal, 2011). The objective of the study was to forecast the stock prices of these seven insurance companies over specified periods. These periods included 15 days (11 business days), one month (23 business days), one and a half months (34 business days), and two months (44 business days). The forecasts were based on the values of the companies from March 1 to April 30. The results demonstrate that all predictions were accurate, particularly those for periods of up to one month.

Çınaroğlu and Avcı (2020) predicted the stock values of Turkish Airlines (TA) traded on BIST. Using daily data from 2015 to 2018, the most successful ANN model was determined by comparing the stock values predicted by different ANN models with the actual values in the same period. In the analysis, the stock value was used as output and the BIST 100 and BIST Transportation indices, oil and dollar prices were used as inputs. In the application phase, the values of the first 4 days of the week were selected as training data and the values of Friday as test data. In the remainder of the study, the most successful model was used to forecast stock prices for the first 10 trading days of January 2019. In the obtained results, it was concluded that the use of ANN for predicting the behavior and trends of stocks is meaningful. It is concluded that the actual stock values of TA are very close to the predicted stock values.

Özçalıcı (2016) predicted the closing prices of stocks after 1 day, 2 days and 20 days. Price and volume data from 2010 and 2015 were used. The forecasting method used in the study was ANN. As a result, it was concluded that the price movements of stocks were predicted accurately up to 72.88% 20 days in advance.

Yakut et al. (2014) used feed-forward neural networks and SVM methods. To predict the BIST index, one, two, and three days prior values of the BIST index, as well as the US dollar exchange rate, the overnight interest rate and the index values of the Japanese (NIKKEI), Brazilian (BOVESPA), British (FTSE), French and German (DAX) stock exchanges were also considered. Using the stock market index values from 2005 and 2012, the BIST index value was estimated using feed-forward ANN and SVM methods. It is concluded that ANN and SVM methods can be used to predict the stock market index.

In Tektas and Karatas (2004), the stock prices of seven companies were predicted using ANN. The ANN method was applied to weekly and daily data, with daily data yielding better results. Then ANN and linear regression methods were applied using daily data. ANN provided better results than regression.

MLP, CNN, and LSTM neural network techniques were used to predict the upward and downward trend movements of the stock market index (Persio and Honchar, 2016). They aimed to highlight the importance of selecting appropriate input features, as well as the suitable learning algorithm. They analyzed the historical time series of the S&P 500 and attempted to predict the trend based on past data. Neural networks (NN) have been shown to predict financial time series movements even when trained with only flat time series data.

Islam and Nguyen (2020) present a comparative study of stock price forecasting using three different methods: ARIMA, ANN, and stochastic process-geometric Brownian motion (GBM). Each method is used to build predictive models using historical stock data collected from Yahoo Finance. The output of each model is compared to the actual stock price. The empirical results show that the traditional statistical model and the stochastic model provide a better approximation for next-day stock price prediction than the NN model.

The objective of Usmani et al. (2016) is to ascertain whether integrating diverse techniques-encompassing statistical, analytical, and data mining methodologies- can facilitate the forecasting of stock market movements and to what extent this can be achieved with a defined level of accuracy. The objective is to predict the closing market performance of the Pakistan Karachi Stock Exchange (KSE) using a variety of ML techniques. In the prediction model, a variety of attributes are used as inputs to forecast market performance in both positive and

negative directions. The model uses a set of attributes, including oil, gold, and silver rates; interest rates; foreign exchange (FX) rates; news; and social media data. The efficacy of various ML techniques, including Single Layer Perceptron (SLP), MLP, Radial Basis Function (RBF), and SVM is evaluated through a comparative analysis. Additionally, an analysis of these features is conducted separately. The results demonstrate that the MLP algorithm exhibits superior performance compared to other techniques. It was also determined that the oil ratio feature is the most pertinent to market performance. It can thus be demonstrated that the performance of the KSE-100 index can be predicted using ML techniques.

In a similar approach, Nguyen and Yoon (2019) proposed a novel framework, called deep transfer with relevant stock information (DTRSI), which utilizes a deep neural network (DNN) and transfer learning. The initial stage of the process entails the construction of a baseline model comprising LSTM cells. This is intended to facilitate pre-training based on a substantial corpus of data drawn from a diverse array of stocks, to optimize the initial training parameters. Secondly, the baseline model aims to implement minor alterations using limited data and different input attributes from the target stock, to enhance performance. The input data include information from the five companies in the Korean and United States (US) markets with the highest market capitalization between 2012 and 2018. The experimental results show that the integrating transfer learning and stock relationship data enhances the model's predictive performance, with the proposed approach exhibiting a notable superiority in terms of prediction accuracy.

In a comparative study conducted by Shah et al. (2019), two ANNs -namely, the LSTM-RNN and the DNN models, were employed to predict the daily and weekly movements of the Indian BSE Sensex index. To prevent overfitting, measures were taken in both networks. To assess the models' generalizability, daily forecasts for Tech Mahindra (NSE: TEMPL) stock price were generated. The two networks demonstrated comparable performance in forecasting the daily price movements of Tech Mahindra. The findings indicated that the LSTM-RNN model exhibited superior performance in weekly forecasting compared to the DNN model, suggesting its potential for longer-term forecasts.

Tahersima et al. (2011) forecasted the closing price movements of the Euro to Japanese Yen (EURJPY) using an hourly dataset from September 20, 2010, to January 21, 2011. An MLP

was utilized to predict the closing price movements of the EURJPY. The MAE of the test data was employed as a criterion for evaluating the model. Nevertheless, the desired accuracy was not achieved even with a 10-step forward prediction. The NN was unable to accurately extract the dynamics of this time series. The subsequent step revealed that the primary cause of this issue was the impact of high-variance noise in the data. By employing a straightforward filter, the prediction accuracy could be significantly enhanced. The findings of this study indicate that neurocomputational models are effective tools for forecasting stock market movements in emerging markets. The results also demonstrated that the removal of noise has a significant impact on the enhancement of forecasts.

Mostafa (2010) predicted the closing price movements of the Kuwait Stock Exchange (BSI) using data from the period between 2001 and 2003. To predict BSI closing price movements, MLP and generalized regression neural networks (GRNN) were employed. The findings of this study indicate that neuro-computational models are effective instruments for forecasting stock market fluctuations in emerging markets. The results also demonstrate that the quasi-Newton training algorithm produces fewer prediction errors compared to MLP and other statistical and stochastic methods. The robustness and flexibility of neuro-computational modeling algorithms make them suitable for predicting stock market price movements. Examples of such algorithms include regression and ARIMA. It is becoming increasingly evident that traditional statistical techniques are no longer the optimal approach.

In their study, Jarrett and Kyper (2011) evaluated the efficacy of predicting fluctuating Chinese stock prices during periods of changing intertemporal economic movements. The Chinese stock market is the subject of study to gain insight into the predictive properties of the fast-growing but unstable market. The database was developed by the Pacific Basin Capital Markets (PACAP) Research Center at the University of Rhode Island (USA) and SINOFIN Information Service Inc., which is affiliated with the China Economic Center. The forecasting analysis indicates the efficacy of the ARIMA-intervention model in elucidating the precipitous decline in the price index values of Shanghai A shares during the global economic downturn in China in 2008. The results substantiate the assertion that the daily prices of Chinese stocks exhibit an autoregressive component. The application of intervention analysis is an effective method for elucidating the intricate dynamics of economic disruption and the time series alterations of a price index with precision and detail.

To supplement and expand existing knowledge on stock market forecasting, Kehind et al. (2023) conducted a scientometric study. This was done to review the progress and trends in this field of study, as well as gaining a broader understanding of bibliometric data, tools and methods. The authors employed science mapping and scientometrics techniques for the analysis of bibliometric data and the visualization of scientific knowledge and contributions within a selected field of study. A search was conducted using Scopus, the most comprehensive academic database, to identify relevant academic articles published over the past 20 years. The reviewed research employed a range of ML and DL techniques, including ANN, SVM, GA, LSTM, fuzzy logic (FL), sensitivity analysis, technical analysis, and others, to predict the stock market's future trajectory within the classification sets. Additionally, the researchers employed technical indicators and daily stock prices as inputs to predict future prices. The results of the experiments demonstrated that it is feasible to construct an efficient and effective forecasting model without undertaking complex tasks such as data preprocessing or seasonality testing. In addition to filtering, another prevalent clustering technique is fuzzy logic, which is a computational technique based on accuracy. The most frequently utilized keywords are "stock market prediction," "stock market," "ML," "DL," and "stock market prediction." A total of 220 articles were analyzed, and it was concluded that ANN and fuzzy-based techniques are the most widely adopted forecasting and clustering techniques, respectively.

In a recent study, Gülmez (2023) used a deep LSTM network optimized with the Artificial Rabbit Optimization algorithm (ARO) model (LSTM-ARO) to predict stock prices. The dataset comprises stocks from the Dow Jones Industrial Average (DJIA) index. The dataset covers the period from January 1, 2018, to January 1, 2023. The closing price of the subsequent trading day was forecast. Eighty percent of the data was utilized for training purposes, while the remaining twenty percent was reserved for testing. The LSTM-ARO model was evaluated in comparison with an ANN model, three distinct LSTM models, and a GA model. All models were evaluated by Mean Squared Error (MSE), MAE, MAPE, and coefficient of determination (R²) criteria. The results demonstrate that the LSTM-ARO model exhibits superior performance compared to the other models.

Nasiri and Ebadzadeh (2023) put forth two innovative methodologies for multi-step stock price prediction. The DCT-MFRFNN method is founded upon the discrete cosine transform (DCT) and the multifunctional recurrent fuzzy neural network (MFRFNN). They employed the DCT

to mitigate fluctuations in the time series and streamline its structure while utilizing the MFRFNN to forecast the stock price. The VMD-MFRFNN approach, which is based on variable mode decomposition (VMD) and MFRFNN, represents a synthesis of the aforementioned methods, combining their respective advantages. VMD-MFRFNN comprises two stages. In the decomposition stage, the input signal is decomposed into several intrinsic mode functions (IMFs) using the VMD technique. In the subsequent prediction and reconstruction stage, each of the IMFs is allocated to a distinct MFRFNN for prediction purposes, with the predicted signals subsequently aggregated to reconstruct the output. The efficacy of the proposed methods was evaluated using three financial time series: the Hang Seng Index (HSI), the Shanghai Stock Exchange (SSE), and the Standard & Poor's 500 Index (SPX). The experimental results demonstrate that VMD-MFRFNN exhibits superior performance compared to other state-of-the-art methods.

In a recent study, Zhang et al. (2023) proposed a CNN with BiLSTM and attention mechanisms to enhance the precision of stock price and index forecasting. In this study, data encompassing a total of 2,675 trading days of the Shanghai Shenzhen CSI 300 Index from January 4, 2011, to December 31, 2021, were utilized for stock price forecasting and closing price prediction. The selected dataset comprises the opening price, high price, low price, closing price, volume, turnover, and return. This method is compared with LSTM, CNN-LSTM, and CNN-LSTM-Attention models. The models are evaluated using three statistical metrics: MAPE, RMSE, and coefficient of determination R^2 . The results demonstrate that the CNN-BiLSTM-Attention model exhibits superior accuracy in predicting stock closing prices, outperforming the other three models.

To enhance the precision of stock price forecasting, Li et al. (2023) proposed an advanced DL framework with clustering to predict stock prices using three established DL prediction models, namely LSTM, RNN, and Gated Recurrent Unit (GRU). The fundamental concept of the proposed framework is the integration of clustering with DL models in diverse contexts into the stock price prediction framework. To achieve effective clustering, a new similarity measure, Logistic Weighted Dynamic Time Deviation (LWDTW), is proposed as an extension of the Weighted Dynamic Time Deviation (WDTW) method. The objective is to capture the relative importance of return observations when computing distance matrices. The dataset comprises comprehensive historical daily price and volume data for US stocks listed on the New York

Stock Exchange (NYSE), the NASDAQ stock market, and the NYSE MKT (formerly the American Stock Exchange, AMEX). It encompasses opening, high, low, and closing prices, as well as trading volume, for the period between 1970 and 2017. The period from January 2005 to November 2017 was selected for examination of recent trends in stock price movements. The models were evaluated using a series of statistical metrics, including MAPE, MAE, Mean Square Error (MSE), RMSE, and R2. The results of the evaluation demonstrate that LWDTW-based cluster forecasting in conjunction with the LSTM model represents the most effective methodology for stock price prediction, exhibiting the lowest MAPE (0.1278), MAE (0.0536), MSE (0.0059), RMSE (0.0745), and the highest average R2.

Kanwal et al. (2022) put forth a hybrid DL forecasting model that incorporates Bidirectional Cuda Deep Neural Network Long Short-Term Memory (BiCuDNNLSTM) and one-dimensional CNN to achieve timely and efficient forecasting. The proposed model (BiCuDNNLSTM-1dCNN) was evaluated using five stock price datasets and compared with other hybrid DL-based and state-of-the-art models for validation purposes. The models were evaluated using RMSE and MAE measures. The proposed model, BiCuDNNLSTM-1dCNN, is compared with four other DL models: LSTM-CNN, LSTM-DNN, CuDNNLSTM, and LSTM. The data set comprises three distinct categories: crude oil (30.03.1983-15.08.2018), crude oil (23.08.2000-15.01.2021), and global x dax Germany ETF (23.10.2014-31.12.2020) for individual stock items. Additionally, it includes the Frankfurt Stock Exchange DAX Performance Index (01.01.2000-16.06.2021) and the Hong Kong Stock Exchange Hang Seng Index (01.01.2000-25.06.2021) for stock market performance indices. The experimental results demonstrate that the proposed model, BiCuDNNLSTM-1dCNN, exhibits superior prediction accuracy in comparison to the other four DL models.

In a recent study, Muhammad et al. (2023) developed an ML model based on the transformer technique to predict the future price of stocks traded on the Dhaka Stock Exchange (DSE), the leading stock exchange in Bangladesh. The transformer model has been extensively employed in the domains of natural language processing and computer vision. This study focuses on the application of the transformer-based model to predict the closing price of eight specific stocks listed on the DSE stock exchange using historical daily and weekly data. The data set includes stock data from October 2012 to December 2020, sourced from Amarstock. The data set includes the following variables: trade code, date, opening price, high price, low price, closing

price, and volume. The RMSE and MAE are employed to assess the efficacy of the models. The proposed model is capable of forecasting not only the price after the trading day but also the price one week hence.

Yang et al. (2023) sought to enhance the precision of stock price forecasting. The stock trading data, including the opening price, closing price, highest price, lowest price, and trading volume, were parsed using the BeautifulSoup library. The historical data of S&P 500 technology stocks, spanning from January 1, 1990, to December 31, 2020, were utilized, amounting to a total of 30 years. Proper segmentation of the dataset is crucial, as it enhances the model's generalizability and effectiveness in real-world applications. The dataset was divided into three subsets in chronological order: 70% for training, 15% for validation, and 15% for testing. The model's prediction performance was evaluated using precision, recall, accuracy, and R^2 . Although SVM is an excellent choice for classification tasks, it may not be optimal for predicting stock prices, which involve continuous value prediction. Experimental results indicate that SVM performs slightly better than Random Forest but still significantly trails behind deep learning-based models. The Temporal Transformer, which was specifically designed for time series data and integrates traditional time series analysis techniques, outperforms both models.

Chen et al. (2023) proposed a vector autoregression (VAR)-based rolling forecast model to predict stock prices and a Gaussian feedforward neural network (GFNN)-based graph signal identification method to recognize different types of stock price signals. Stocks were randomly selected from Yahoo Finance's trading records was employed, with historical data spanning from January 28, 2019, to January 25, 2022. To facilitate a comparative analysis, the Temporal Convolutional Network (TCN), GRU, and LSTM methods were utilized. The efficacy of the model was evaluated using metrics including accuracy (Acc), MSE, and explained variance (Evar), which were calculated using specific equations. The proposed approach demonstrated superior performance compared to the GRU and LSTM methods. While the proposed method demonstrated comparable performance to TCN, it achieved higher accuracy on a greater number of datasets than TCN. Therefore, the proposed approach outperformed the other methods in comparison.

Wang et al. (2024) proposed an advanced interval-valued decomposition integration model for stock price prediction. This model is based on comprehensive feature extraction and optimized DL. To extract internal features and decompose interval values into interval trends and residues, they introduced Feedback Interval Variational Modal Decomposition (FIVMD). This method effectively performs external feature extraction using appropriate feature selection and compression techniques, identifies the most influential factors, and enhances the modeling capability for high-dimensional data. The stock indices used in this study encompass the Shenzhen Stock Exchange Index (SZI), the Shanghai Stock Exchange Composite Index (SSEC), and the China Securities 100 Index (CSI100). The dataset comprised stock data from April 10, 2014, to April 20, 2023. The training set consisted of 80% of the samples from each dataset, with the remaining 20% used as the test set. A performance analysis was conducted using three statistical metrics: RMSE, MAE, and MAPE were employed as performance metrics. The results showed that the proposed model exhibited superior accuracy and stability compared to other models. Additionally, it exhibited proficiency in forecasting interval-valued stock prices.

Various factors contribute to fluctuations in stock prices within the market. Therefore, selecting appropriate input features to ensure the efficacy of the prediction model. To address these challenges, Nguyen and Yoon (2019) proposed a novel framework for Deep Transfer Learning with Stock Information (DTRSI), which employs DNN and transfer learning. The study employed data from the five largest companies in the Korean and US markets, spanning the period from 2012 to 2018. The base model was pre-trained with LSTM cells and a substantial amount of data from diverse stocks, to optimize the initial training parameters. To enhance performance, the base model was fine-tuned with a limited amount of data from the target stock and a variety of input features. The efficacy of the proposed methodology was assessed using six distinct categories of input features: Company of Interest (COI), Index on Day (CI), CI+CS (cosine similarity), CI+SF (similar field), CI+HMC (highest market capitalization), and CI+RND (randomly on day). The results showed that the proposed approach achieved superior performance in terms of average prediction accuracy compared to SVM, RF, and k-nearest neighbors (KNN). Additionally, it was determined that stocks associated with the company of interest (COI) showed greater efficacy in financial time series prediction tasks.

Zhong and Enke (2019) conducted a comprehensive study employing big data analytics and hybrid ML algorithms to predict the daily return direction of the SPDR S&P 500 ETF. Among the ML techniques addressed in this series of studies, the combination of PCA-DNN classifiers with an appropriate number of hidden layers demonstrated the highest level of classification accuracy.

Yadav and Jha (2020) selected ICICI Bank, Reliance Industries, TCS, and Maruti from various sectors within the National Stock Exchange Fifty (NIFTY 50) stocks to demonstrate the performance of LSTM on time series affected by various factors. The researchers used a dataset of 2560 days of observations from December 29, 2008, and May 24, 2019, to predict the closing prices of the four companies under consideration. The MSE was used as the loss function, and the Adam (Adaptive Moment Estimation) algorithm was applied for optimization in the LSTM models. The Root Mean Squared Error (RMSE) was used to assess the efficacy of the models. The results demonstrated that the optimal configuration in terms of average RMSE was achieved with a hidden layer count of $n = 1$. This finding was confirmed by a one-way ANOVA test. The results indicate that a single hidden layer is the optimal configuration for most cases, as it offers enhanced accuracy, simplified training, and a reduced risk of overfitting.

Anand (2021) used stock data from Bharti Airtel, Tata Steel, HCL, Infosys, and Bajaj Fiserv, which are traded on the NSE of India. The dataset included variables, including the number of trades, turnover, total trade quantity, average price, closing price, last price, lowest price, highest price, opening price, previous close, stock date, stock series, and stock symbol. These variables were used in a predictive model to forecast the closing price. The forecasting was carried out using various DL architectures, including CNN, LSTM, RNN, MLP, and SVM. The results showed that the ARIMA model had superior predictive accuracy compared to the DL models.

In a recent study, Mehtab and Sen (2020) proposed a novel hybrid approach for stock price prediction, integrating ML and DL methods. The researchers used data from the NIFTY 50 index of the NSE of India for the period spanning 2015 to 2018. A one-week forecast was created to predict the closing value of the NIFTY 50 for the year 2019. The study concluded that CNN-based DL models performed better than machine learning-based predictive models.

In their 2019 study, Şener and Şener examined the closing values of airline stocks at the end of the trading day. They conducted a comparative analysis between the Turkish and American stock markets on a TL and USD basis, employing the Monte Carlo Simulation method. The objective of this study is to propose an alternative methodology for examining the impact of fluctuations in the USD exchange rate on the Turkish stock market. A notable discrepancy was observed in the annual volatility values of TA and American airlines in response to fluctuations in the dollar exchange rate. The volatility value for American Airlines was observed to be 0.400099, while that for TA was observed to be 12.03305. While American Airlines stock data exhibited minimal fluctuation in financial markets, marginal fluctuations were observed due to fluctuations in the dollar/TL exchange rate of TA. As volatility increases, so too does risk. The simulation study concluded that the Turkish Stock Market will lose value against the dollar. Additionally, an environment suitable for short-term investment will not be created.

Şişmanoğlu et al. (2020) developed a stock prediction system utilizing three distinct DL models, namely LSTM, GRU, and BiLSTM. The data set comprised time series values of stocks from the New York Stock Exchange, spanning the period from 1968 to 2018. Furthermore, test studies were conducted with International Business Machines Corporation (IBM) stock. The error measurement models RMSE, MSE, MAE, and Directional Accuracy (DA) were employed to assess the efficacy of the models utilized in the study. Among the models, BiLSTM trained with 5-day inputs demonstrated the most optimal performance, exhibiting a directional accuracy rate of 63.54% and an RMSE value of 0.01.

Islam and Nguyen (2020) conducted a comparative study for stock price prediction using three distinct methodologies. The methods employed were ARIMA, ANN, and GBM. The study employed S&P 500 daily stock data for the period between January 1, 2015, and December 31, 2019. The dataset comprises six variables. The dataset comprises six variables: daily opening, high, low, closing, volume, and adjusted closing price. In addition to the aforementioned six variables, two further variables were created: average and return. A predictive model was constructed to forecast the adjusted closing price for the subsequent day based on the variables that were forecast for that day. To facilitate comparison, absolute percentage error (APE), average absolute error (AEE), average relative percentage error (ARPE), and RMSE error measurements were calculated for each method. The analysis of the S&P 500 index yielded the

conclusion that the ARIMA and stochastic model-GBM models exhibited superior performance to the ANN models in short-term next-day stock price prediction.

Nikou et al. (2019) employed a range of advanced ML algorithms, including RF, SVM, LSTM, and ANN models, to predict the closing price of the iShares MSCI UK. The daily closing price data of the iShares MSCI UK exchange-traded fund between 2015 and 2018 were used to predict daily closing prices using RF, SVM, LSTM, and ANN models. The models were evaluated using MAE, MSE, and RMSE as error measurements. The findings indicate that the LSTM blocked recurrent network approach exhibits superior performance in forecasting the iShares MSCI UK closing price compared to other methodologies. Additionally, the SVR method demonstrates heightened sensitivity compared to both the neural network and RF.

The majority of stock information is derived from sector classification data sourced from third-party environments. It should be noted, however, that these data are generally approximate. Zhao et al. (2023) put forth a time series relational model (TSRM) that integrates temporal and relational data. The TSRM employs a K-means model to automatically derive stock classification based on trading data. A total of 87 stocks were scanned in Shanghai, while 68 stocks were scanned in Shenzhen. The experimental dates selected for analysis in this study range from January 1, 2019 to September 1, 2020. The experimental period includes the training set (January 1, 2019 to January 1, 2020), the validation set (January 1, 2020 to March 1, 2020), and the testing set (March 1, 2020). The data set is divided into three intervals, beginning on September 1, 2020. The selected stock data properties were the opening price, closing price, high price, low price, volume, and turnover rate. It is typical, widespread, and easily accessible, and can be listed on any exchange. The time series information, extracted using the LSTM model, and the relationship information, extracted by the graph convolutional network (GCN), are integrated to predict stock prices. The results demonstrate that the TSRM model exhibits a notable enhancement in cumulative returns and maximum drawdown relative to conventional LSTM models within the Chinese Shanghai and Shenzhen stock markets.

3. MATERIAL AND METHOD

The data set utilized in this study encompasses the daily closing prices of a particular stock traded on the Istanbul Stock Exchange BIST100. The data set was gathered for four years, from 2020 to 2023. After each trading day, the market is surveyed to ascertain the daily closing prices, which are then duly recorded. Furthermore, the data set incorporates technical indicators such as moving averages (MA), Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD). This data has been employed to analyze and predict changes in stock prices over time. In the Section 3.1., information about ML is given. 3.2. Detailed information about DL is given in section. In the section 3.3., detailed information is given about the ANN method, its history, biological structure, architectural structure, educational process, types of learning, and backpropagation for the overfitting problem encountered in the learning process. 3.4. Detailed information about the CNN method is given in the section. 3.5. Detailed information about the RNN method is given in the section. 3.6. In this section, detailed information about the working principle of the LSTM method is given. 3.7. Detailed information about the BiLSTM method is given in the section. 3.8. Detailed information about the hybrid CNN-LSTM method is given in the section.

3.1. Machine Learning

The advent of digitalization has brought about the emergence of new terms and concepts, including the Internet of Things (IoT), cloud computing, big data, cybersecurity, AI, and machine learning, which have become integral aspects of our lives. The extraction of the useful part of a data set allows for the prediction of future events. Future projections are based on the experiences and training received. Nevertheless, individuals often encounter difficulties when attempting to make rational decisions, due to the influence of emotional factors. Furthermore, when an excessive amount of data is available, the failure to consider pertinent details may result in erroneous conclusions. However, machines that have been trained to perform a specific task can evaluate potential scenarios and examine millions of data points with great speed, enabling them to make rational decisions in a very short time. In the context of ML, it is of paramount importance to define the human mindset and the scope of decision-making authority. ML is the process of creating a model with a data-appropriate algorithm that is designed to imitate human intelligence (Tosunoğlu, 2021). ML is defined as the capacity of systems to learn

from problem-specific training data, thereby automating the process of building analytical models and solving related tasks (Janiesch, 2021). ML is a field of study that aims to teach machines how to process data in an efficient manner. The objective of ML is to enable computers to learn from data. (Mahesh, 2020). Consequently, the objective is to automate the process of developing analytical models to perform cognitive tasks, such as object detection or natural language translation. ML demonstrates considerable applicability, particularly in the context of high-dimensional data, such as classification, regression, and clustering. It assists in the generation of reliable and reproducible decisions by learning from previous calculations and extracting meaningful data from extensive databases (Janiesch, 2021).

3.2. Deep Learning (DL)

The foundation of DL algorithms is rooted in the computational modeling of neurons within the human brain. The field of DL has its roots in the work of Donald Hebb on the structure of NN. Subsequently, the construction of a mathematical model within a computer environment represents the inaugural stage of ANN. Since their inception, ANNs have undergone significant advancements. In recent years, the term "deep learning" has emerged as a prominent concept in this field. Moreover, the majority of studies have concentrated on this topic (Doğan, 2018).

DL represents a subcategory of ML. DL employs a multitude of non-linear processing units to facilitate feature extraction and transformation. The output of the preceding layer serves as the input for the subsequent layer. In the context of DL, a structure is employed that enables the learning of multiple levels of features or the representation of data. The derivation of high-level features from lower-level features results in the formation of a hierarchical representation. This representation learns multiple levels of representation, corresponding to different levels of abstraction. The fundamental premise of DL is the acquisition of knowledge through the interpretation of data representation. In the context of image representation, a vector of intensity values per pixel or features such as edge clusters, special shapes, and so forth can be taken into account. Some of these features are more effective at representing the data (Şeker, 2017).

The specific characteristics of ANNs are contingent upon the intended application and context in which they are utilized. The optimal model is contingent upon the nature of the problems at hand. Classification, detection, diagnosis, prediction, and so forth. The selection of an

appropriate NN model is crucial for achieving optimal results. Furthermore, the optimal NN model may vary depending on the intended use of the data. Furthermore, the optimal neural network model can be determined based on the data type, which may be image, sound, or signal (Doğan, 2019).

1. ANN
2. CNN
3. RNN
4. LSTM
5. BiLSTM
6. CNN-BLSTM

3.3. Artificial Neural Networks

3.3.1. Introduction and History of ANN

For centuries, researchers have sought to develop methods for animating and activating inanimate objects. In light of the growing need for data storage, research in the fields of AI and ANN has seen a notable increase. The ANN model was initially developed by Warren McCulloch and Walter Pitts in 1943 (Keskenler, 2017). The concept of AI was first articulated at a conference held at Dartmouth College in Hanover, New Hampshire, in 1956 (Öztürk & Şahin, 2018). AI refers to the capacity of computers to perform tasks associated with logical processes, including problem-solving, comprehension, interpretation, generalization, and learning from past experiences, utilizing characteristics observed in humans (Öztürk, 2016). In 1943, McCulloch and Pitts established the theoretical framework for ANN by developing the fundamental structure of the first ANN cell. They demonstrated that any logical expression can be formulated using an ANN.

In their 1943 publication, McCulloch and Pitts articulated the mathematical theory of neural behavior as follows:

1. A neuron can be driven to emit a signal that can be either on or off.
2. Neurons possess activation functions.
3. Upon stimulation of the inputs, the resulting output is received instantaneously.

4. When the inputs to the neuron are blocked, the stimulation of the neuron is prevented at that moment.
5. The configuration of neuronal networks within the nervous system remains unaltered.

Moreover, neurons receive inputs, perform multiplication and addition with weights, and then generate an output through the application of a specific transfer function.

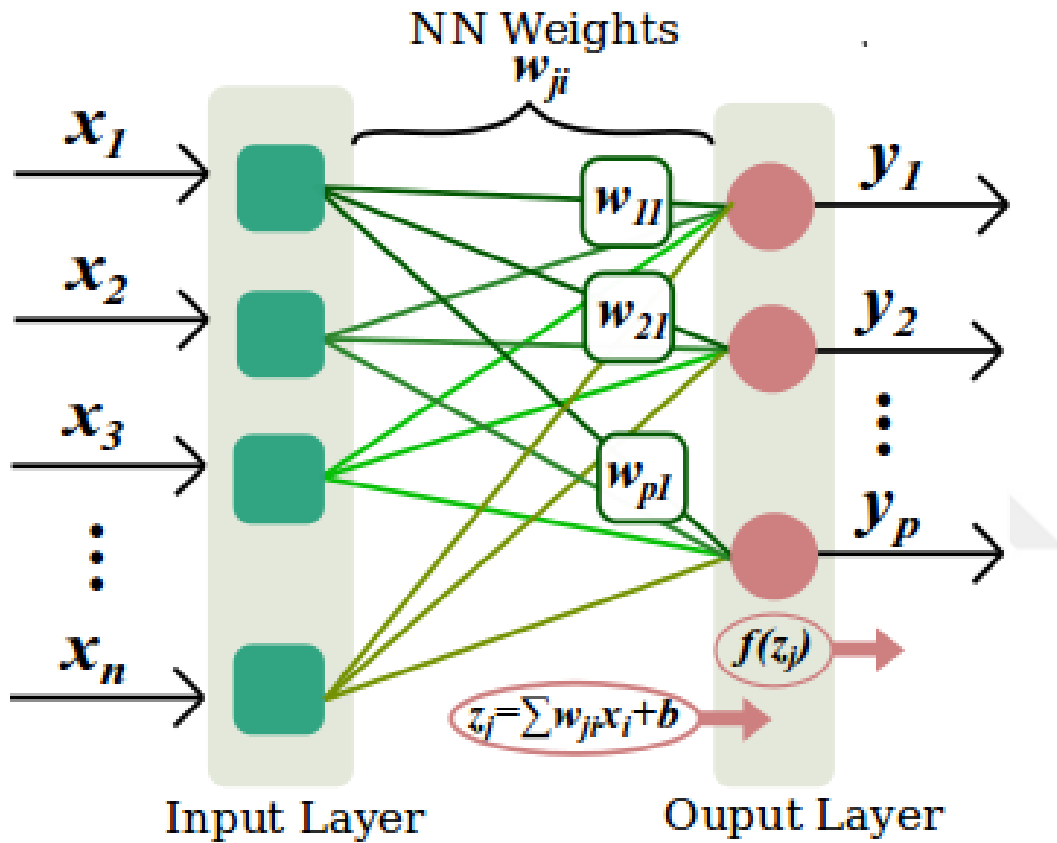


Figure 3.1. The single-layer network structure (Melli, 2017)

$$y_p = \sum_{i=1}^n (w_{ji} * x_i) + b \quad (3.1)$$

In the equation 3.1, the y_p value represents the output signal of neuron i ; the variable w_{ij} denotes the synaptic connection between neuron i and neuron j ; and the variable x_i signifies the output of neuron j . The aforementioned formula is illustrated in Figure 3.1. An input vector $(i_1, \dots, i_n)$, comprising variables from the existing system, is multiplied by the weight vector $(w_1, \dots, w_p)$ associated with the neuron. The output of neuron i is equal to the signals originating from

neuron j and directed to neuron i , multiplied by the weights and the sum of all products. The term refers to a constant value that is added to the input of the b activation function, which is also known as the bias (McCulloch and Pitts, 1943).

Subsequently, Frank Rosenblatt proposed the perceptron, a rudimentary ANN model, in 1957. The perceptron takes the input data and multiplies it by the weights, applies a threshold value, and produces an output. Nevertheless, Rosenblatt's perceptron model is employed for the resolution of relatively straightforward classification issues.

ANN is employed in many applications, including signal analysis, variable selection, classification of data, and pattern discovery. The ability of this model to learn from data sets by observation makes it applicable to a wide range of areas.

3.3.2. Biological Infrastructure and Architecture

ANN constitutes one of the most widely used ML algorithms in the field of binary classification, with applications including the prediction of stock market indices (Ayyıldız & İskenderoğlu, 2024). The concept of AI draws inspiration from the intricate workings of the human brain (Agatonovic-Kustrin & Beresford, 2000). They are composed of layers of interconnected nodes, or "neurons," which process and transmit information. Each neuron receives input from other neurons and produces an output by applying a mathematical function to that input. Subsequently, the output is transferred to other neurons in the subsequent layer (Gülmez, 2023; Ayyıldız & İskenderoğlu, 2024). The acquisition of logical and inductive knowledge is facilitated by methods, tools, and systems that emulate the processes of human reasoning based on brain activity, thereby enabling problem-solving (Agatonovic-Kustrin & Beresford, 2000). The nervous system is comprised of a network of neurons, which are structured around a soma and an axon. Synapses are typically situated between the axon of one neuron and the soma of another neuron. For a neuron to elicit a response at a specific point in time, the excitation threshold must be surpassed. The threshold value is intrinsic to the neuron and remains constant regardless of the location or timing of the stimulation. The impulse initiated at the stimulation point propagates throughout the neuron (see Figure 3.2.)(McCulloch & Pitts, 1943).

Neuron

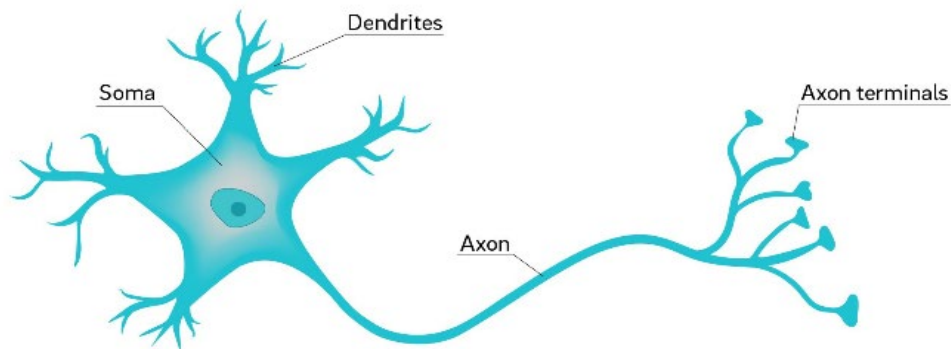


Figure 3.2.The neural network (Rakesh, 2018)

Two principal categories of AI enhancements can be identified. The initial category encompasses methodologies and systems that emulate human experience and derive conclusions from a set of rules, exemplified by expert systems. The second category of AI improvements encompasses systems that model the functioning of the brain. One such system is the ANN (Figure 3.2). Expert systems are knowledge-based systems that represent an extension of traditional computing. They are sometimes referred to as "fifth generation computing." The knowledge base enables an expert to define rules that simulate a thought process. It offers a straightforward method for reaching conclusions and resolving issues by adhering to a prescribed set of rules. The fundamental premise of expert systems is that logical reasoning can be represented by a set of logical propositions and a series of logical operations applied to these propositions. ANNs are digitized models of the human brain. They are computer programs designed to simulate how the human brain processes information. ANNs learn or are trained not through programming but through experience with appropriate learning examples, in a manner analogous to that observed in humans (Agatonovic-Kustrin & Beresford, 2000).

A neuron in an ANN is composed of several different elements. These are the elements in question: Inputs are defined as the data that are fed into the neuron, which are typically numerical values. Weights are values that are assigned to each input, and they are used to determine the importance of that input in the overall computation performed by the neuron.

Bias is defined as a constant value that is added to the weighted sum of the inputs. The activation function is a mathematical function that determines the output of the neuron depending on the inputs and weights. The most prevalent activation functions employed in ANN are the softplus, sigmoid, tanh, and ReLU functions. The output represents the final result of the neuron's calculation and is subsequently transmitted to other neurons within the network. The error is defined as the discrepancy between the predicted output of the neuron and the actual output. This is utilized to adjust weights and biases to enhance the accuracy of the network (see Figure 3.3.)(Gülmez, 2023).

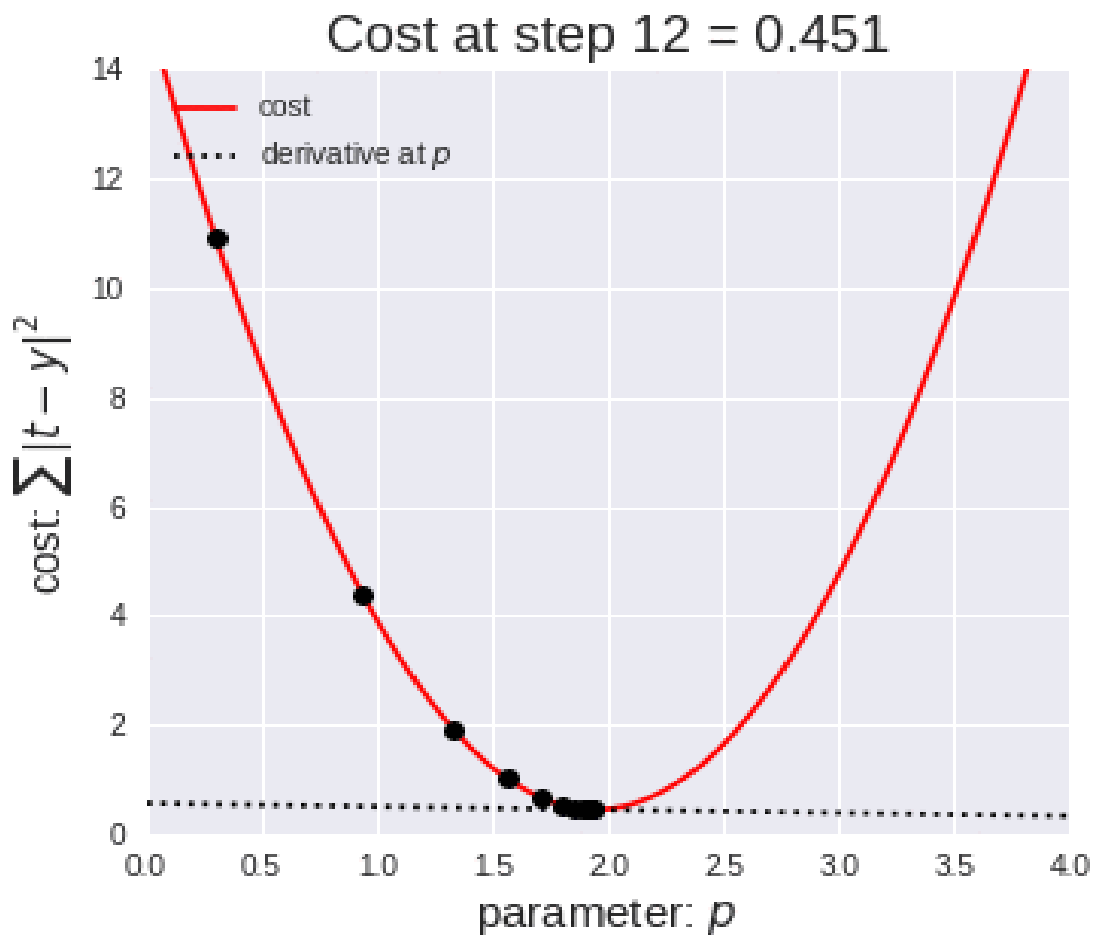


Figure 3.3. Adjust weights and biases (Tekyaygil, 2020)

A layer in an ANN is defined as a group of interconnected neurons that function collectively to process and transmit information. An ANN typically comprises multiple layers, with the input layer receiving the raw data and the output layer producing the final result. The intermediate layers, situated between the input and output layers, are responsible for performing the requisite

calculations and transformations on the input data, thereby producing the desired output. As illustrated in the Figure 3.4., the number of hidden layers and the number of neurons in each layer may vary depending on the specific task the ANN is designed to perform (Gülmez, 2023).

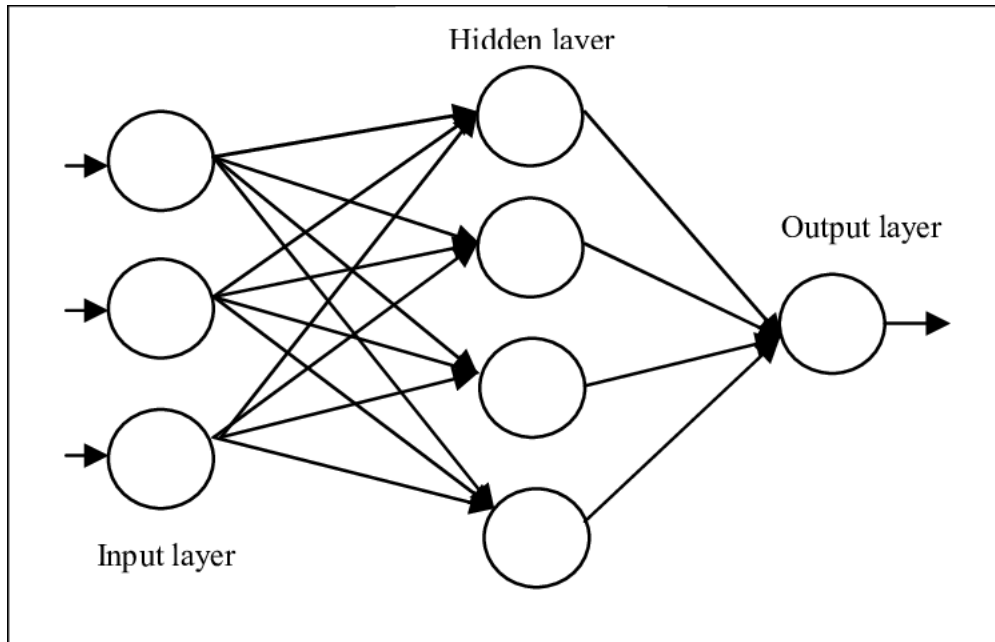


Figure 3.4.The number of hidden layers and the number of neurons (Vui et al., 2013)

3.3.3. ANN Structure

In a single-layer ANN system, the perceptron represents the fundamental unit. The structure and functionality of ANNs can be modified through a process known as training. During the training phase, the network is presented with a set of inputs and the corresponding outputs. The network adjusts its weights and biases to minimize the discrepancy between the predicted output and the actual output. This process enables the network to "learn" and enhance its performance on tasks such as classification, regression, and clustering. A neuron, also referred to as a perceptron or processing element, represents the fundamental computational unit within an artificial neural network. In essence, the neuron receives input from other neurons or external sources, processes the input using a set of weights and biases, and produces an output based on this processing. If the result exceeds the threshold value, the output is 1; otherwise, it is -1. A neuronal model comprises three fundamental features (Gülcü, 2022). These three fundamental

characteristics (arousal and threshold value, synaptic weights, and the final one, timing and temporal dynamics) facilitate a comprehensive understanding of the intricate behaviors of the nervous system by enhancing the biological fidelity and functionality of neuronal models.

As illustrated in Figure 3.5., neurons possess a set of weights. To illustrate, the input signal, denoted by x_i , is linked to the weight associated with the w_i neuron. In other words, the input signal will be multiplied by the relevant weight. It is crucial to assign names to the weights. In the index of weight, the first indicator refers to the neuron, while the second indicator refers to the input. The sign of a weight may be positive or negative (Palit & Popovic, 2005).

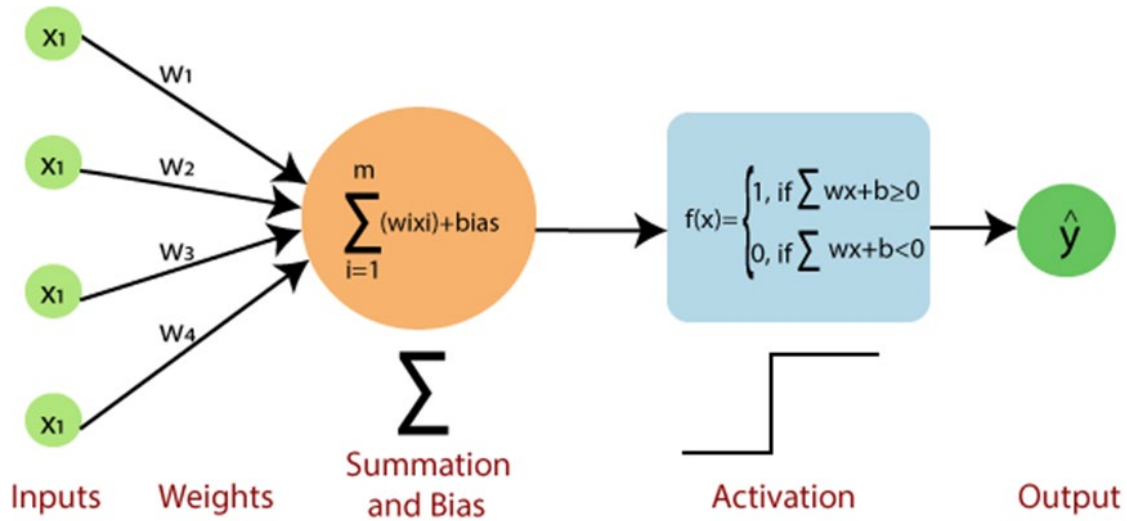


Figure 3.5.Functionality of neuronal models (Kılıç, 2023)

$$z_j^i = w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + \dots + w_m * x_m \rightarrow \sum_{i=1}^m (w_i * x_i) + b \quad (3.2)$$

$$y = f(w_i * x_i + b) \quad (3.3)$$

In these equations 3.2 and 3.3, $x_1, x_2, \dots, x_m$ represent the input signals, while $w_1, w_2, \dots, w_m$ denote the weights of the neuron. The term z_{ji} signifies the sum of the input signals, whereas the slope of b represents the activation function. Finally, y denotes the output of the neuron (Dökmen, 2023).

The ANN is capable of forecasting future price movements employing an analysis of the historical data of stock prices. ANNs can be continually trained with new data, enabling them to adapt to shifts in market conditions. In comparison to statistical methodologies, ANNs tend

to demonstrate superior predictive capabilities. ANNs are well-suited for modeling data structures that are often nonlinear and complex, such as those observed in stock markets. Furthermore, real-time data analysis enables the prompt formulation of decisions (Islam, 2020). The application of ANN to the prediction of stock prices is not without its challenges. These include the following: 1) data preparation, 2) model training and validation with feature selection. The primary objective of partitioning the data set into two distinct subsets, namely the training and test data sets, is to assess the model's capacity for generalization and to mitigate the potential for overfitting. The division of the data set into two subsets allows for the evaluation of the model's performance on real-world data, measurement of its ability to generalize, and prevention of overfitting.

3.3.4. Training of ANN

In the context of ANN, the training dataset is employed to facilitate the learning process of the model, the validation dataset is utilized to assess the model's performance and implement necessary adjustments, and the testing dataset is utilized to evaluate the model's final performance. A training dataset is a data set utilized by an ML model for learning. As the model processes the data set, it learns the relationship between the input data and the desired outputs, subsequently adjusting its weights. Consequently, the model is able to make precise predictions when presented with novel data. The validation set is employed for the assessment of the model's performance during the training phase. During the training process, the model is periodically evaluated with the validation dataset to prevent overfitting. Subsequently, the model's performance is evaluated on the validation dataset, and any necessary adjustments are made to the hyperparameters. Hyperparameters are values that are manually adjusted during the learning process of the model and are determined before the commencement of the training process. These values determine how the model is trained and the structure of the model itself. The learning rate is the degree to which the model alters its weights at each stage of the learning process. The number of epochs defines the number of times the model will be trained on the entirety of the training data set. The term "batch" denotes the number of samples that are processed at each update step. Furthermore, the data set may be divided into smaller groups, and the weights may be updated after each batch. The number of layers and the number of neurons are structural parameters, such as the depth and width of neural networks, that define the architecture of the network. Regularization parameters are utilized to prevent the model

from exhibiting overfitting tendencies. Once the optimal values for the hyperparameters have been identified, the Adam algorithm and other optimization techniques are employed to train the model. The Adam optimization algorithm is a prevalent method for optimizing DL and ANN training. The Adam optimization algorithm updates the model parameters with the specified hyperparameters and optimizes the learning process. It facilitates an enhancement in the overall performance of the model concerning its parameters, while also enabling more effective generalization against new data. The test set is the data set utilized to assess the model's ultimate performance. Following the training and validation phases, the model is evaluated on the aforementioned data set, with particular attention paid to its overall performance and capacity for generalization.

One of the most fundamental capabilities of ANN is its capacity to process and predict nonlinear events. Multilayer perceptrons have been developed as a means of overcoming the inherent difficulties posed by nonlinear problems. Multilayer perceptrons (MLP) are composed of three fundamental layers. The input layer is the initial receptacle for information, the output layer is the processing site for information entering from the input layer, and the intermediate layer is the site of processing information from the input layer.

The number of intermediate layers between the input and output layers is dependent on the specific problem at hand. However, there is no established method for determining the optimal number of intermediate layers. Instead, this number is typically determined through a process of trial and error. Once the structure of the ANN has been established, the training process can commence. In the course of the training process, the weights are determined randomly (Özkaynak, 2003). A variety of learning rules exist, but the most prevalent is the delta rule, also known as backpropagation. The learning rules employed in the training and optimization of ANN are of significant consequence. A NN is trained to map a set of input data through an iterative process of adjusting weights. Information from the inputs is conveyed forward through the network to optimize the weights between neurons. The optimization of weights is achieved through the propagation of the error in a backward direction during the training or learning phase. ANN reads the input and output values in the training data set and modifies the value of the weighted connections to reduce the discrepancy between the predicted and target values. The discrepancy between the predicted and target values is reduced over numerous training cycles until the network attains a specified level of accuracy. Nevertheless, if a network is

permitted to undergo training for an excessive duration, it will become overtrained and consequently lose its capacity for generalization (Ayyıldız, 2024; Agatonovic, 2000).

3.3.5. Types of Learning

Types of learning fall into three main categories. These are supervised learning, unsupervised learning, and semisupervised learning. These categories are classified based on how the model is trained and how it interacts with data.

3.3.5.1. Supervised Learning

The training of ANN is not a process that can be completed in a single step. During the learning process, the network generates outputs in response to the presentation of inputs. The learning rate is a requisite element of the educational process. This ratio indicates the extent to which the weights must be modified to facilitate learning. It is not the case that the previous weight is removed from the system as a whole. In contrast, the discrepancy between the new and previous weights is calculated, and an adjustment is made to the original weight (Agatonovic-Kustrin, 2000; Özçalıcı, 2016). As MLP networks employ supervised learning strategies, both the inputs and the outputs that the network is expected to generate in response to those inputs are presented to the network during the training phase. The learning rule employed is to minimize the discrepancy between the outputs generated by the network during training and the outputs it should produce over time, by distributing the error to the weights of the network (Özmerkez, 2003). If such an error can be presented to the training function, then supervised training is indicated. In supervised training, ANN undergo continual weight modification until they generate actual values (Agatonovic-Kustrin, 2000; Özçalıcı, 2016). This type of calculation is referred to as a forward calculation. The weights are adjusted through a process of comparison between the actual output and the desired output, with the resulting discrepancy being distributed in a backward direction. This type of calculation is referred to as a backward calculation (Öztemel, 2003). Many factors can be identified that impact the multi-layer sensor network's capacity to learn events.

1. The following samples are presented:
2. Inputs and outputs to the network
3. Numerical representation of inputs and outputs

4. Assignment of initial values of weights
5. Determination of learning and momentum coefficients
6. Presentation of samples to the network
7. Times of changing weights
8. Scaling of inputs and outputs
9. Determination of the stopping criterion
10. Growth and pruning of networks

To select the most appropriate approach for a given solution, it is essential to consider the aforementioned factors when creating network designs. To evaluate the efficacy of the network training, post-training assessments are conducted by presenting the network with examples that were not included in the training data set. If the network produces accurate results for samples it has not previously encountered, it can be concluded that its performance is satisfactory. As demonstrated by Özkaynak (2003), MLP networks have been successfully applied in many fields, including classification, prediction, recognition, interpretation, and diagnosis.

3.3.5.2. Semi-Supervised Learning

Semi-supervised learning represents a class of supervised learning tasks and techniques that also permit the utilization of input data (unlabeled) for training purposes. Data labeling is the process of marking and assigning meaning to visual, text, or audio data by identifying and categorizing its features. Semi-supervised learning represents a combination of supervised and unsupervised learning systems. It is not uncommon for unlabeled data to exceed labeled data in quantity. A substantial body of research indicates that the incorporation of unlabeled data, even in small quantities, can markedly enhance the accuracy of learning outcomes (Onan, 2016).

3.3.5.3. Unsupervised Learning

The process of unsupervised learning allows for the discovery of insights and relationships within unlabeled data within a given system. In the process of training the network, unlabeled data is employed. The objective of employing unlabeled data is to utilize it for comprehension, identification, and discovery. For this reason, the anticipated outcomes of the system are typically unknown in advance, and the training process is conducted without supervision. It is not possible to achieve education with correct answers using this model. He is required to

identify patterns independently. In the context of unsupervised learning, the system generates outcomes based on the principles of similarity and difference, whereby the available data is grouped according to its intrinsic characteristics. The data is then divided into discrete groups, with similar or dissimilar items placed in separate lists (Dike, 2018).

3.3.6. Gradient Descent and the Delta Rule

While the perceptron rule is effective in identifying the weight vector when the training samples are linear, it may encounter difficulties in converging if the samples are nonlinear. A second training rule, designated as "Delta," has been developed to overcome this challenge. If the training samples are not linearly separable, the delta rule facilitates convergence towards the most appropriate approach to the target.

The fundamental premise of the delta rule is to utilize gradient descent to navigate the hypothesis space of potential weight vectors, thereby identifying the weights that most closely align with the training examples. Gradient descent serves as the foundation for the backpropagation algorithm, which is capable of learning networks comprising numerous interconnected units.

An understanding of the delta training rule can be achieved by considering the task of training a non-threshold perceptron. This expression represents a linear unit in which the output is provided 3.3.

$$o(x) = w * x \quad (3.4)$$

A linear unit is analogous to the initial stage of a perceptron devoid of a threshold. In linear units, the training error is employed as a means of deriving a learning rule for the weight vector, based on the training samples. There are numerous potential avenues for circumventing this issue. However, the most efficacious approach is as follows:

$$E(x) = \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2 \quad (3.5)$$

In equation 3.5, D represents the training sample set and the target output for training sample d. Additionally, D denotes the output of the linear unit for the training sample d. E(x) is defined as the square of the difference between the target output t_d and the linear unit output o_d , which

is equal to half the sum of all training samples. In this context, E is defined as a function of w , given that the linear output is contingent upon the weight vector. Concurrently, E is contingent upon the particular set of training examples, yet it is presumed to be unchanging throughout the training process.

The gradient descent algorithm can be expressed as a function of the space of possible weight vectors and the entire hypothesis (see Figure 3.6.). In this context, w_0 and w_1 represent the two possible weights of the simple linear unit. In consequence, w_0 and w_1 signify the entirety of the hypothesis space. The vertical axis, designated as E , represents a fixed training error. Consequently, the error level depicted in the figure represents the desirability of each weight vector within the hypothesis space. The objective is to derive a hypothesis with the lowest possible error. For linear units, the error surface must always be parabolic with a single global minimum.

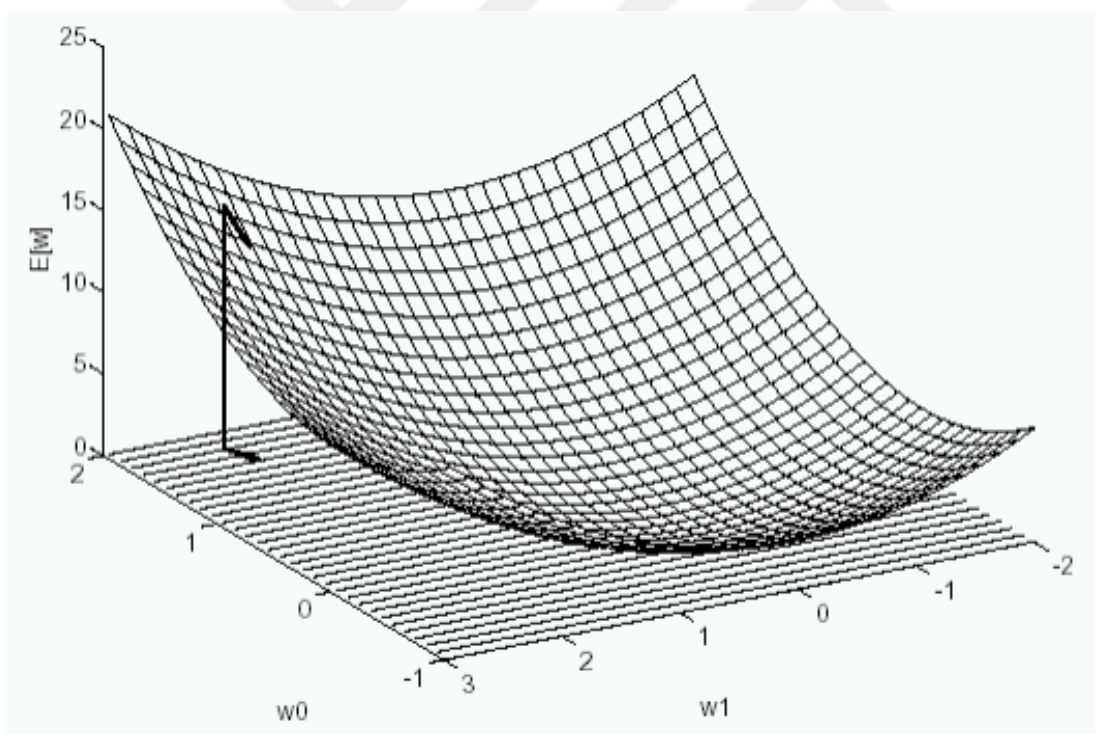


Figure 3.6. A function of the space of possible weight vectors (Mitchell, 1997)

3.3.6.1. Derivation of The Gradient Descent Rule

The direction of the steepest descent along the error level can be calculated by taking the derivative of the E function, which depends on each component of the w vector. This vector derivative is referred to as the gradient of E.

$$\Delta E(w) = \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_n} \right] \quad (3.6)$$

In equation 3.6, the vector variable $\Delta E(w)$ is defined as the partial derivative of E and w_n for each. When interpreted as a vector in weight space, the gradient denotes the direction that produces the steepest increase in E. Consequently, the training rule of gradient descent is expressed as follows in equation 3.7, as the gradient determines the direction of the steepest increase of E.

$$w \leftarrow w + \Delta w \quad (3.7)$$

$$\overrightarrow{\Delta w} = -n \Delta E(\overrightarrow{w}) \quad (3.8)$$

Equation 3.8 defines n as a positive constant, referred to as the learning rate, which determines the step size between gradient descent. The negative sign in the formula is a consequence of the weight vector moving in the direction of decreasing E. Furthermore, this formula can be expressed in the form of equations 3.9 and 3.10.

$$w_i \leftarrow w_i + \Delta w_i \quad (3.9)$$

$$\Delta w_i = -n \frac{\partial E}{\partial w_i} \quad (3.10)$$

The slope is calculated by taking the derivative of the weights iteratively.

$$\frac{\partial E}{\partial w_i} = \frac{\partial}{\partial w_i} \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2 \quad (3.11)$$

$$= \frac{1}{2} \sum_{d \in D} \frac{\partial}{\partial w_i} (t_d - o_d)^2 \quad (3.12)$$

$$= \frac{1}{2} \sum_{d \in D} 2(t_d - o_d) \frac{\partial}{\partial w_i} (t_d - o_d) \quad (3.13)$$

$$= \sum_{d \in D} (t_d - o_d) \frac{\partial}{\partial w_i} (t_d - w * x_d) \quad (3.14)$$

$$\frac{\partial}{\partial w_i} = \sum_{d \in D} (t_d - o_d) (-x_{id}) \quad (3.15)$$

The subscript " x_{id} " denotes the single input component, " x_i " for the training sample.

$$\Delta w_i = n \sum_{d \in D} (t_d - o_d) x_{id} \quad (3.16)$$

Linear unit training is like a gradient descent algorithm. A random weight vector is chosen. The linear unit is applied to all training examples, then the Δw_i value is calculated for each weight. Each w_i weight is updated by adding Δw_i and this process is repeated. This process is shown in the table 3.1 pseudo code. The error surface contains only a single global minimum. Therefore, if a small learning rate is used, the algorithm will converge to a weight vector with minimal error even if the training samples are not linearly separable. A common modification of the algorithm is to gradually decrease the value of n as the number of gradient descent steps increases (Mitchell, 1997).

Table 3.1. Gradient descent pseudocode

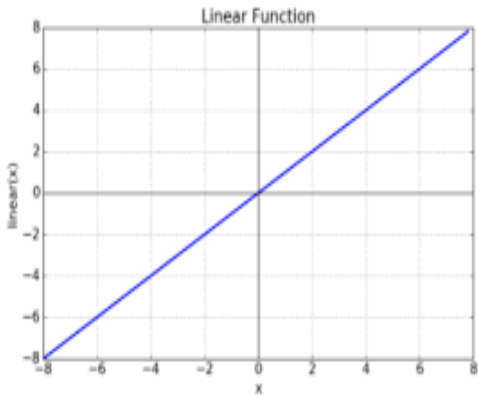
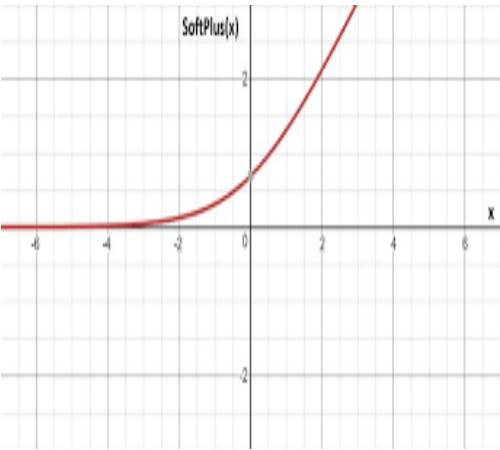
Beginning:
-Generate random weights and a bias
-Set a learning rate
-Determine the number of iterations (epochs)
Learning:
Start loop (from 1 to epochs):
Calculate estimates:
- Multiply data points by weights and add bias
- Get prediction values
Calculate error:
- Calculate the difference between predictions and actual values
Update weights and bias:
- Calculate the gradient of the error
- Update weights and bias based on gradient
- Update: weight = weight - (learning rate * weight_gradient)
- Update: bias = bias - (learning rate * bias_gradient)
Exit:
Trained weights and return bias

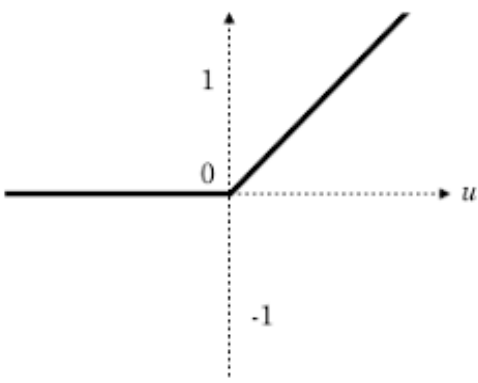
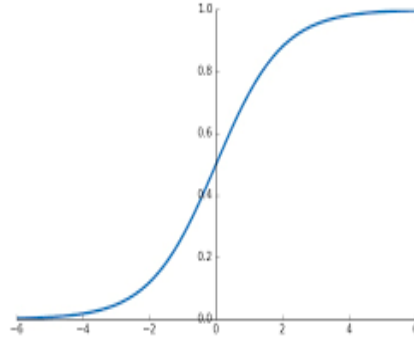
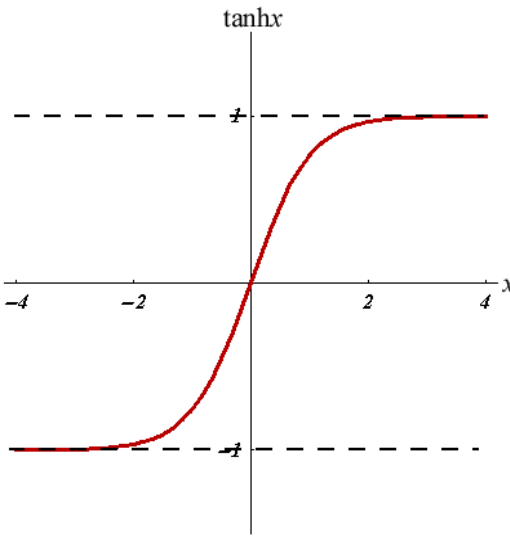
3.3.7. Activation Function Types

Activation functions are used when processing the input data of the ANN and passing it to the next layer. The threshold used to determine whether the output is passed to other layers must be determined. The value of the input to the artificial neuron can be in the range $-\infty, +\infty$, so the neuron may not know the limits of the real value. The neuron needs activation functions to decide whether it is active or not (Ser, 2019). It is usually nonlinear. It is important to choose the right function in the model. Therefore, it will affect the output performance. When applying the backpropagation algorithm in the learning process of the model, it is important to use the activation function whose derivative can be easily calculated for the speed of the algorithm.

Linear, softplus, relu, sigmoid, tangent hyperbolic activation functions used in Table 3.2 are some of them.

Table 3.2. Activation functions

Activation Functions	Graphic	Explanation and Formula
Linear Transfer Function		$y=x$
Softplus		Produces differentiable output that produces values between $\log(1 + e^x)$ 0 and $+\infty$

Rectified Linear(RELU)	$f(u) = \max(0, u)$ 	$f(x) = \max(0, x)$ If the values obtained after the activation function operations are negative, the result is 0, and when positive results are obtained, the result is 1.
Sigmoid		$f(x) = \frac{e^x}{e^x + 1}$ It is also known as the logistic activation function. It produces values between 0 and 1 (Kaytan, 2022).
Tanh		$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ The function produces values between -1 and +1 for each of the input values (Ser, 2019).

3.3.8. Backpropagation

The backpropagation algorithm is the process of propagating errors from the output layers to the input layer during the learning process. In the ANN network, training is done by adjusting the weights during training and minimizing the difference between the actual values and the predicted values. Backpropagation is necessary because the hidden units do not have target values to use. Therefore, these units must be trained based on errors from previous layers. The output layer has a target value, the actual value, which is used to compare with the computed value. The weights of the links are continuously updated as errors propagate backward between nodes. Training continues until the errors in the weights are acceptably small. The steps of the backpropagation algorithm are as follows:

Step 1: The normalized input data sample is fed, resulting in the corresponding output being calculated;

Step 2: The error between the outputs and the actual values is calculated;

Step 3: Link weights and membership functions are set;

Step 4: IF Error > Tolerance, Step 1 repeats, IF NOT, the process ends. These steps are mentioned in Table 3.3. as pseudo code (Khan, 2011).

Table 3.3. Backpropagation algorithm pseudocode

```
function BACK-PROP-LEARNING(examples, network) returns a neural network
  inputs: examples, a set of examples, each with input vector  $x$  and output vector  $y$ 
         network, a multilayer network with  $L$  layers, weights  $w_{i,j}$ , activation function  $g$ 
  local variables:  $\Delta$ , a vector of errors, indexed by network node
  repeat
    for each weight  $w_{i,j}$  in the network do
       $w_{i,j} \leftarrow$  a small random number
    end for
    for each example  $(x, y)$  in examples do
      /* Propagate the inputs forward to compute the outputs */
      for each node  $i$  in the input layer do
         $a_i \leftarrow x_i$ 
      end for
      for  $l = 2$  to  $L$  do
        for each node  $j$  in layer  $l$  do
           $in_j \leftarrow \sum_i w_{i,j} a_i$ 
           $a_j \leftarrow g(in_j)$ 
        end for
      end for
      /* Propagate deltas backward from the output layer to the input layer */
      for each node  $j$  in the output layer do
         $\Delta[j] \leftarrow g'(in_j) \times (y_j - a_j)$ 
      end for
      for  $l = L - 1$  to  $1$  do
        for each node  $i$  in layer  $l$  do
           $\Delta[i] \leftarrow g'(in_i) \times \sum_j w_{i,j} \Delta[j]$ 
        end for
      end for
      /* Update every weight in the network using deltas */
      for each weight  $w_{i,j}$  in the network do
         $w_{i,j} \leftarrow w_{i,j} + \alpha \times a_i \times \Delta[j]$ 
      end for
    end for
  until some stopping criterion is satisfied
  return network
```

3.4. Convolution Neural Network (CNN)

CNN is a DL algorithm that takes images as input and extracts features by filtering or transforming them with kernels (Chauhan, 2018). It is employed in a multitude of domains, including image recognition, natural language processing, and time series analysis. The CNN algorithm draws inspiration from the visual centers of animals (Hossain, 2019). The CNN consists of three principal layers: convolutional, pooling, and fully connected. The convolution process may be conceptualized as a neuron's response to stimuli originating from the stimulus field. (Şeker, 2017). The parameters of the layer exhibit a limited receptive field. However, it is comprised of a sequence of learnable filters that span the entirety of the input volume.

3.4.1. Convolution Layer

CNN is a convolutional neural network that employs a feature map to divide an input image into distinct features. It is a process of matrix multiplication, whereby the input image is shifted over using one or more pixels. Pixels are instrumental in defining the characteristics of an image. In its operational methodology, the input features are subjected to periodic scanning, and any discrepancies are incorporated through the addition of the input features and multiplication by the matrix elements within the receptive field (Wu, 2023). Pixels are related to adjacent and nearby pixels, and thus convolution allows for the preservation of the relationship between different parts of the image. Convolution preserves the relationship between pixels. The objective is to filter the image with a smaller pixel filter, thereby reducing the image's overall size (Hossain, 2019).

The conversion of an image with an sized filter allows for the same features to be learned throughout the image. In the context of CNNs, this implies the ability to recognize analogous patterns and features across the entire image through the convolution process (Chauhan, 2018). The process then continues with the sliding of the window, whereby the features are learned through feature maps (see Figure 3.7). Feature maps capture the local receptive field of the image and operate with shared weights and biases. The size of the unfilled output matrix is expressed as equation 3.17 (Chauhan, 2018).

$$N \times N * f \times f = N - f + 1 \quad (3.17)$$

$$Output(i, j) = \sum m \sum n Input(i + m, j + n) \times Filter(m, n) \quad (3.18)$$

Output(i, j) : is the element at the position (i, j) of the output matrix obtained as a result of the convolution process.

Input(i + m, j + n) : is the element of the input matrix at position (i + m, j + n).

Filter(m, n) : is the element at position (m, n) of the filter matrix

$\sum m$ and $\sum n$, perform addition operations on all elements of the matrices (Equation 3.18).

This formula demonstrates the process of passing the filter matrix for each pixel of the input matrix and multiplying the filter matrix and pixel values element by element.

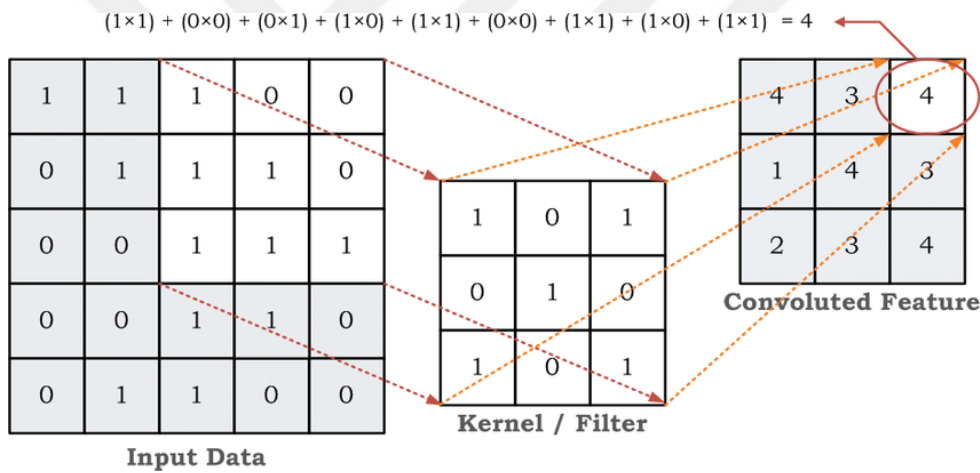


Figure 3.7.Feature map (Pratiwi, 2020)

3.4.2. Pooling layer

It is a standard practice to incorporate pooling layers following each convolution layer within a CNN model. Consequently, the pixel size is diminished. The incorporation of a pooling layer serves to reduce the number of parameters, thereby minimizing the computational complexity. Furthermore, it assists in addressing the issue of overfitting. A pooling size is selected (see Figure 3.8) to reduce the number of parameters by identifying the maximum, average, or total values within pixels. Pooling is frequently employed to diminish the pixel size and guarantee that the acquired features remain unaltered in the event of transformation, scaling, and

orientation shifts. The ReLU function is typically employed as an activation function in CNN layers. This is because the conversion of negative values to zero leaves positive values unchanged, thereby enabling the network to learn at a faster rate.

$$Output(i, j) = \max_{m,n} Input(i + m, j + n) \quad (3.19)$$

Output(i, j): is the element at position (i,j) of the output matrix obtained as a result of the retting process.

Input(i + m, j + n): is the element at position (i+m,j+n) of the input matrix subjected to retting.

max_{m,n}: finds the maximum value within a given region (Equation 3.19).

The pooling process is performed by subtracting the maximum pixel value in each region. The maximum value of a specific region is contingent upon the assumption that this region most accurately represents a particular feature. The pooling process facilitates the preservation of salient features while reducing the dimensionality of feature maps (Wu, 2023).

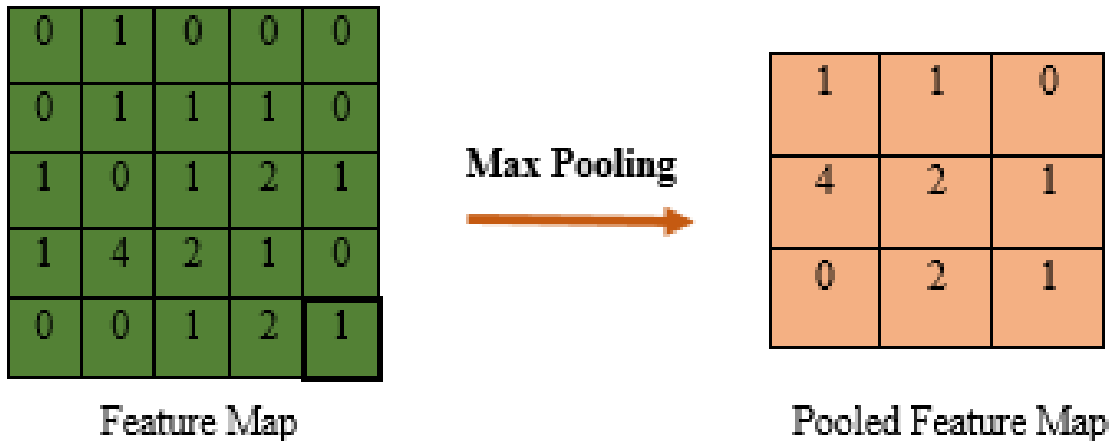


Figure 3.8.Maximum Pooling (Haq, 2023)

3.4.3. Fully connected layer

Subsequently, the full connection layer is typically introduced following the convolution and pooling layers. The fully connected layer is situated in the final portion of the hidden layer of the CNN. It serves only to transmit signals to other fully connected layers. The convolution and

pooling layers of a CNN are capable of performing feature extraction on the input data. The objective of the fully connected layer is to non-linearly combine the extracted features to obtain an output. The function of a fully connected layer in a convolutional neural network is that of a classifier. A fully connected network model is constructed for image classification. The objective of the layers is to establish a connection between the features extracted from the feature maps and the input data, thereby facilitating the generation of the final classification or regression results (Wu, 2023).

3.5. Recurrent Neural Networks (RNN)

The RNN was first proposed by Jeff Elman in 1988. RNNs are a type of ANN in which units are connected in a repeating pattern. Furthermore, it is a specific type of ANN designed to process sequential data (Pawar, 2018). The fundamental concept underlying RNN is the utilization of sequential data. The designation "recurrent" ascribed to the RNN architecture signifies that the same task is performed for each element in the series, contingent upon the preceding outputs (Şeker, 2017). This enables the utilisation of internal memory to facilitate the processing of the input sequence. The applications of RNN include handwriting recognition, text generation, stock market analysis, and speech recognition (Pawar, 2018). RNNs are capable of modeling long-range correlations within sequential data. This functionality enables the user to retain information from previous steps, thereby facilitating its utilization in subsequent steps. An RNN model typically comprises three principal layers: an input layer, one or more RNN cells, and an output layer. The input layer serves to provide the model with external data. The basic structure of an RNN cell contains an inner loop. The output layer is responsible for generating the model's predictions or classifications. There are numerous ways in which RNNs can be implemented, with the most prevalent approaches being: In a single-layer RNN, information flows in a unidirectional manner, typically progressing in a forward direction. In contrast, a bidirectional RNN facilitates the transfer of information in both forward and backward directions, incorporating both past and future data at each stage. LSTM is a type of traditional RNN that was developed as a solution to the memory problem. It is more effective at handling longer-term connections. In contrast, the GRU was developed as a solution to the memory problem, similar to the LSTM, and has a simpler structure (see Figure 3.9.) (Patel, 2018).

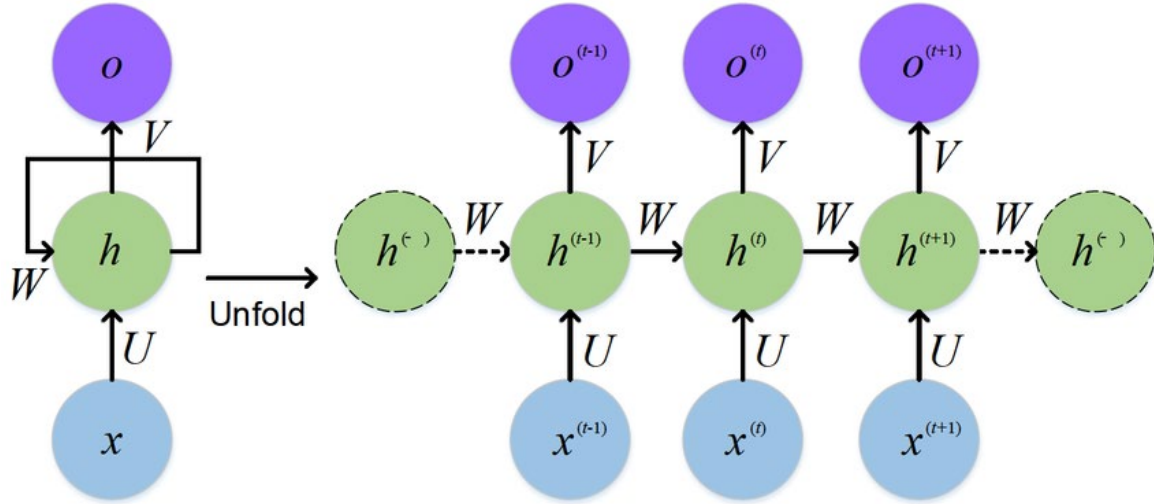


Figure 3.9.RNN Architecture Explained (Feng, 2017)

x_t is included in the model as input in step t .

$h_t, h_t = f(x_t * U + W * h_{t-1})$ is in the occult layer at that time. This is the memory of the network. Nonlinear activation functions such as tanh or ReLU are often used.

3.6. Long Short Term Memory Networks-LSTM

RNN has short-term memory. Therefore, if the network is long, it will be difficult to move information to the next network. When prediction is made in the network, RNN may exclude some of the information. LSTM provides a suitable solution to the problems of vanishing and exploding gradients of RNNs that occur when analyzing large data sets with RNN. Gradients are values used to update the weights of the neural network. Backpropagation travels from the output back to the input layer, propagating the error gradient. The vanishing gradient problem is that the gradient gets smaller over time as it propagates backward. If the gradient value is too small, it will not contribute much to learning. In RNN, layers that receive gradient updates stop learning. Those that contribute less to learning are usually the more advanced layers. Since these layers do not learn, RNNs can forget what they see in longer sequences. Therefore, they have short-term memory (Zhang, 2023).

As one progresses from higher-level to lower-level layers, trends tend to become less pronounced. This phenomenon is particularly evident at the input layers, which are situated at the forefront of the network. Consequently, the weights of the neurons remain constant at these

lower levels. Activation functions or batch normalization techniques may be proposed as potential solutions to the issue of gradients vanishing at a certain scale. However, the length of the sequence input, such as that of long time series data, may result in a reduction in the efficiency of the training process. As information is lost with each step that passes through the RNN, the network will eventually forget the initial inputs. It is therefore necessary to develop a long-term memory solution for RNN. The LSTM cell was developed in 1997 to address the vanishing gradient problem (Greff, 2016).

LSTM is an artificial neural network designed for the processing of sequential data, such as that represented by voice or text. The ability to handle long-term data is advantageous, as the output at a given time step is contingent upon the information obtained from previous time steps. LSTM is capable of retaining information for extended periods through the use of memory cells, input gates, output gates, and forgetting gates. These gates regulate the inflow and outflow of information into and out of memory cells, enabling the network to selectively store and retrieve information as needed. LSTM is commonly employed in applications such as language translation, speech recognition, and stock price prediction (Gülmez, 2023).

The LSTM architecture comprises an input layer, a forget layer, an output layer, a block input layer, a fixed failure loop, an output activation function, and peephole connections (Şeker, 2017). LSTMs possess intrinsic regulatory mechanisms, termed gates, which facilitate the control of information flow. LSTMs comprise four layers that interact with one another to generate an output (see Figure 3.10.). These layers include three logistic sigmoid gates and one tanh layer. The function of the gates is to restrict the flow of information through the cell. The subsequent cell is responsible for determining which portion of the information will be retained and which will be discarded. The LSTM architecture comprises a set of recursively linked subcontexts, collectively referred to as memory blocks. Each memory block comprises a memory cell, which stores the state; an input gate, which controls the input of new information; a forgetting gate, which controls the forgetting of old information; and an output gate, which controls the degree of change in the content. LSTM blocks represent a specific type of RNN structure, designed with the express purpose of addressing long-term dependencies, such as those encountered in time series. The gate mechanisms in LSTMs are employed to update the contents of memory cells and determine the quantity of information to be retained and that to be discarded. These gates are designated as "forget," "update," and "exit" gates, respectively,

and each is regulated by a sigmoid activation function. The structure of LSTMs enables them to accurately recall past information and learn long-term dependencies with precision (Chen & Ge, 2019).

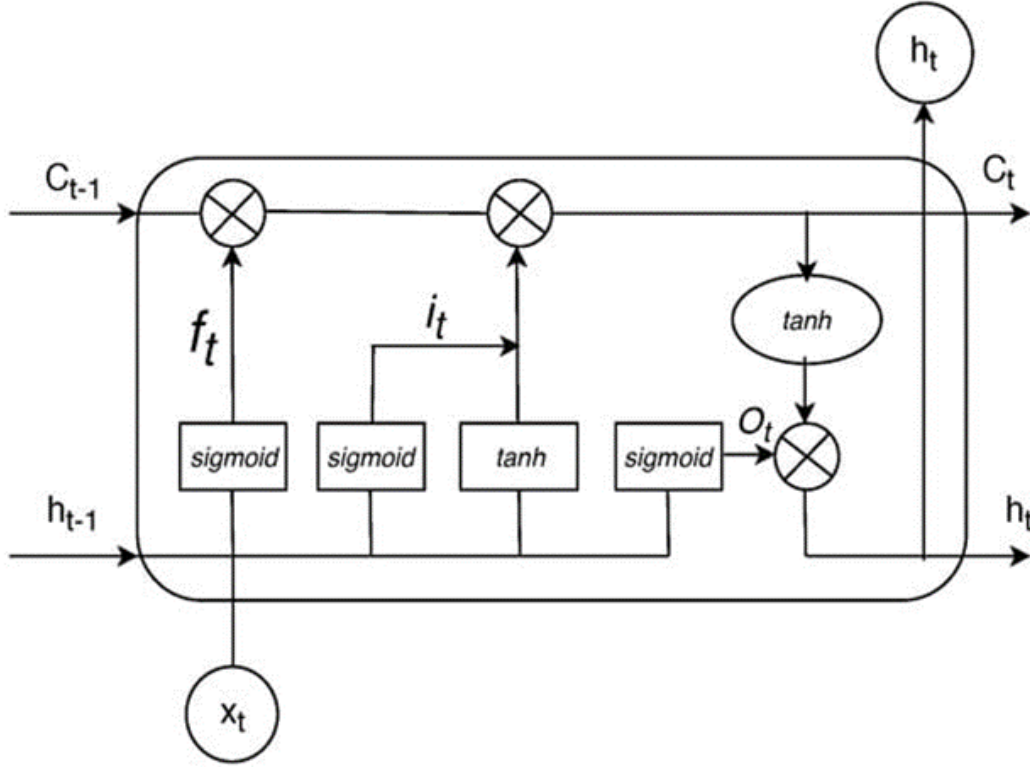


Figure 3.10. The LSTM architecture (Nyugen et al., 2019)

The term "forget gate" is used to describe the section between the forget gate and the cell, wherein information that is no longer useful is removed. Two inputs, x_t (input at a specific time) and $(h_t - 1)$ (previous cell output), feed the gate. The gate is then multiplied by weight matrices and the bias is added (Equation 3.20). The result obtained is processed in an activation function that gives a binary output. If the output is 0 for a particular cell state, the piece of information is forgotten and the information for output 1 is stored for reuse.

$$forgetgate(f_t) = \sigma(W_f * [h_{t-1}, x_t] + b_f) \quad (3.20)$$

Input Gate: The input gate contributes valuable data to the cell state. Information is organized using the sigmoid function, which filters the values to be remembered using the h_{t-1} and x_t inputs. This process is analogous to that of the forget gate. Subsequently, a vector comprising all potential values from h_{t-1} and x_t is generated using the tanh function, resulting in an output

range of -1 to +1. Ultimately, the values of the vector and the edited values are multiplied to yield useful information (Equation 3.21).

$$inputgate(i_t) = \sigma(W_i * [h_{t-1}, x_t] + b_i) \quad (3.21)$$

The output gate is responsible for the generation of the final output. The function of the output gate is to extract pertinent information from the current cell state and present it as the output. Initially, a vector is generated through the application of the tanh function to the cell. Subsequently, the information is organized using the sigmoid function. The filtering process is conducted by the values that are to be retained, utilising the h_{t-1} and x_t inputs (Equation 3.22). Ultimately, the values of the vector and the arranged values are multiplied and transmitted to the subsequent cell as both the output and the input.

$$outputgate(O_t) = \sigma(W_o * [h_t - 1, x_t] + b_o) \quad (3.22)$$

Cell state, sufficient information must be available for the calculation of the cell state. The cell state is pointwise multiplied by the forgetting vector. If this is multiplied by values close to 0, there is a possibility that the values in the cell state will decrease. In the next process, an expression is taken from the input gate. An addition is made that updates the cell state to new values that the neural network considers relevant. The result obtained shows the new cell state (Equation 3.23).

$$memorycell(c_t) = \tan \tan h(W_c * [h_t - 1, x_t] + b_c) \quad (3.23)$$

Hidden state: It is the output of the LSTM unit used to provide information or predictions to the next LSTM unit (Equation 3.24 and 3.25) (Gülmez, 2023-Zhang, 2023- Sunny, 2020).

$$hiddenstate(h_t) = O_t * \tan \tan h(U_t) \quad (3.24)$$

$$U_t = i_t * c_t + f_t * c_{t-1} \quad (3.25)$$

3.7. Bidirectional LSTM

Bidirectional RNN, developed by Schuster and Paliwal, is a type of RNN designed to train the network using past and future input data sequences. This structure is used especially in speech processing and time series analysis. BiLSTM is an improved version of this approach and aims to achieve the best learning performance by using all data. BiLSTM is based on the BiLSTM model, which includes previous and next input sequences. It has been able to overcome the vanishing gradient problem (Wöllmer, 2011). The BiLSTM eliminates the single-step truncation present in the LSTM and implements full error gradient computation. The error gradient approach facilitates the implementation of bidirectional LSTM (Althelaya, 2018). In addition, it can learn the optimal amount of contextual information relevant to the classification task (Wöllmer, 2011). An LSTM layer consists of repeatedly connected memory blocks, each containing one or more memory cells, and input, output, and forget gates, called multiplicative gates. The gates perform read, write, and reset operations (Althelaya, 2018). Each layer performs operations using the reverse time step direction (Wöllmer, 2011).

3.7.1. Bidirectional LSTM Structure

Bi-LSTM uses pre-processed data as input. It then passes back and forth through the LSTM neural network layer. Then it goes to the full connection layer and the prediction result is generated (Figure 3.11). The RNN that solves the long-term dependency of RNN and LSTM is Bi-LSTM. Since it combines LSTM in two different directions, it extracts forward and backward information data simultaneously (Althelaya, 2018).

LSTM posits that BLSTM can utilize temporal information in both directions simultaneously, thereby facilitating the removal of unused data (Jia, 2019). The rationale behind this architectural advantage over one-way training is that two-way training is employed for the classification of phonemes (Wöllmer, 2011).

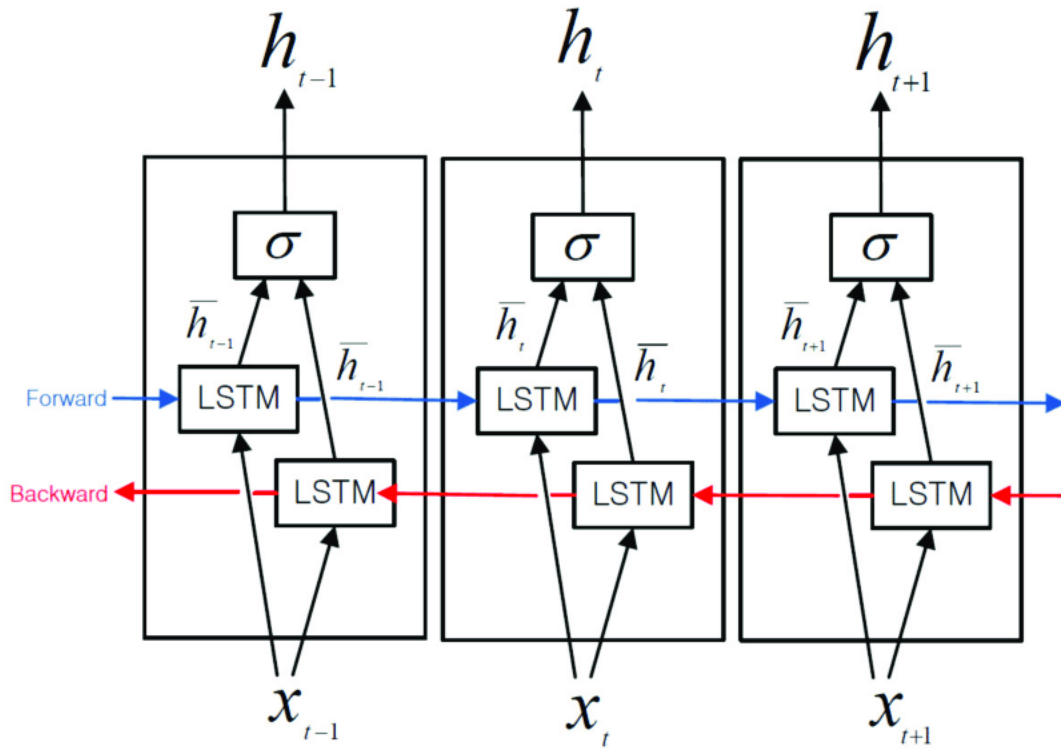


Figure 3.11.Bi-LSTM Structure (Park, 2020)

3.7.2. Bidirectional LSTM Principle

Sequential data is provided as the input to the BiLSTM from the input layer. In the forward layer, forward calculations are performed from time point 1 to time point t , and the output of the forward hidden layer is obtained and recorded at each time point. In the backward LSTM layer, a process of time reversal is employed to calculate and store the output of the backward hidden layer at each step. In the backward LSTM, the output at the current time step is calculated using the input information from each time step and the hidden states from subsequent time steps. The outputs of the forward LSTM and backward LSTM are then combined to create a combined output containing both previous and next information at each time step. The resulting outputs are typically routed to an appropriate output layer for classification, regression, or another task. The purpose of BiLSTM is to model long-term dependencies by creating a context that includes information from both previous and subsequent time steps in addition to the current input at each time step (Althelaya, 2018).

3.8. Hybrid Model CNN-LSTM

CNN is capable of focusing on the most pertinent elements within the field of view. LSTM possesses the capability of time series expansion. Furthermore, it is frequently employed in the domain of time series analysis (Lu, 2020). In the context of CNN-LSTM training, the initial step involves the utilisation of the requisite training data. A CNN is employed to extract features from images. A CNN generates a feature map through the process of feature extraction. Subsequently, the resulting feature map is transformed into a time series data set through the application of a CNN. The LSTM is employed for the processing of the aforementioned feature map. The CNN-LSTM model is trained with the training data set (Figure 3.12.).

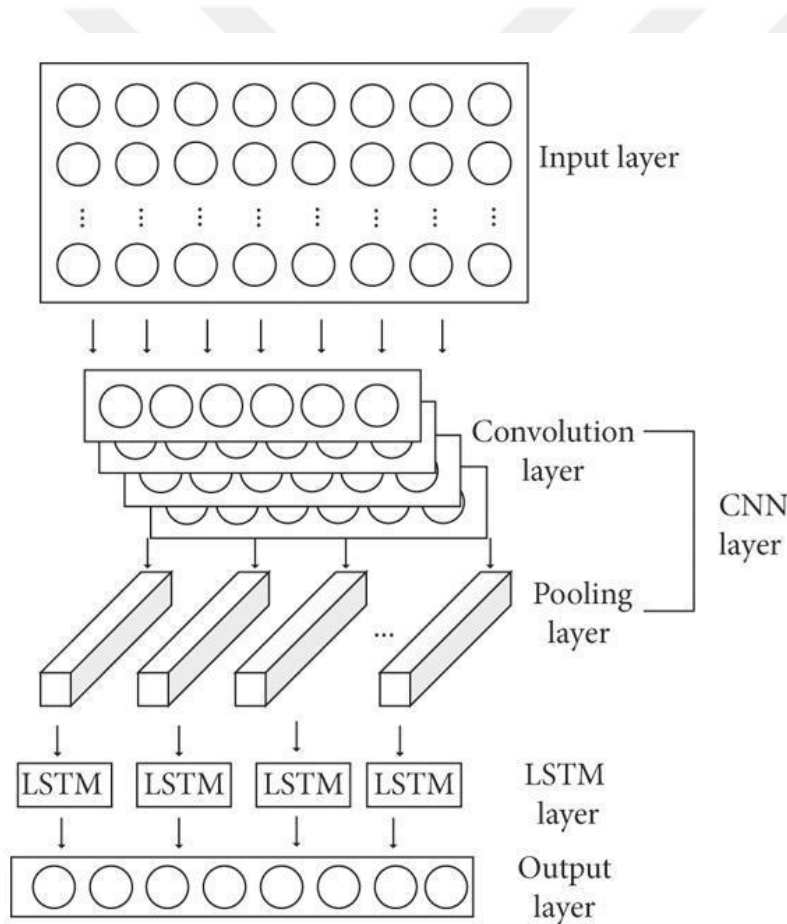


Figure 3.12.The CNN-LSTM model (Lu, 2020)

3.9. 10 Commonly Used Indicators in Technical Analysis

Technical indicators are employed to analyze past performance and price trends of stock prices. Technical indicators furnish insight into the direction and velocity of price movements, as well as the identification of trends and the realization of profits. Additionally, they facilitate the discernment of trend changes.

The objective of technical analysis is to examine past price and volume movements. Furthermore, it endeavors to anticipate the prospective trajectory of stock prices through the examination of indicator and formation analyses. Indicators are mathematical calculations employed to analyze price movements and trading volume. The fundamental tenet of technical analysis is that it is not a definitive science and instead deals with potential outcomes (Bahar, 2023).

Financial time series comprise price and volume data, which are collectively referred to as raw data. Nevertheless, the utilization of this data in direct conjunction with DL models will not yield sufficiently accurate outcomes. Consequently, this circumstance will impede the learning process of the models and diminish the accuracy of the predictions. The use of raw data may not provide an accurate reflection of market dynamics, resulting in predictions that are not sufficiently predictive. For this reason, technical indicators are employed to extract meaningful and descriptive features from raw data, thereby rendering it suitable for use in models. These indicators identify trends, momentum, volatility, and other pertinent information in price and volume data. The inherent random and unpredictable nature of financial data can present a challenge for models attempting to discern meaningful patterns. In other words, noise can obfuscate the data elements that convey meaningful information, thereby impeding the precision of predictive outcomes. For this reason, technical indicators serve to reduce the noise in the data, thereby facilitating the revelation of more meaningful and predictable patterns. Furthermore, the deployment of technical indicators is beneficial in that it enhances the precision of forecasting for the identification of specific patterns within financial time series. Subsequently, the data is subjected to normalization. Normalization is the process of scaling data within a specified range (typically between 0 and 1) to facilitate faster and more accurate learning by the model. The distribution of prices on the chart can be examined using technical analysis. Nevertheless, this analysis is insufficient for stock forecasting. Consequently, an

accurate stock forecast can be generated by selecting the most pertinent technical indicators for forecasting and employing DL methodologies.

In this study, the most commonly used fundamental indicators were selected based on the results obtained through a comprehensive review of the literature on stock prediction in technical analysis. These consist of the following technical indicators: cumulative log return, exponential moving average (EMA), SMA, MACD, Fibonacci's weighted moving average (WMA), Bollinger bands (BB), RSI, stochastic oscillator, momentum, ichimoku, and KDJ parameters.

3.9.1. Cumulative Log Returns

Cumulative log returns is a method used to calculate the total return on an investment. This is achieved by taking the sum of the logarithms of individual period returns. Log return refers to the logarithmic change between two consecutive closing prices and is calculated as follows (Equation 3.26):

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right) \quad (3.26)$$

r_t : log return at time t,

P_t : price at time t,

P_{t-1} : the price at time t-1.

Cumulative log return is the sum of log returns calculated over a certain period (Equation 3.27):

$$R_{cum} = \sum_{t=1}^T r_t \quad (3.27)$$

R_{cum} : Cumulative log return,

r_t : log return at time t,

T : total number of periods.

Since the cumulative return is obtained directly, it facilitates time series analysis and long-term return calculations. The abbreviation "CUMLOGRET_1" denotes the cumulative logarithmic return on investment over a specified time interval. An index value of "1" signifies that the specified cumulative return encompasses a single period.

3.9.2. Momentum indicator

The momentum indicator is utilized to ascertain the velocity of price fluctuations on a specific asset, thereby facilitating an assessment of the prevailing trend's strength. This indicator quantifies the extent and direction of price fluctuations over a specified time interval. The momentum indicator is usually calculated by the formula (Equation 3.28):

$$\text{Momentum} = P_t - P_{t-n} \quad (3.28)$$

P_t : price at time t,

P_{t-n} : price n periods ago,

N : the number of periods selected (usually used as 10 or 14 periods).

It is an oscillator that oscillates around a central line, usually 100 or 0. When this line is in the positive zone, it signals an upward trend, and when it is in the negative zone, it signals a downward trend. The "MOM_10" indicator represents the calculated value of the momentum indicator for a period of 10 periods (time frame). The most frequently cited momentum indicators are the Relative Strength Index and MACD.

3.9.3. Relative Strength Index (RSI)

This indicator referred to as RSI, is an indicator that measures the strength and speed of the stock. This indicator, also referred to as RSI, is a technical analysis tool that gauges the strength and velocity of a stock's movement. RSI indicator was first introduced by Welles Wilder in 1978 in his book, *New Concepts in Technical Trading Systems*. The indicator attempts to ascertain the future direction of prices by comparing the magnitude and direction of the stock's closing values. In other words, it determines overbought or oversold conditions by measuring the speed and changes in price movements of an asset. The formula utilized to calculate RSI is as follows (Equation 3.29 and 3.30) (Güneş, 2023; Anand, 2021).

$$RSI = 100 - \frac{100}{1+RS} \quad (3.29)$$

$$RS = \frac{(n)\text{average upward movement}}{(n)\text{average down movement}} \quad (3.30)$$

RSI is calculated based on a specific period, designated as n . The length of the RSI period (n) has a direct impact on the speed of price reaction. A longer period results in a slower reaction, while a shorter period leads to a faster reaction. RSI indicator can assume a value between 0 and 100. A decline in the RSI indicator to a value below 30 is indicative of an oversold market, whereas an increase above 70 signals an overbought market. A reading above 30 indicates a buy signal, while a reading below 70 indicates a sell signal. Furthermore, an increase in the RSI indicator above the 50 lines indicates a potential for a buy signal, whereas a decline below the 50 lines suggests a possibility of a sell signal (Güneş, 2023; Öztürk, 2016; Birgili, 2013).

The Stochastic RSI is a technical indicator often used to determine whether an asset is overbought or oversold. "14_14_3_3" in STOCHRSIk_14_14_3_3 and STOCHRSId_14_14_3_3. The first "14" indicates how far back the calculation is made. In this case, the Stochastic RSI values of the last 14 periods will be checked. In other words, it refers to the number of periods for which the RSI is calculated. The second "14" represents the number of periods used to calculate the %K line. It takes RSI values of the last 14 days and determines the highest and lowest points of these values. The third term, the %K line, refers to the calculation by taking the SMA of the Stochastic RSI values of the last 3 days. The last term, the %D line, is calculated by taking the SMA of the %K values of the last 3 days.

3.9.4. Moving Average Convergence Divergence (MACD)

MACD indicator was developed by Gerald Appel in 1979. The MACD is typically comprised of the examination of three EMA. MACD line is derived by subtracting the 26-day long-term EMA from the 12-day short-term EMA of stock prices. In the MACD indicator, a signal (trigger) line is generated from the nine-day moving average of the MACD line. A buy signal is generated when the MACD line crosses the signal line from bottom to top, whereas a sell signal is produced when it crosses the signal line downwards. The formula utilized to calculate the MACD indicator is as follows (Equation 3.31 and 3.32) (Güneş, 2023; Öztürk, 2016; Birgili, 2013).

MACD Line is obtained by subtracting the 26-day EMA from the 12-day EMA.

Signal Line is the 9-day EMA of the MACD line.

MACD Histogram shows the difference between the MACD line and the signal line.

$$MACD_{(s,l)} = EMA_{(s)} - EMA_{(l)} \quad (3.31)$$

$$\text{Signal Line}_{(t)} = EMA(MACD_{(s,l)}) \quad (3.32)$$

$$\text{MACD Histogram}_t = MACD_{(s,l)} - \text{Signal Line}_{(t)} \quad (3.33)$$

EMA, (Equation 3.34):

s: short-term period,

l: long-term period,

t: signal line period.

$$EMA_t = (P_t - EMA_{t-1}) * \frac{2}{n-1} + EMA_{t-1} \quad (3.34)$$

P_t: Price at time t,

EMA_{t-1}: EMA value of the previous day,

n: Number of days for EMA to be calculated.

The indicator parameters "MACD_8_21_9" are defined as follows: the number 8 represents the number of periods of the short-term Exponential Moving Average. In this instance, the short-term EMA is calculated using eight periods. The number 21 represents the number of periods of the long-term EMA. In this case, the long-term EMA is calculated using 21 periods. The number 9 represents the number of periods of the EMA of the MACD line. In this case, the EMA of the MACD line is calculated using 9 periods. MACD histogram represents the difference between the line of the MACD indicator and the signal line.

3.9.5. Bollinger Band (BB)

The Bollinger band is an indicator developed by John Bollinger in 1983. It is used to identify the direction of a stock's trend and to ascertain information regarding price volatility. Bollinger bands expand when market volatility is high and contract when volatility is low. The Bollinger indicator comprises three distinct bands: the middle band, the upper band, and the lower band. Middle Band is usually a 20-period SMA (Equation 3.35). The Upper Band is a specified

number of standard deviations above the middle band. The Lower Band is a specified number of standard deviations below the middle band (k is usually equal to 2).

$$\text{Middle Band} = SMA_{20} \quad (3.35)$$

$$\text{Upper Band} = SMA_{20} + (k * \text{Standard deviation}_{20}) \quad (3.36)$$

$$\text{Lower Band} = SMA_{20} - (k * \text{Standard deviation}_{20}) \quad (3.37)$$

$$SMA = \frac{\sum_{i=1}^{20} P_i}{20} \quad (3.38)$$

P_i : is the closing price of the i th day.

The formation of Bollinger bands is based on the calculation of a 20-day SMA. The upper and lower bands are calculated as the double standard deviation of the middle band (Equation 3.36 and 3.37). By the Bollinger indicator, an anticipated correction is indicated by stock prices exceeding the upper band, which is regarded as a sell signal. A decline in stock prices below the lower band is indicative of a perceived reduction in price, which is regarded as a potential buying signal (Güneş, 2023; Öztürk, 2016; Birgili, 2013).

The abbreviations "BBL," "BBM," "BBU," "BBB," and "BBP" are used to refer to the various components of the Bollinger Bands indicator. BBL is an abbreviation for "Bollinger Bands Lower." This represents the lower Bollinger Band. The lower limit of prices is determined by this indicator, which is often considered to be a support level. BBM is an abbreviation for "Bollinger Bands Middle." This illustrates the central Bollinger Band, which represents the mean price movement. The BBU is defined as follows: The abbreviation for "Bollinger Bands Upper." This represents the upper Bollinger band. The upper limit of prices is determined by this indicator, which is often considered to represent a resistance level. BBB The abbreviation for "Bollinger Bands Width." The term "BBB" is used to denote the width of Bollinger Bands. The width is defined as the distance between the upper and lower bands. The width is calculated by multiplying the standard deviation by a coefficient that indicates the number of times it is around the moving average. BBP: Abbreviation for "Bollinger Bands Percent B." This is an additional variant of the BB indicator, which is frequently employed to gauge momentum or to identify overbought or oversold conditions. This indicator provides insight into the proximity of the price to the upper or lower bands. To illustrate, the designation "BBL_5" in BBL_5_2.0 signifies that the moving average of BB is calculated over a five-period interval. The

designation "2.0" denotes the bandwidth coefficient. The bandwidth coefficient demonstrates the number of times the standard deviation is situated around the moving average. Thus, a width coefficient of 2.0 signifies that the upper and lower bands are, respectively, twice and half the standard deviation away from the moving average.

3.9.6. Stochastic Indicator

The stochastic indicator was first identified by George Lane in the 1950s. It is a momentum indicator that determines the position of the closing price of an asset in a specific time frame relative to the highest and lowest price range observed within the same time frame. The Stochastic Indicator is comprised of two distinct lines. The %K line is designated as the "fast stochastic line." The indicator demonstrates the position of the current price relative to the highest and lowest price range over a specified period. The %D line is referred to as the slow stochastic line. The indicator is calculated as the moving average of the %K line, which produces a more gradual and less volatile signal. The formula utilized in the stochastic indicator is as follows equations 3.39 and 3.40 (Güneş, 2023):

$$\%K = \frac{C - L_5}{H_5 - L_5} * 100 \quad (3.39)$$

$$\%D = \frac{H_3}{L_3} * 100 \quad (3.40)$$

C: Closing Price

H₅: Highest level in the last 5 days

L₅ : Lowest level in the last 5 days

The Stochastic Oscillator is a technical indicator that is frequently utilized to gauge the momentum of price fluctuations and discern overbought or oversold conditions. The parameters "STOCHk_14_3_3" and "STOCHd_14_3_3" are defined as follows: The number "14" indicates the number of periods used in calculating the Stochastic Oscillator. In this instance, it signifies the price fluctuations observed over the preceding 14 periods. The digit "3" represents the number of periods based on the EMA utilized to calculate the %K line (baseline). The %K line is an indicator that is calculated based on the closing of prices within a specified period. The remaining "3" refers to the number of periods based on EMA utilized to calculate the %D line, otherwise known as the signal line. The %D line is defined as EMA of the %K line.

3.9.7. Moving Averages

A moving average is a statistical measure that represents the average value of a given set of data points over a specified period. An increase in the moving average indicates an upward trend in prices, whereas a decline signifies a downward trend. Moving averages are among the most frequently utilized indicators in the field of technical analysis. The direction of general price movements can be discerned from moving averages drawn on a specific day (period). A rising long-term moving average is indicative of the robustness of the market. In the context of financial markets, the utilization of moving averages is typically contingent upon their respective maturity periods, as delineated below: The short-term moving average is calculated for five to ten days, while the short-to-medium term moving average is calculated for 21 to 22 days. The medium-term moving average is calculated for 50 days, and the long-term moving average is calculated for 100 to 200 days.

The interpretation of moving averages is a multifaceted process. The initial interpretation is to compare prices with moving averages. A break in the moving average from the bottom to the top is indicative of a "buy signal." The final price or prices observed on the last trading day are higher than the average price, which is interpreted as an indication of increased demand for the underlying asset. A breakdown in moving average from top to bottom is indicative of a sell signal. The second interpretation is a comparison of two moving averages. In this context, transactions are initiated when a short-term moving average crosses a long-term moving average. A crossover of the short-term moving average with the long-term moving average, occurring in an upward direction, is interpreted as a BUY signal. Conversely, a crossover occurring in a downward direction is perceived as a SELL signal. The third interpretation is concerned with the relationship between two different moving averages with the same maturity. To illustrate, if the 22-day EMA crosses the 22-day SMA in an upward direction, it could be interpreted as a BUY signal. Conversely, if it crosses the 22-day SMA in a downward direction, it could be interpreted as a SELL signal. This is because EMA exhibits a greater rate of change than SMA.

Furthermore, moving averages serve to establish support and resistance levels for prices. Short-term moving averages are more responsive than long-term moving averages, resulting in a

greater number of buy-sell signals. While multiple moving averages are calculated in technical analysis software, the most commonly utilized are SMA, WMA, and EMA (Li, 2023).

3.9.7.1. Simple Moving Average

It is defined as the arithmetic mean of prices. To illustrate, when calculating the 10-day SMA, the arithmetic mean is determined by adding the closing prices of the last 10 days and dividing by 10. The fundamental principle of the simple arithmetic mean is that each day is assigned equal weight. SMA is calculated using the following formula 3.41:

$$SMA = \frac{\sum_{i=1}^n P_t}{n} \quad (3.41)$$

P_t: Closing Price

n: day

In this context, "SMA_8" is used to denote an 8-period SMA. In technical analysis, the term "period" refers to the time span over which SMA is calculated. If SMA_8 is employed on the daily chart, SMA is calculated using 8 days.

3.9.7.2. Weighted Moving Average

In contrast to SMA, this method does not assign equal weight to each day. The calculation is based on the principle of assigning greater weight to the most recent days. If an investor deems the prices observed on the most recent days to be of greater consequence than those recorded on preceding days, WMA may be employed. This is expressed as follows equation 3.42 (Li, 2023).

$$WMA = \frac{((n * P_t) + ((n-1) * P_{t-1}) + ((n-2) * P_{t-2}) + \dots + ((n * (n-1) * P_{t-(t-1)}))}{(n + (n-1) + \dots + (n * (n-1)))} \quad (3.42)$$

P_t: Last day's closing

P_{t-1}: 1 day ago closing

n: number of days

3.9.7.3. Exponential Moving Average

Although EMA is primarily concerned with the most recent days, it does not disregard the initial days. The weights decrease exponentially following the price movements of the past. EMA can be calculated as follows equations 3.43 and 3.44 (Li, 2023).

$$EMA_t = EMA_{(t-1)} + \alpha(P_t - EMA_{(t-1)}) \quad (3.43)$$

$$EMA_t = \alpha P_t + (1 - \alpha) EMA_{(t-1)} \quad (3.44)$$

α coefficient is $\frac{2}{(n+1)}$,

EMA_t: moving average at time t,

EMA_(t-1): moving average at time t-1,

P_t: price at time t

n: is the total number of periods for which the EMA will be calculated.

In technical analysis, the term "EMA_8" is used to denote an 8-period EMA. In technical analysis, the term "period" refers to the period over which EMA is calculated. If EMA_8 is employed on a daily chart, the EMA is calculated using 8 days.

3.9.8. KDJ indicator

KDJ indicator is one of the most frequently utilized technical indicators in the context of stock investment decision-making. The momentum oscillator is designed to identify instances of convergence and divergence between the price of a stock and its moving averages. The indicator is composed of three distinct lines: the K line, the D line, and the J line (see Figure 3.13.). The %K Line is the fast line of the stochastic oscillator. The %D Line is the moving average of the %K line (Equations 3.45, 3.46, and 3.47). The %J Line is calculated as twice the difference between the %K and %D lines and reflects the difference between the %K line and the %D line in a more aggressive manner.

$$\%K = \frac{(C - L_n)}{(H_n - L_n)} * 100 \quad (3.45)$$

C: Current closing price,

L_n : n daily low price,

H_n : n daily high price.

$$\%D = SMA_3(\%K) \quad (3.46)$$

$$\%J = 3 * \%K - 2 * \%D \quad (3.47)$$

The stochastic oscillator is the source of both the K line and the D line, and the J line indicates the deviation of the D line from the K line. A crossing of the K line above or below the D line can be utilized as a trading signal when employing KDJ. Furthermore, KDJ indicator can be utilized to identify overbought and oversold positions when the K line is above 80 or below 20. It is recommended that the K and D lines be examined to identify any discernible patterns. A "golden fork" buy signal is indicated when the K line crosses above the J line while it is above the D line, indicating an uptrend. A sell signal designated a "dead fork," occurs when the K line is below the J line while the D line is below it, indicating a downtrend. It is imperative to supplement these signals with supplementary indicators, such as Average True Range (ATR) and Average Directional Index (ADX). These additional indicators can have subsequent effects that may result in erroneous signals (Anand, 2021).

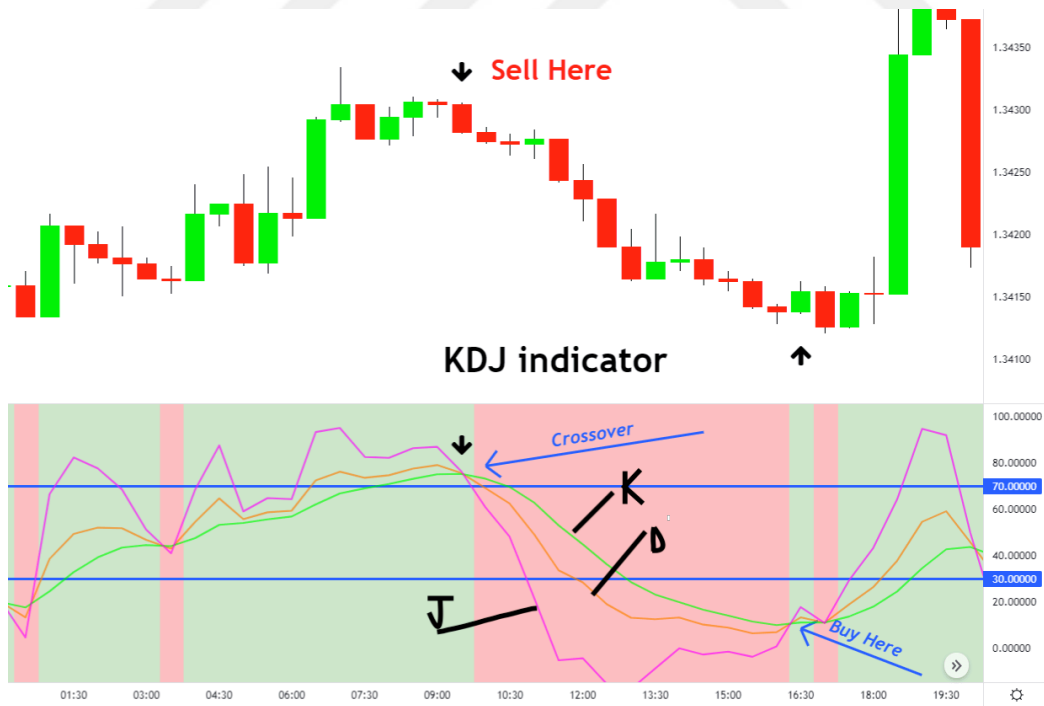


Figure 3.13. The KDJ indicator explained (Muhammad, 2022)

The indicator is comprised of the "K_9_3," "D_9_3," and "J_9_3" components, collectively known as the "KDJ" indicator.

- The K_9_3 component represents the %K component of the KDJ indicator. It is typically calculated as a 9-period stochastic oscillator. This component serves to ascertain the position of the current closing price with the highest and lowest prices observed over a specified time frame. The component is calculated using a "9" period Stochastic Oscillator and is frequently employed to gauge short-term momentum.
- D_9_3 represents the %D component of the KDJ indicator. The %D component is defined as a three-period moving average of the %K component. The %D component is typically regarded as a more refined version of the %K indicator, offering a more nuanced representation of market momentum. The "D" component is frequently employed to assess the intensity of the prevailing trend and the direction of momentum.
- J_9_3 represents the J component of the KDJ indicator. In most cases, J represents the difference between %K and %D. The J value is frequently employed to ascertain the points of intersection between %K and %D, as well as to identify conditions of overbought or oversold activity. The J value typically exhibits a more rapid change than the other two components, thereby providing a more sensitive signal.

3.9.9. Fibonacci Retracements

The use of Fibonacci retracements allows investors to identify potential support and resistance levels that could influence a company's price. These levels can then be converted into signals, which can in turn be used to identify the most probable points at which the stock is likely to encounter these levels. The Fibonacci Retracement method employs five specific points, namely 0%, 23.6%, 38.2%, 50%, and 61.8%, to ascertain support and resistance levels. The 100% level, representing the stock's highest price point, is of particular significance. The 23.6%, 38.2%, and 61.8% levels represent the stock's previous highs. The 50% level is of limited significance, as it is not regarded as a high point; rather, it represents the midpoint of the stock. To transform Fibonacci Retracements into a signal, traders and investors must a stock that is trading below the 23.6%, 38.2%, or 61.8% Fibonacci Retracement levels. Once this has been accomplished, the optimal course of action is to purchase the stock when it reaches the 50% Fibonacci Retracement level. To illustrate, if a stock is trading at \$50 and the 23.6% Fibonacci retracement level is \$46.15, the 38.2% Fibonacci retracement level is \$44.70, and the 61.8% Fibonacci retracement level is \$41.80. The stock's 50% Fibonacci retracement level is projected to reach \$48.50. In this instance, the trader will purchase the stock when it reaches

the 50% Fibonacci Retracement level, to reach the 61.8% Fibonacci Retracement level, and with the understanding that losses will be incurred at the 100% Fibonacci Retracement level. This strategy can be applied to any stock, but it is most effective for stocks that are undervalued by the market.

The expression FWMA₈ represents a calculation of an 8-period FWMA. This is a technical indicator that calculates a moving average of time series data over a specified period, typically days, weeks, or months.

3.9.10. Ichimoku

Ichimoku is a methodology that employs graph analysis. Ichimoku was initially developed by Japanese journalist Goichi Hosada in the 1930s, before the onset of World War II. Ichimoku began to be implemented in the West in the early 21st century with the publication of Elliott's (2007) book, *Ichimoku Charts*. It is a system that follows trends to provide support for decision-making processes. In comparison to other technical indicators, Ichimoku contains a greater number of bullish and bearish signals that are capable of reflecting market trends or price conditions with greater clarity. Ichimoku Kinko Hyo, or Ichimoku for short, is a five-line chart system that is calculated based on historical price data, specifically high and low prices. Four of these lines represent the average price, while the remaining line represents the delayed price. The five lines are the Tenkan-sen, Kijun-sen, Chikou-span, Senkou-Span A, and Senkou-Span B. The last two lines constitute the Kumo, or cloud. The fundamental principles of the Ichimoku system are derived through a methodology analogous to that employed in the calculation of moving averages. A distinctive attribute of this system is its incorporation of daily maximum and minimum prices into the analytical process. The Tenkan-Sen line, which is also referred to as the transformation line, constitutes a pivotal element within the Ichimoku system. The Tenkan-Sen line is calculated by taking the average of the highest and lowest prices observed over the previous nine periods, including the current period. The Tenkan-Sen line offers insight into short-term price movements, assisting traders in identifying potential trend reversals or entry points. It is regarded as the most rapid and informative indicator within the Ichimoku system, due to its swift responsiveness to price fluctuations. It is calculated using the following equation 3.48:

$$Tenkansen(T) = \frac{9 \text{ periods (highest price + lowest price)}}{2} \quad (3.48)$$

The Kijun-Sen line, also known as the basic line, complements the Tenkan-Sen line and represents the midpoint of prices over a longer range of twenty-six periods. Like the Tenkan-Sen line, the Kijun-Sen line helps investors assess the overall direction of the market. However, it reacts more slowly to price fluctuations, making it a useful indicator for identifying medium-term trends. When the Tenkan-Sen line crosses above the Kijun-Sen line, it may indicate an uptrend, while a cross below it may indicate a downtrend. It is calculated using the following equation 3.49:

$$Kijunsen(T) = \frac{26 \text{ periods (highest price + lowest price)}}{2} \quad (3.49)$$

The Chikou Spread, also referred to as the Lag Spread, offers a retrospective perspective on price movements by plotting the current period's closing price on a chart extending back twenty-six periods. This lag position enables traders to assess the historical strength of a signal generated by the Tenkan-Sen and Kijun-Sen lines. When the Chikou Span is in alignment with the prevailing prices and the Tenkan-Sen/Kijun-Sen lines, it serves to confirm the validity of a trend or reversal (as illustrated in equation 3.50).

$$Chikouspan = \text{Current closing price (deferred 26 periods)} \quad (3.50)$$

The Senkou Span A and Senkou Span B together form the Kumo, or cloud, which provides valuable insight into potential future market trends. The calculation of Senkou Span A entails the averaging of the Tenkan-Sen and Kijun-Sen lines over a period of twenty-six trading days. In contrast, Senkou Span B represents the average of the highest and lowest prices over a period of fifty-two trading days. The area between Senkou Span A and Senkou Span B forms the cloud, which can act as a support or resistance level. A thicker cloud indicates higher volatility and potentially stronger support/resistance levels, while a thinner cloud indicates lower volatility and weaker support/resistance areas (as illustrated in equations 3.51 and 3.52).

$$Senkouspan A = \left(\frac{Tenkansen + Kijunsen}{2} \right) 26 \text{ periods postponed} \quad (3.51)$$

$$SenkouspanB = \frac{52periods(highest+lowestprice)}{2} 26periods\text{postponed} \quad (3.52)$$

The Ichimoku system integrates these elements with the cloud, offering traders a comprehensive and visual methodology for analyzing market trends, potential entry and exit points, and overall market health. The Ichimoku system is a popular tool, particularly within the context of trend-following strategies. The lines of the Ichimoku system are calculated in a manner analogous to moving averages in technical analysis. However, in contrast to the typical use of closing prices in moving average calculations, the Ichimoku indicator employs daily highs and lows for its average calculations (Bilen et al., 2023).

The term "Ichimoku Kinko Hyo" is derived from the Japanese language, where it translates to "graph of balance" or "graph of equality." It is a technical analysis tool utilized in financial markets, frequently employed to discern trends and identify potential support and resistance levels. The Ichimoku Kinko Hyo system is comprised of several key elements, including (see Figure 3.14.):

- Tenkan-sen (ISA): Typically, it is calculated as a nine-period moving average, which assists in the identification of short-term trends.
- Kijun-sen (ISB): Typically, a 26-period moving average is employed to ascertain medium-term trends.
- Senkou Span A (ITS) is defined as follows: The sum of the Tenkan-sen and Kijun-sen is divided by two to create a cloud that indicates potential future support and resistance levels.
- Senkou Span B (IKS): Typically, it is calculated as a 52-period moving average, which generates a cloud that indicates potential future support and resistance levels.
- Chikou Span (ICS): A version of the current closing price is retroactively shifted back 26 periods on the chart and is employed to compare current price movements with past price movements.

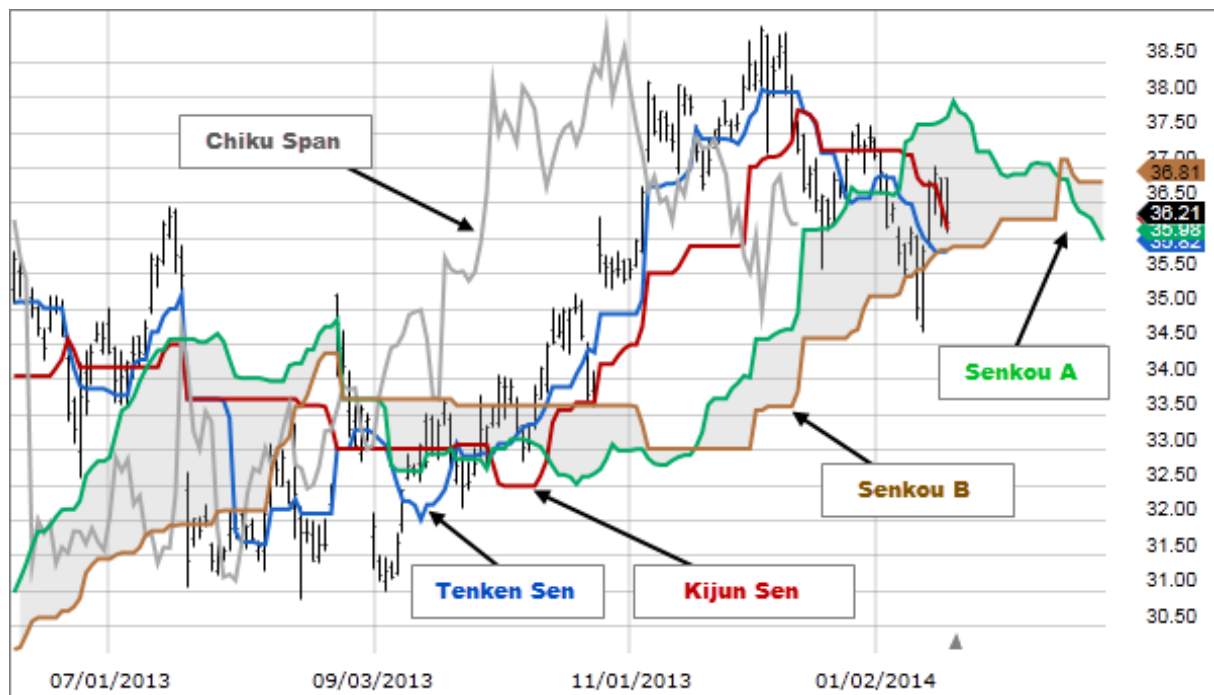


Figure 3.14.The Ichimoku Kinko Hyo system (Fidelity Investments. (n.d))

3.10. Data Preprocessing

Feature normalization represents a crucial phase in the preprocessing of data for the implementation of ML and DL models on multidimensional datasets. As the true distribution of each feature is unknown a priori, a variety of feature normalization methods are employed on the data prior to the commencement of the ML and DL training phase. The dataset to be utilized in numerous problems contains data that is inherently noisy, which represents one of the most challenging processes in machine learning, necessitating the elimination of noisy examples. The presence of disparate features at varying scales can render the learning process for the model challenging. For this reason, normalization is employed to align all features within a comparable scale range, thereby ensuring that the model assigns equal weight to each feature. The use of normalized data enables optimization algorithms to operate more rapidly and stably. The presentation of data at analogous scales can facilitate more precise predictive outcomes. The dataset is of considerable magnitude and complexity, and the presence of a greater number of features than examples gives rise to specific issues (Alexandropoulos, 2019). The use of large-scale data can result in numerical instability. Normalization serves to prevent the introduction of numerical errors by reducing the data to a smaller and more stable range. Normalization of data facilitates more precise adjustment of the model's weights. Furthermore,

in the context of NNs, the accurate adjustment of weights is of paramount importance with respect to the learning process of the model. Consequently, it facilitates the accurate updating of the network weights, thereby enhancing the efficiency of the learning process.

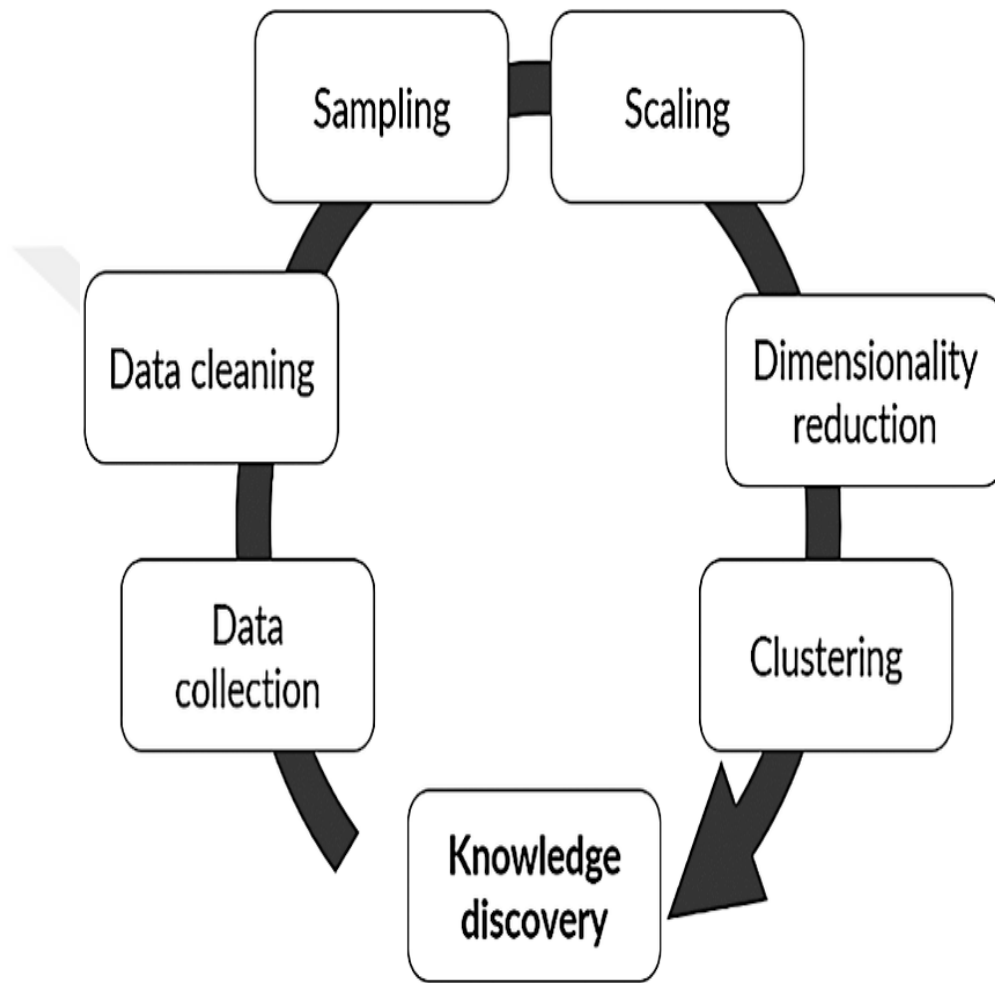


Figure 3.15. Data Flow Chart (Briceno-Mena, 2022)

In the various datasets examined, there are often considerable discrepancies between the values of different features, including maximum and minimum values. The most commonly employed methods for data normalization include Min-Max Scaling, Z-Score Normalization, Robust Scaling, Maximum Absolute Scaling, Unit Vector Normalization, Box-Cox Transformation, and Rank Transformation. In this study, the most commonly employed method was min-max normalization (Alexandropoulos, 2019).

3.10.1. Min-Max Scaler

It is essential to scale and standardize the data for it to be suitable for feeding to ML models. The Min-Max scaler scaling method is employed to normalize the data in advance of its introduction to the model. The Min-Max scaler is a method employed data normalization. The Min-Max scaler scaling method is employed to rescale the values of a given feature or variable to a specific range. The method entails the rescaling of data between specified values, with due consideration of the minimum and maximum values of a given feature or variable. In particular, it is employed for scaling within a range between 0 and 1. It compresses the data to a specific range without altering the distribution of the data. The calculation is as follows: (equation 3.53).

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (3.53)$$

X_{scaled} : scaled value

X : actual value

X_{min} : minimum value of the feature

X_{max} : maximum value of the feature

Min-Max Scaler is often used to make data comparable to each other and to make some ML algorithms perform better (Dey, 2018- Hossain, 2018).

3.10.2. Data Information

The BIST 100 is an index of Turkey's 100 largest companies, which serves to measure the performance of stocks traded on the Istanbul Stock Exchange (Borsa Istanbul). The BIST 100 index is a representation of the stocks of Turkey's most prominent and largest companies. Consequently, the stock market is one of the most frequently utilized indices in Istanbul. The index encompasses the securities of prominent corporations traded across a range of sectors. Moreover, the BIST 100 index serves as an indicator that reflects the economic situation and market conditions in Turkey (Yılmaz, 2016). In this study, the top five companies according to market capitalization value between 2020 and 2023 in BIST 100, as traded on Yahoo Finance, were utilized. Seventy percent of the data was utilized as training data, while the remaining thirty percent was employed as test data. The closing price value was employed as a predictive indicator for the subsequent day's data.

These indicators enhance the understanding of the direction, strength, and momentum of price movements. Neural network models perform better when fed with technical indicators, thus increasing the overall accuracy and predictive ability of the model. Furthermore, since these indicators visualize price data, better analyses can be conducted during the model's training and testing phases. In this study, technical indicators with a correlation value of 0.80 and above were selected for use in predicting closing prices in the dataset, which had become more informative through the inclusion of these indicators. Therefore, the use of technical indicators on normalized data aims to enhance both the accuracy and overall performance of the model.

3.10.3. Preparing Data for the Model

The data are divided into two distinct sets: a training set and a test set. The training set is used for the training and evaluation of the model, while the test set is reserved for evaluation purposes only. The data set, which includes features such as stock prices and technical indicators, is divided into two categories: input (X) and output (Y) variables. Input variables represent the values of technical indicators, including stock prices, trading volume, moving averages, RSI, and MACD. The output variables are used to predict the value that the model is attempting to predict, namely the closing price of the stock on the following day. In the context of time series data, future prices are predicted using past data. Consequently, both the input data (X) and the output data (y) are shifted over time. This ensures that the input data from a specific time period is matched with the output value from the subsequent time period. Subsequently, the data set is divided into two distinct sets: a training set and a test set. In this study, 70% of the data is utilized for training purposes, while the remaining 30% is employed for testing. The Closing Price value represents the dependent variable in this analysis. The training data set is employed to learn the model, whereas the test data set is utilized for the evaluation of the model's performance. The learning process of the models is hindered by the presence of data at disparate scales. Accordingly, normalization is employed to guarantee that the data is within the same scale range. The configuration of hyperparameters in the constructed model is a crucial aspect that influences the model's overall performance. In other words, hyperparameter setting involves attempting various combinations to optimize the performance of the model and identifying the combination that yields the most favorable results. This process is typically conducted using techniques such as "grid search" or "random search." In this study, the random search method was employed. In the process of hyperparameter setting, the number of layers

and the number of neurons in each layer are initially established. The number of layers and the number of neurons in each layer are then determined. It is typical to utilize neuron numbers such as 32, 64, or 128 in each hidden layer with two or more. This is because it enhances the intricacy and learning potential of the model. Subsequently, the activation functions to be utilized in each layer are selected. In this study, the most preferred activation functions, namely Relu, Sigmoid, and Tanh, were employed. Subsequently, the inputs are conveyed through the network's layers. Subsequently, the inputs are passed through the activation functions by multiplying them with the weights in each layer. The error value is calculated as the difference between the actual and estimated outputs. This is employed to update the error weights. The backpropagation algorithm is used to optimize the weights and deviations to minimize the error rate (Kumar, 2021). The learning rate is calculated to update the weights of the model. This is done to prevent the model from overfitting with an exceedingly high learning rate. The subsequent configuration specifies the number of instances that the model will be updated with in each iteration. The batch size divides the training of the model into discrete units, or "batches," rather than processing the entire data set at once, as is typical with smaller data sets. The most preferred batch sizes are 32, 64, and 128. The final parameter to be defined is that which determines the number of iterations of the training set that the model will undergo. The learning rate is a hyperparameter that determines the extent to which the gradient will be advanced by the gradient descent algorithm during the training of an ML model. The optimal learning rate is identified by evaluating the model performance with varying learning rates. Once the hyperparameters have been identified, the Adam algorithm and other optimization techniques are employed in the model training process. The Adam optimization algorithm is a widely utilized optimization method in the context of DL and ANN training. The Adam optimization algorithm updates the parameters of the model by the specified hyperparameters, thereby optimizing the learning process. The term "step size" is defined as the product of the gradient and the learning rate, as calculated by a gradient descent algorithm in the context of optimization. A smaller step size results in a reduced update of the gradient at each iteration. The models are created with the prepared data, and then trained and evaluated. The efficacy of the model is assessed by making predictions on the test data set that was used for training. Performance metrics, including MSE, MAE, and MAPE, were employed to assess the precision and efficacy of the output values. The computer utilized for model training was equipped with the following physical specifications: The processor is a: The computer utilized for training the

models is equipped with an Intel® Core™ i7-8700K CPU at 3.70GHz, 64GB of RAM, and a 64-bit operating system. The computer employs a 64-bit operating system.

3.11. Performance Evaluation

Performance metrics are employed to assess the precision and efficacy of output values in relation to input values within the model. The primary indicators utilized for the assessment of the main model in performance evaluation are as follows: The following performance metrics are employed: MSE , MAE , MAPE , R2 , RMSE , and absolute maximum error (AME). Additionally, the following metrics are employed: Mean Squared Logarithmic Error (MSLE), Mean Directional Accuracy (MDA), and Explained Variance Score (EVS) (Gülmez, 2023; Bakas, 2021).

MSE is a measure of the discrepancy between the predicted and actual values (Equation 3.54). The discrepancy between the predicted and actual values is squared and subsequently calculated by averaging all squared differences. The value is always positive and ranges from 0 to infinity. The MSE is employed to assess the precision of the model. A lower MSE value indicates a superior model.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.54)$$

MAE represents the discrepancy between the predicted and actual values (equation 3.55). The calculation is performed by taking the absolute value of the difference between the forecast and actual values and then averaging these values. Additionally, it is employed to assess the precision of the model. A lower MAE indicates a greater degree of accuracy in the model's predictions, as compared to the actual values (Chen, 2021).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3.55)$$

MAPE is a measure of the accuracy of a model expressed as a percentage (equation 3.56). The reason for its pervasive utilization in the assessment of forecasts is that the interpretation is straightforward and not contingent on the concept of relative error. MAPE is calculated by taking the absolute value of the difference between the predicted and actual values, dividing it

by the actual value, and taking the average of all these percentages. MAPE is employed to evaluate the accuracy of the model. A lower MAPE value indicates a superior model.

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (3.56)$$

RMSE is a statistical measure of the discrepancy between a forecast and the actual values. The discrepancy between the projected and actual values is squared. The mean of the sum of all squared differences is calculated by taking the square root of the resulting value (Equation 3.57). In contrast to MSE, the square root is employed so that its value can be readily interpreted. The value of this statistic ranges from 0 to infinity. A value closer to zero indicates superior performance.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (3.57)$$

R2 is a measure of the model's goodness of fit to the data. The R2 statistic is calculated by taking the sum of the squares of the differences between the forecast and actual values, dividing it by the sum of the squares of the differences between the actual values, and subtracting this value from 1 (Equations 3.58). This metric indicates the proportion of the total variation in the dependent variable that is explained by the independent variables. It is used to evaluate the accuracy of the model. A value of 1 indicates that the model perfectly explains the dependent variable, whereas a value close to 0 indicates that the model is not statistically significant.

$$R2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (3.58)$$

MSLE is calculated by taking the logarithm of the actual and predicted values, and then taking the average of the resulting squared errors. A superior error measurement is obtained by taking the logarithm of the estimated values. A lower MSLE value indicates a greater degree of proximity between the model's predictions and the true values Equation 3.59).

$$MSLE = \frac{1}{n} \sum_{i=1}^n (\log(1 + y_i) - \log(1 + \hat{y}_i))^2 \quad (3.59)$$

MDA is defined as the ratio of correctly predicted samples to the total number of samples. This metric is employed in the context of classification problems, to quantify the accuracy of the model's predictions. A higher MDA value indicates greater consistency between the model's predictions and the actual classes (Equation 3.60).

$$MDA = \frac{\text{number of correct guesses}}{\text{total number of samples}} \quad (3.60)$$

EVS is defined as the ratio between the total variance between actual and predicted values and the variance of the deviation from the mean of actual values. The value of the coefficient of determination ranges between 0 and 1. A value approaching 1 indicates that the model is capable of explaining a significant proportion of the variance in the target variable. Conversely, a score approaching zero indicates that the model is unable to account for the observed variance. An elevated EVS value indicates an improved capacity of the model to elucidate the variance of the target variable (Equation 3.61).

$$EVS = 1 - \frac{\text{Var}(y - \hat{y})}{\text{Var}(y)} \quad (3.61)$$

AME is a metric that quantifies the largest discrepancy between the model's predictions and the actual values. It is used to assess the model's worst-case prediction error, highlighting the maximum deviation between the predicted and true values (Equation 3.62).

$$AME = \max (|y_i - \hat{y}_i|) \quad (3.62)$$

y_i , actual value,

$\hat{y}_i$, predicted value.

A low AME indicates that the model performs well even in the worst-case scenario, while a high AME indicates that the model may make large errors in some predictions.

3.12. Programming Languages and Libraries Used in Deep Learning

A plethora of open-source software is available for the training and testing of DL models. The most commonly utilized programming languages in the present era are Python, C++, and Julia. Python is a highly prevalent programming language in recent years, utilized for a multitude of purposes, including coding, data analysis, and machine learning. It boasts a comprehensive library of resources. Furthermore, it is a flexible language with high code readability. Furthermore, the most prominent libraries for DL are TensorFlow, Keras, PyTorch, Pandas, and others. It is typically developed using the Python programming language.

TensorFlow is a highly popular and widely utilized DL library developed by Google. It is employed in the construction of both low-level and high-level artificial neural networks, and it facilitates training operations.

Keras is a DL library developed by Google. It is a high-level DL library. It is frequently utilized as the upper layer of TensorFlow. It provides an accessible and adaptable user interface.

PyTorch is a DL library developed by Facebook. It is a DL library developed by Facebook. Its foundation is the dynamic computation graph model. It is particularly prevalent in research and prototyping contexts.

Pandas is a library that provides high-performance, user-friendly data structures and data analysis tools for the Python programming language. Pandas is employed in a multitude of data science and analysis tasks, including data manipulation, data cleaning, data transformation, data visualization, and data analysis (Erden, 2021).

The three-year data set was taken into consideration in this study. A Python program was employed for stock price prediction. In comparison to other available software packages, this program is notable for its ease of use and flexibility.

4. RESULTS AND DISCUSSION

The global economy and the financial sector have exhibited considerable volatility over time. Such fluctuations also have a direct impact on stock values (Bilen, 2023). A multitude of factors influence the performance of stocks. The topic of stock price prediction is a popular one among researchers. It is not feasible to predict stock prices due to their inherently non-linear nature. Nevertheless, a considerable body of research has been conducted on this topic, and the findings have been corroborated to a certain extent. The majority of factors examined in the studies referenced in the literature were designed to be more comprehensive and to identify potential relationships that had not been previously observed. In the field of research, a multitude of technical indicators have been employed to predict future price movements through the analysis of historical price data and transaction volumes. In this study, the most commonly utilized technical indicators were selected through an examination of the extant research literature. The identified technical indicators are calculated based on the closing price of the pertinent stock and prepared for use in training the models. In addition to statistical methods for stock price prediction, computational intelligence methods, which have been developed in conjunction with the increasing performance of computers, are also employed in the literature.

The data set was selected by the market capitalization value of the stock traded on Bist100. The term "market cap" is used to describe the total market value of a company or asset. Figure 4.1. illustrates the market capitalization values of the five companies with the highest market capitalization values of the stocks traded on BIST-100. These companies are Koç Holding, Turkish Airlines, Ford Otosan, Türkiye Petrol Rafinerileri, and İş Bankası. The data set utilized for stock estimation in the examined studies is restricted to a specific temporal period.

The stocks selected for the research were considered, and the objective was to estimate the closing value and compare the performance of the models. The model data was designed with a 70% training and 30% test set, and MSE was employed as the loss function.

Table 4.1. The market capitalization values of stock prices

Stocks	MarketCap (1/1000000000 TL)	Industry
KCHOL.IS	443.3900	Conglomerates
THYAO.IS	389.5000	Airlines
FROTO.IS	344.5900	Auto Manufacturers
TUPRS.IS	324.8600	Oil & Gas Refining & Marketing
ISCTR.IS	304.3800	Banks - Regional

The empirical data regarding the stock market indexes are presented in tabular form. The data set comprises a total of 1,002 days, with the commencement and conclusion dates for each selected stock being 02/02/2020-29/12/2023. The data were then employed in the calculation of technical indicators for training purposes. Following the calculation of the technical indicator, 899 data points remained after the removal of any empty lines. Of the total data set, 629 were allocated for training purposes, while the remaining 270 were designated for testing. The prediction process was conducted using selected DL models.

The mean values of the stock data are 48.6176 (KCHOL.IS), 69.5491 (THYAO.IS), 325.6616 (FROTO.IS), 44.2027 (TUPRS.IS), and 6.9869 (ISCTR.IS), in the order presented in Table 4.2. The calculation was performed as follows: To ascertain the degree of dispersion or clustering of the values in the data set, the standard deviation values of the stocks were calculated. The resulting values are 40.5019 (KCHOL.IS), 78.8417 (THYAO.IS), and 255.0122 (FROTO.IS), as presented in the table. The standard deviation values were calculated as 41.5899 (TUPRS.IS) and 6.3509 (ISCTR.IS), respectively.

A box plot is a graphical representation of a data set, comprising a five-number summary: minimum, first quartile, median, third quartile, and maximum. The objective of this chart is to provide a visual representation of the data set and facilitate an in-depth examination of its statistical characteristics. A box plot enables the user to readily discern the quartiles, median, dispersion, and outliers of a given data set. The lowest recorded values for daily published stock prices are 12.9900 (KCHOL.IS), 7.7400 (THYAO.IS), 43.6000 (FROTO.IS), 9.5857 (TUPRS.IS), and 2.0475 (ISCTR.IS). The highest recorded values are 158 (KCHOL.IS),

268.5000 (THYAO.IS), 965 (FROTO.IS), 169.7000 (TUPRS.IS), and 25.9400 (ISCTR.IS). The initial quarter values of the stocks in the aforementioned table are as follows: 19.2500 (KCHOL.IS), 12.6200 (THYAO.IS), 126.5250 (FROTO.IS), 13.9161 (TUPRS.IS), and 2.4165 (ISCTR.IS). The median quarter values are 31.3000 (KCHOL.IS), 23.1400 (THYAO.IS), 246.3000 (FROTO.IS), 24.1143 (TUPRS.IS), and 3.5056 (ISCTR). The third quarter values are 72.9000 (KCHOL.IS), 127.6250 (THYAO.IS), 482.7750 (FROTO.IS), 67.2964 (TUPRS.IS), and 10.7075 (ISCTR.IS). They are presented in the form of a table.

Table 4.2. The stock is related to data information

Index	KCHOL.IS	THYAO.IS	FROTO.IS	TUPRS.IS	ISCTR.IS
Begin Date	2/2/2020	2/2/2020	2/2/2020	2/2/2020	2/2/2020
End Date	12/29/2023	12/29/2023	12/29/2023	12/29/2023	12/29/2023
Count	1002	1002	1002	1002	1002
Mean	48,6176	69,5491	325,6616	44,2027	6,9869
Std	40,5019	78,8417	255,0122	41,5899	6,3509
Min	12,9900	7,7400	43,6000	9,5857	2,0475
25%	19,2500	12,6200	126,5250	13,9161	2,4165
50%	31,3000	23,1400	246,3000	24,1143	3,5056
75%	72,9000	127,6250	482,7750	67,2964	10,7075
Max	158	268,5000	965	169,7000	25,9400

The closing price values for five stocks are presented in Figure for the years 2020-2023. As evidenced in Table 4.2, the data set in question does not exhibit a stationary structure. The presence of a significant upward or downward trend in the time series, as indicated by the trend analysis, is indicative of a lack of stationarity. Accordingly, stationarity in a time series signifies the presence of statistical properties that remain constant over time.

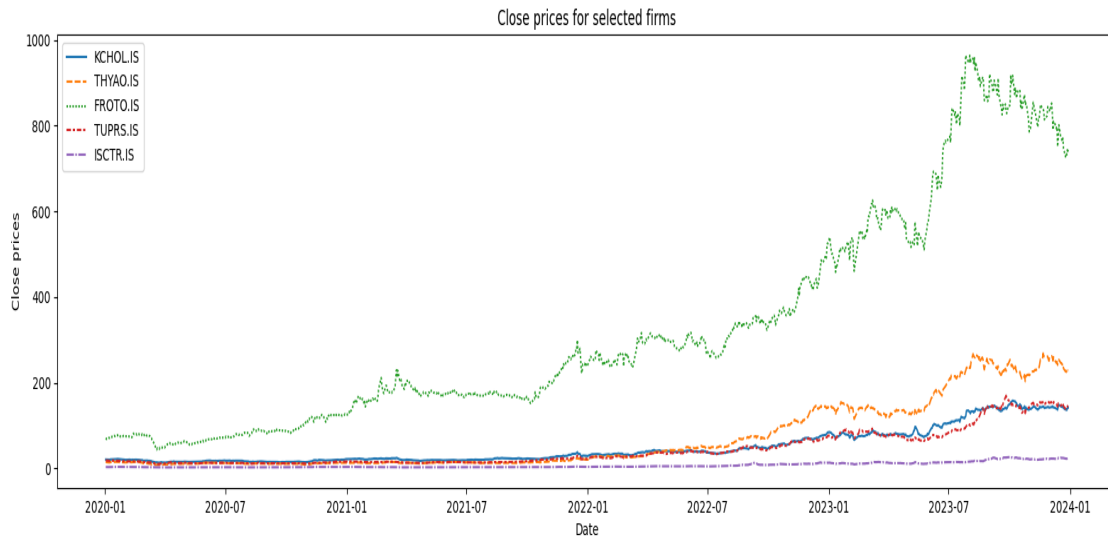


Figure 4.1. Close price for selected firms

The utilization of technical indicators is a crucial aspect of technical analysis. Technical indicators assist in the determination of the direction of the trend, the strength of the trend, support and resistance levels, and potential trend changes (Çetinyokuş, 2002). Indicators furnish investors with data regarding the present status of the market and potential future trends. As previously stated in Section 3, it is of paramount importance to select the most appropriate indicators for stock prediction. The use of technical indicators facilitates the prediction of future price movements. However, given the inherent volatility of financial markets, it is imperative to recognize that past performance does not necessarily guarantee future outcomes (Bahar, 2023). Technical indicators can forecast the direction in which a stock is likely to move and measuring its strength (Çetinyokuş, 2002).

The 1002-day data on the stocks in question was subjected to analysis, employing the aforementioned technical indicators. This analysis aimed to ascertain the direction and strength of these stocks. The results of this analysis assist investors in making more informed decisions. While technical indicators can be effective tools for forecasting price movements in financial markets, they may not be a sufficient standalone approach. It is important to note that there is no guarantee that a strategy that has demonstrated success in the past will continue to do so in the future. Such signals may be the result of sudden changes in price movements or market fluctuations, which can lead to erroneous conclusions. Moreover, technical indicators do not incorporate fundamental factors such as macroeconomic events, company news, and political developments. Accordingly, the integration of technical indicators within DL algorithms

guarantees that the resulting forecasts are in close alignment with the actual values, thereby facilitating the acquisition of more precise outcomes.

Table 4.3. Technical Indicators

Technical Indicators	
Cumulative Log Return	CUMLOGRET_1
Exponential Moving Average	EMA_8
Simple Moving Average	SMA_8
MACD Moving Average Convergence Divergence	MACD_8_21_9
	MACDh_8_21_9
	MACDs_8_21_9
FWMA Fibonacci's Weighted Moving Average	FWMA_8
Bollinger Bands: bbands	BBL_5_2.0
	BBM_5_2.0
	BBU_5_2.0
	BBB_5_2.0
	BBP_5_2.0
Stochastic RSI: stochrsi	STOCHRSIk_14_14_3_3
	STOCHRSId_14_14_3_3
Stochastic Oscillator: stoch	STOCHk_14_3_3
	STOCHd_14_3_3
Momentum: mom	MOM_10
Ichimoku Kinkō Hyō: ichimoku	ISA_9
	ISB_26
	ITS_9
	IKS_26
	ICS_26
KDJ	K_9_3
	D_9_3
	J_9_3

The most fundamental technical indicators were selected from the Table 4.3 by the data obtained from the literature review. The application of technical indicators facilitates the extraction of more meaningful information by reducing the influence of noise in the normalized data. The correlation values between the closing prices and the various subcategories of the technical indicators for stocks were calculated. Indicators with a correlation value of 0.80 or above were selected for use in closing price estimation for technical indicators. Consequently, the application of technical indicators facilitated the generation of more meaningful and predictable patterns by attenuating the influence of extraneous data.

In Figure 4.2, the technical indicators have been determined according to the threshold value of the correlation of each technical indicator considered with the closing price for KCHOL.IS are FWMA_8, BBL_5_2.0, BBM_5_2.0, BBU_5_2.0, EMA_8, SMA_8, ISA_9, ISB_26, ITS_9, IKS_26, and ICS_26, and CUMLOGRET_1. These terms typically pertain to particular indicators or indicators employed in technical analysis. These are typically special formulas or encodings based on specific calculation methods and parameters. The character structure of the stock is directly proportional to the abundance of available data (Çetinyokuş, 2002).

Figure 4.2. illustrates the concentration of the closing price of KCHOL.IS within specific ranges. A histogram is a graphical representation that is utilized to illustrate the distribution of data. The histogram is an invaluable tool for swiftly grasping essential characteristics of a data set, including density, dispersion, and central tendency. Furthermore, it enables the visualization of data distribution across specific ranges. As illustrated in Figure 4.3., the distribution is asymmetrical, with a greater concentration of closing price values in the 10-20 range and a lower concentration in the 90-120 range.

Features Correlating with Close Price

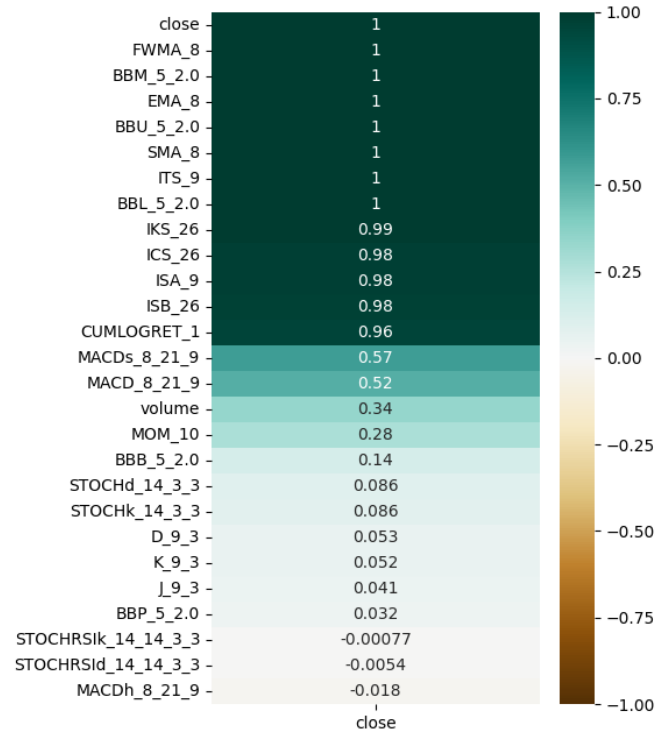


Figure 4.2. Feature Selection Criteria for KCHOL.IS

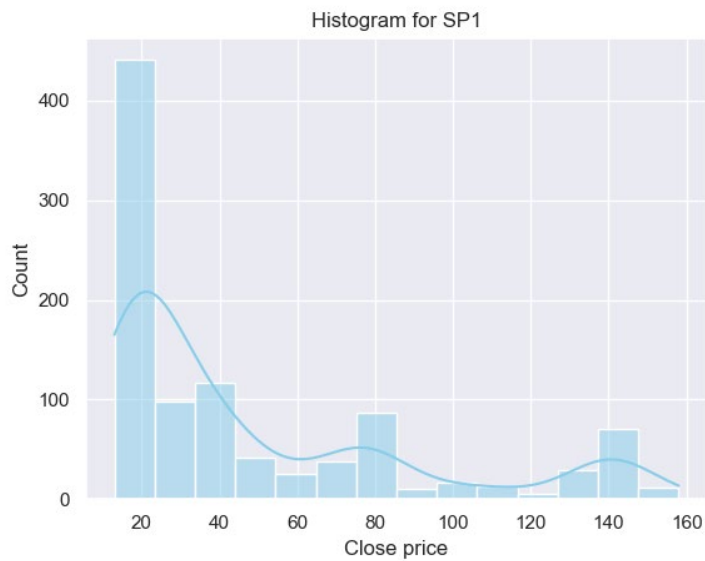


Figure 4.3. Histogram for KCHOL.IS

In Figure 4.4, the technical indicators are determined according to the threshold value of the correlation of each technical indicator considered with the closing price for THYAO.IS. The identified technical indicators are FWMA_8, BBL_5_2.0, BBM_5_2.0, BBU_5_2.0, EMA_8, SMA_8, ISA_9, ISB_26, ITS_9, IKS_26, and ICS_26. Additionally, CUMLOGRET_1 is identified as a potential indicator. The Figure 4.5 illustrates the concentration of the closing price of THYAO.IS within specific ranges. Figure 4.5 demonstrates that the distribution is asymmetrical, with a greater concentration of closing prices in the 0-50 range and a notable decline in the 120-150 range.

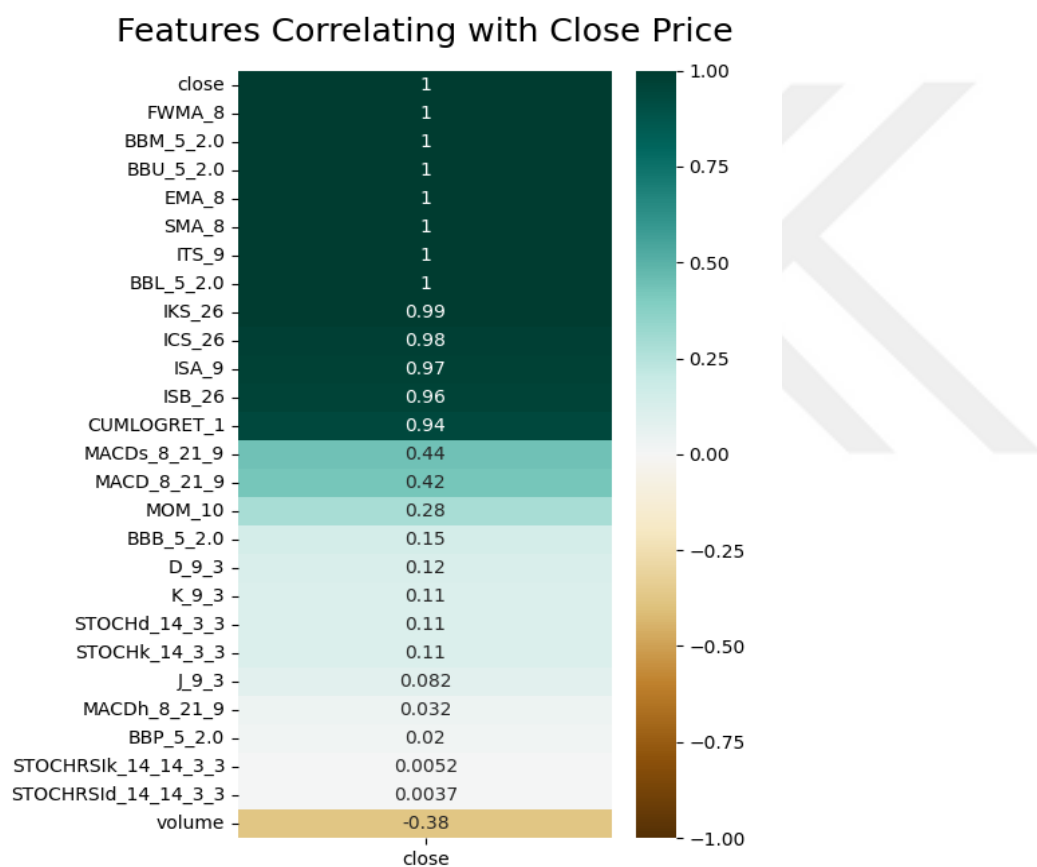


Figure 4.4. Feature Selection Criteria for THYAO.IS

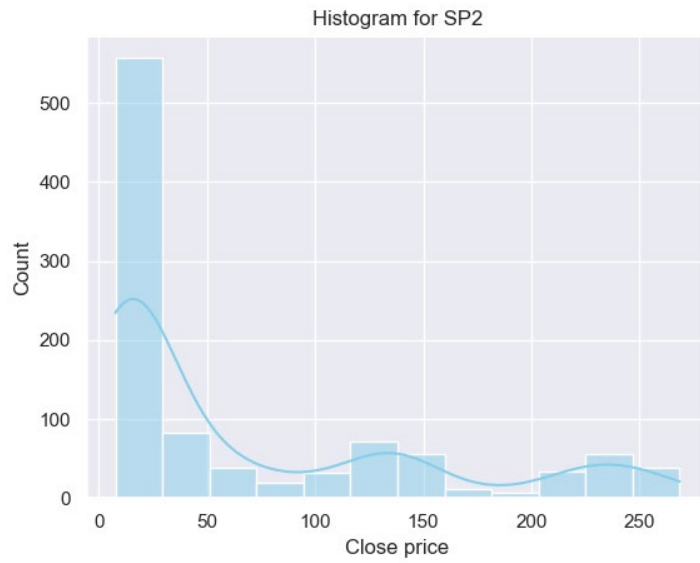


Figure 4.5. Histogram for THYAO.IS

In the Figure 4.6, technical indicators are determined according to the threshold value of the correlation of each technical indicator considered with the closing price for FROTO.IS. The indicators are FWMA_8, BBL_5_2.0, BBM_5_2.0, BBU_5_2.0, EMA_8, SMA_8, ISA_9, ISB_26, ITS_9, IKS_26, ICS_26, and CUMLOGRET_1. The figure illustrates the concentration of the closing price of FROTO.IS within specific ranges. Figure 4.7 illustrates that the distribution is not symmetrical, with a greater concentration of closing prices in the 0-200 range and a lower concentration in the 600-800 range.

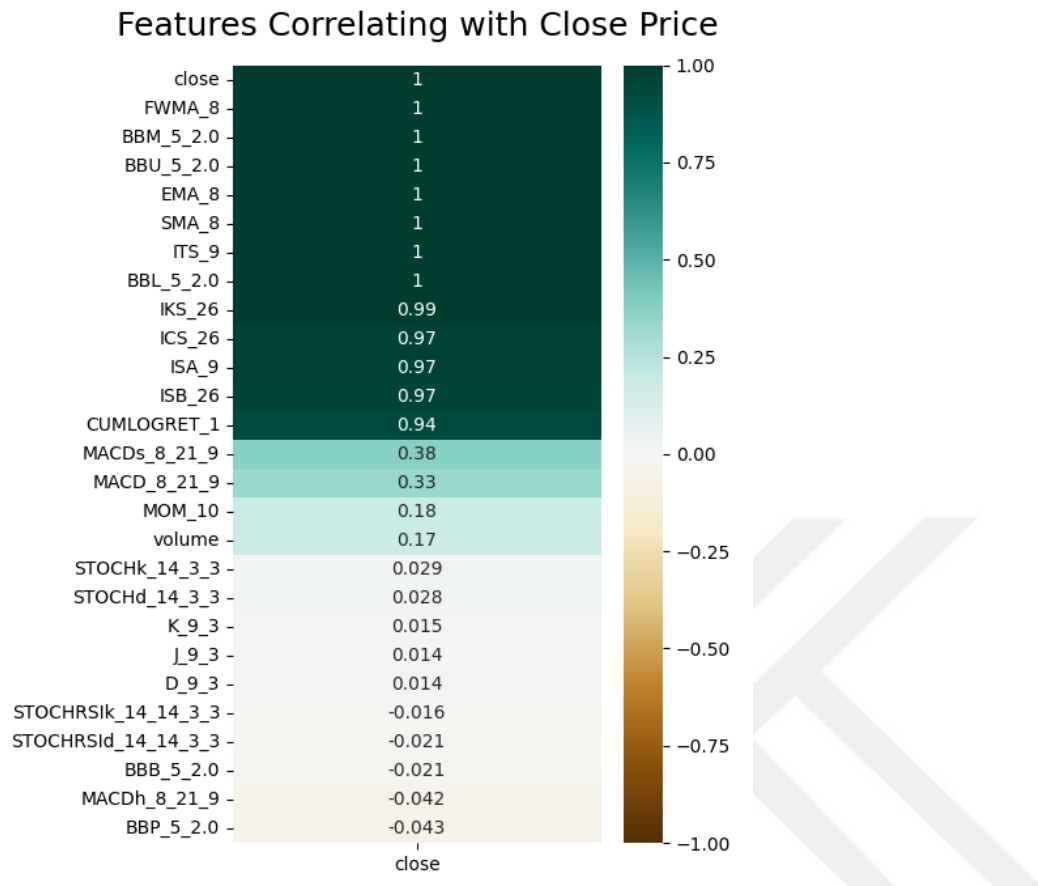


Figure 4.6. Feature Selection Criteria for FROTO.IS

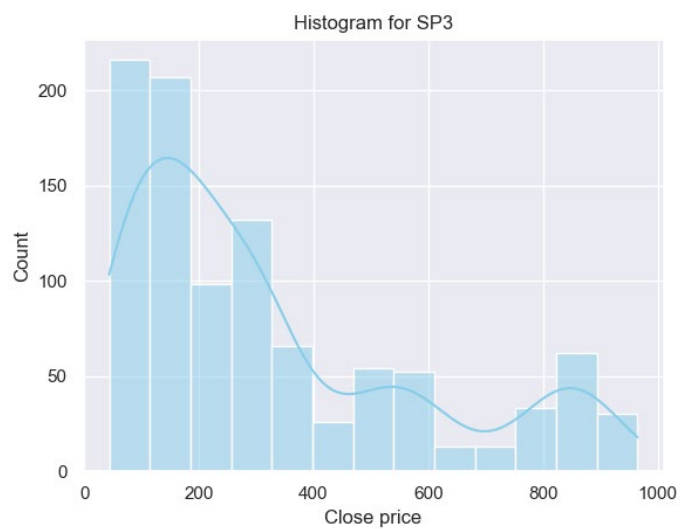


Figure 4.7. Histogram for FROTO.IS

In Figure 4.8, the technical indicators are determined according to the threshold value of the correlation of each technical indicator considered with the closing price for TUPRS.IS. The identified technical indicators are FWMA_8, BBL_5_2.0, BBM_5_2.0, BBU_5_2.0, EMA_8, SMA_8, ISA_9, ISB_26, ITS_9, IKS_26, and ICS_26. Additionally, CUMLOGRET_1 is identified as a technical indicator. Figure 4.9 illustrates the concentration of the closing price of TUPRS.IS within specific ranges. Figure 4.9 demonstrates that the distribution is not symmetrical, with a greater concentration of closing prices in the 0-20 range and a lower concentration in the 100-140 range.

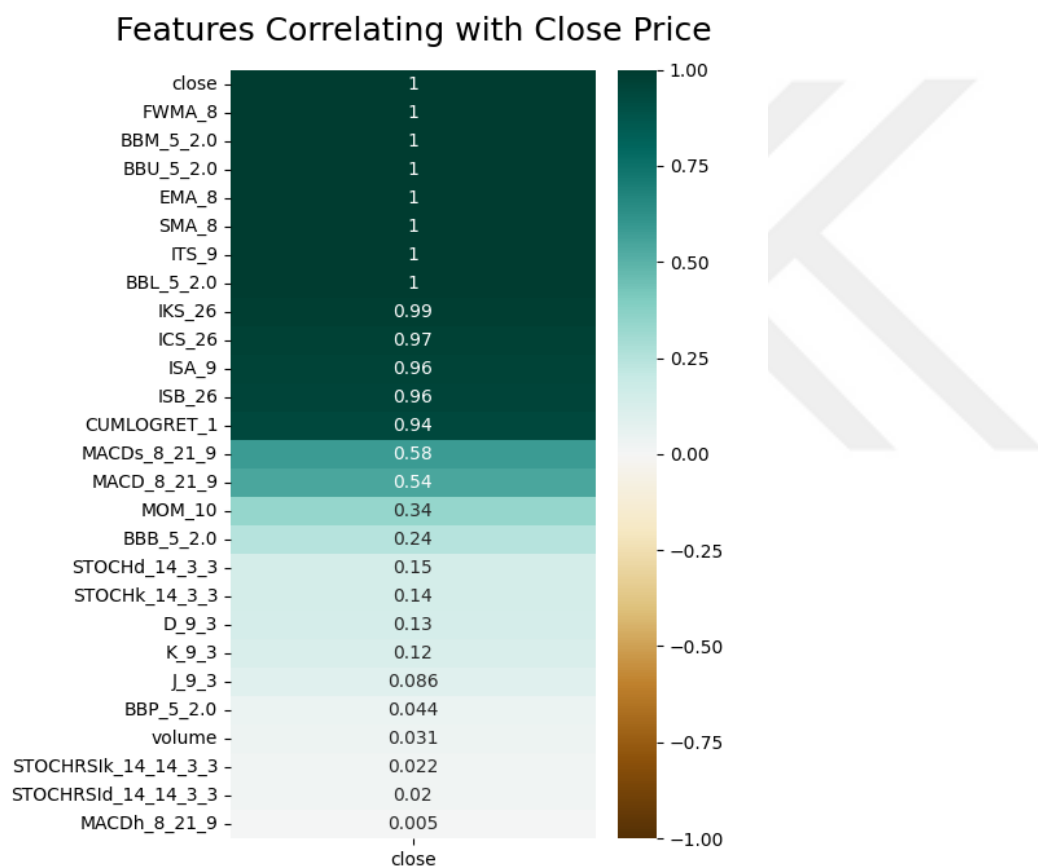


Figure 4.8. Feature Selection Criteria for TUPRS.IS

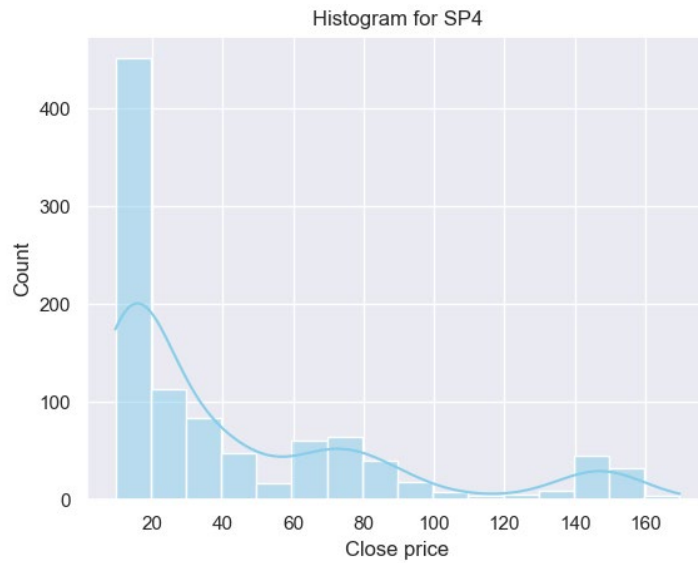


Figure 4.9. Histogram for TUPRS.IS

In Figure 4.10, the technical indicators are determined according to the threshold value of the correlation of each technical indicator considered with the closing price for ISCTR.IS. The resulting indicators are FWMA_8, BBL_5_2.0, BBM_5_2.0, BBU_5_2.0, EMA_8, SMA_8, ISA_9, ISB_26, ITS_9, IKS_26, and ICS_26. The final indicator is CUMLOGRET_1. The Figure 4.11 illustrates the concentration of the closing price of ISCTR.IS within specific ranges. Figure 4.11 demonstrates that the distribution is not symmetrical, with the closing price exhibiting a greater concentration in the 0-5 range. Conversely, the closing price values in the 1%-20 range are comparatively lower.

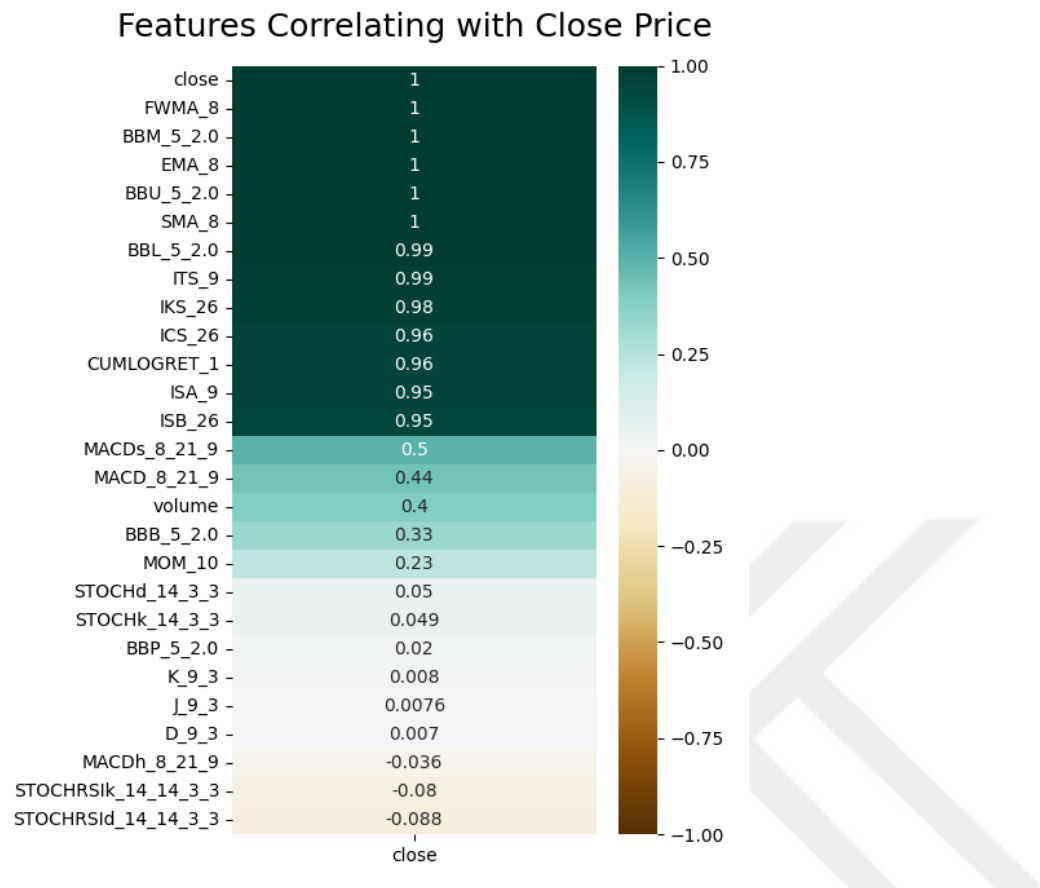


Figure 4.10. Feature Selection Criteria for ISCTR.IS

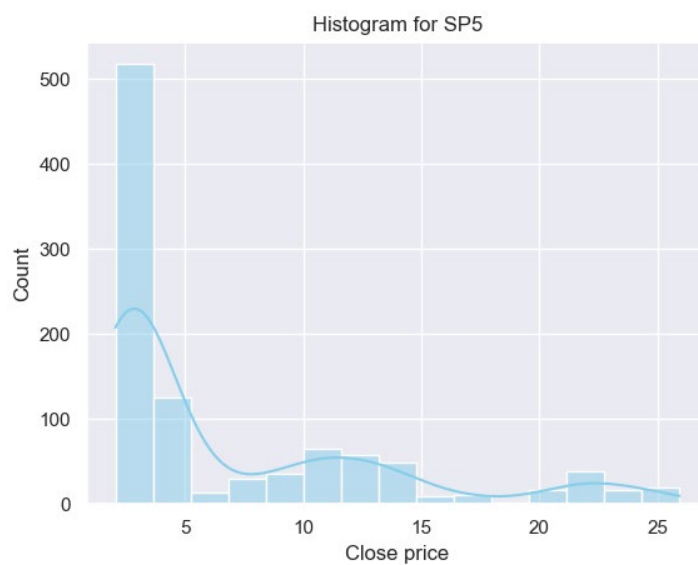


Figure 4.11. Histogram for ISCTR.IS

4.1. Computational Results

The objective of this study was to gain valuable insights by examining non-stationary time series through the lens of neural network architectures. The use of recurrent neural networks and their derivatives has resulted in a notable degree of success in the prediction of stock data, particularly given their efficacy in processing sequential data. The closing prices of KCHOL.IS, THYAO.IS, FROTO.IS, TUPRS.IS, and ISCTR.IS were estimated using historical data of stocks. The combination of technical indicators and raw data resulted in the generation of a more meaningful, less noisy, and richer information set. Subsequently, the procedures for data preparation, model creation, training, and prediction were carried out to facilitate the generation of predictions through the utilization of DL models. Subsequently, the technical indicators of the data were calculated and subsequently divided into two distinct categories: testing and training. The application of the min-max scaler method for the scaling of features in the data set has been identified as a key factor in enhancing the performance of the model. The parameters of the DL models utilized in this thesis were determined through a process of trial and error. The trial-and-error method entails a systematic examination of diverse parameter combinations to enhance the model's performance. In this process, a single-layer LSTM model was trained with each parameter set and its performance on validation data was evaluated. The final model was installed by selecting the parameters that yielded the optimal results. The parameters identified through this experimental process were maintained as constant variables in subsequent deep learning methods, facilitating comparative analysis of the resulting numerical data.

In the architectural design of the model, the number of layers and activation functions were defined, and subsequently. A total of 729 data points were utilized in the model training process. The weights and biases are optimized using the training data. In this process, the model is trained to identify patterns and relationships within the data set. The objective is to facilitate the model's learning process by providing it with training data. The epoch is defined based on a process of trial and error and determines the number of times the model passes over the entire training data set. The "batch size" value, which denotes the number of data samples to be processed at each stage, has been set to 8. Once the model has undergone training, it is then tested on data that it has not previously encountered. A variety of metrics, including MSE, MAE, and R^2 , are employed to assess the performance of the model.

The results presented in Table 1 were obtained using the MSE loss function and the Adam optimizer. Adam is an optimization algorithm utilized to update recurrent network weights based on training data (Pawar, 2019). The Adam optimizer is used to update the weights and biases of the model to minimize the loss function. A variety of metrics can be employed to assess the accuracy of a prediction model. One such metric is the MSE.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (4.1)$$

It is a commonly utilized performance metric for gauging accuracy. The initial value is represented by y_i , while the predicted value is represented by $\hat{y}_i$. In this context, n represents the total amount of data (Equation 4.1).

In Table 4.4, the hyperparameter settings for the single-layer LSTM model are established. Common choices for the number of neurons per layer are 32, 64, and 128; we chose 64 neurons in our model. The activation functions for each layer are 'relu', 'sigmoid', and 'tanh', and the ReLU function is chosen for this model. The learning rate is set to 0.001 to prevent the model from overfitting. The learning rate is very important in updating the weights of the model. It involves determining the batch size, which defines how many samples the model will be updated in each iteration. Considering the usually large size of the training dataset, the batch size divides the training process into smaller batches instead of processing the entire dataset at once. A batch size of 8 is selected among the common batch sizes in our model. After the hyperparameters are set, the Adam algorithm and other optimization techniques are used during model training. Adam updates the parameters of the model and optimizes the learning process. In optimization algorithms, the "step size" is obtained by multiplying the slope by the learning rate determined by the gradient descent algorithm.

Table 4.4. First hidden Layer LSTM

First hidden Layer LSTM	
Neuron	64
Activation Function	ReLU
Optimizer	Adam
Obj Function	MSE
Epochs	100
Batch Size	8
Learning Rate	0.001
Step Size	8

In Table 4.5., the number of neurons in the second layer and the number of neurons in the third layer in the three-layer LSTM method was determined to be 128. The activation function employed in these layers was Tanh. Once the training of the model is complete, the prediction step is initiated using the test data. A variety of measurement metrics have been employed to assess the model's performance error.

Table 4.5. The three-layer long short-term memory

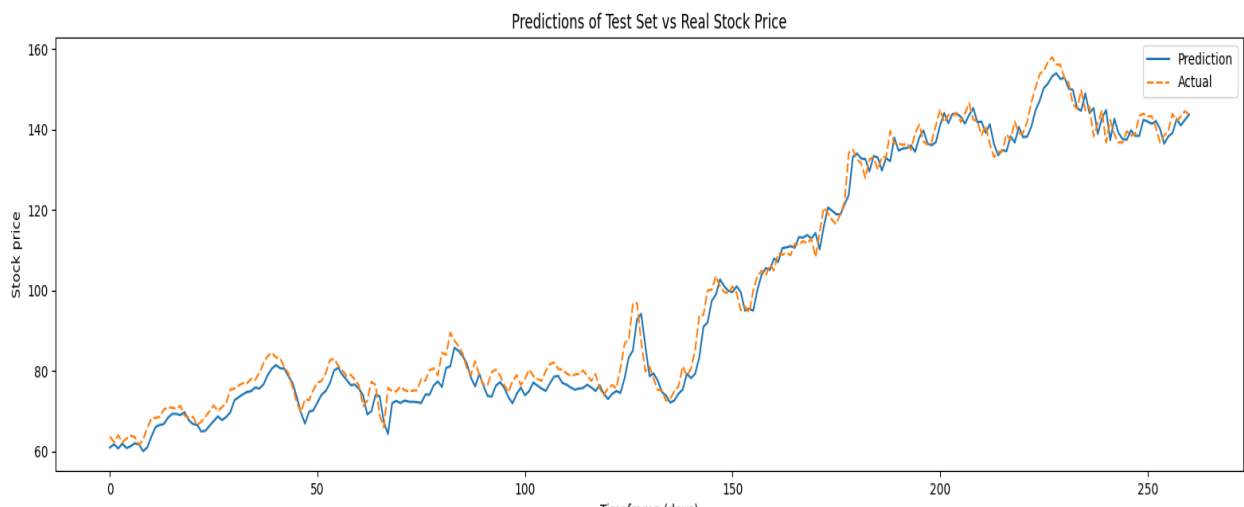
	First Hidden Layer LSTM	Second Hidden Layer LSTM	Third Hidden Layer LSTM
Number Of Neurons	64	128	128
Activation Function	ReLU	Tanh	Tanh
Optimizer	Adam		
Obj Function	MSE		
Epochs	100		
Batch Size	8		
Learning Rate	0.001		
Step Size	8		

In Table 4.6, the number of neurons in the second layer and the number of neurons in the third layer in the three-layer BiLSTM method was determined to be 128. In these layers, the activation function was Tanh.

Table 4.6. The three-layer BiLSTM

	First Hidden Layer (Bidirectional)	Second Hidden Layer (Bidirectional)	Third Hidden Layer (Bidirectional)
Neuron	64	128	128
Activation Function	ReLU	Tanh	Tanh
Optimizer	Adam		
Obj Function	MSE		
Epochs	100		
Batch Size	8		
Learning Rate	0.001		
Step Size	8		

The following section presents the application of the proposed BiLSTM model for forecasting the future price trajectory of KCHOL.IS, THYAO.IS, FROTO.IS, TUPRS.IS, and ISCTR.IS shares, based on historical data. The abscissa represents the number of days, while the ordinate depicts the share price. Upon plotting the estimated test price (blue) and the actual price (red), it becomes evident that there is a negligible discrepancy between the two. The data from the 250-day interval are presented in Figures 4.12, 13, 14, 15, and 16. As illustrated in Figures 4.12, 13, and 14, the BiLSTM method demonstrates the capacity to discern the shifts in trend over the 200-to-250-day interval. However, as illustrated in Figures 4.15 and 4.16, the network demonstrates less success in capturing trends and dynamics within the period between 200 and 250 days.

**Figure 4.12.** Plot for Real value vs Predicted value for KCHOL.IS using BiLSTM

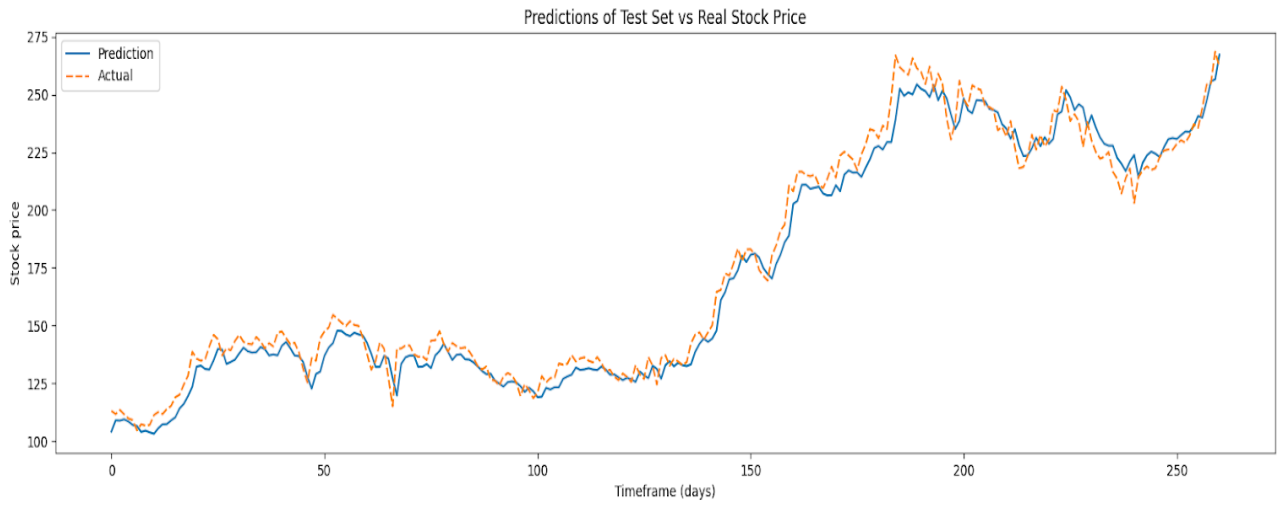


Figure 4.13. Plot for Real value vs Predicted value for THYAO.IS using BiLSTM

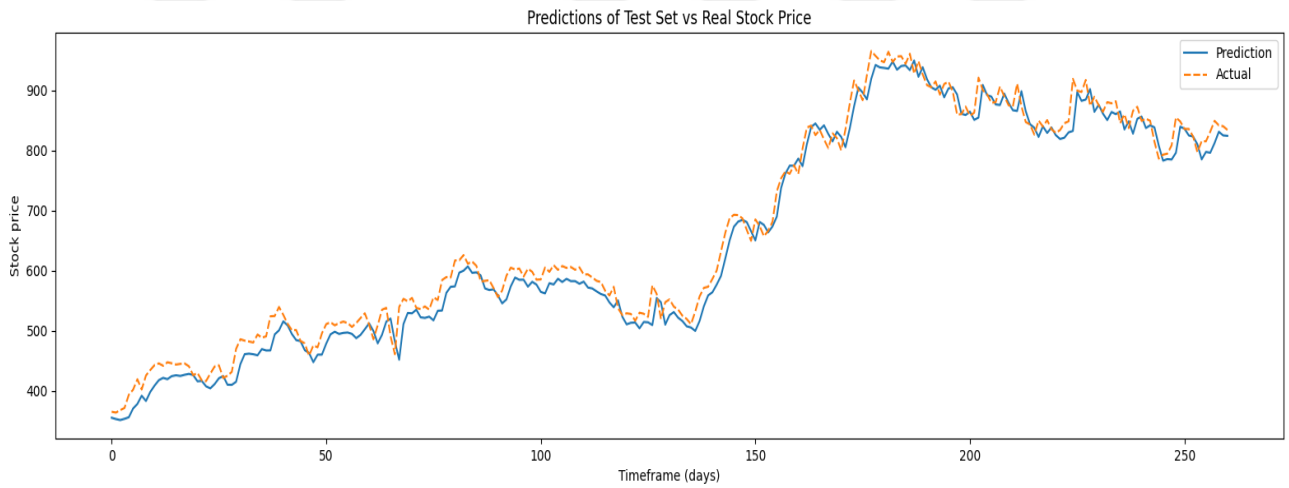


Figure 4.14. Plot for Real value vs Predicted value for FROTO.IS using BiLSTM

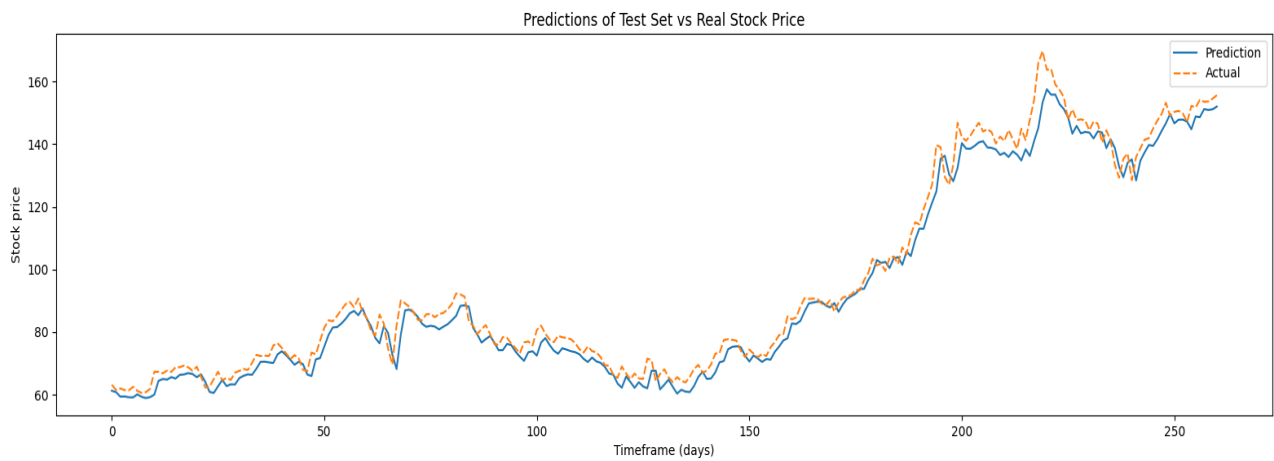


Figure 4.15. Plot for Real value vs Predicted value for TUPRS.IS using BiLSTM

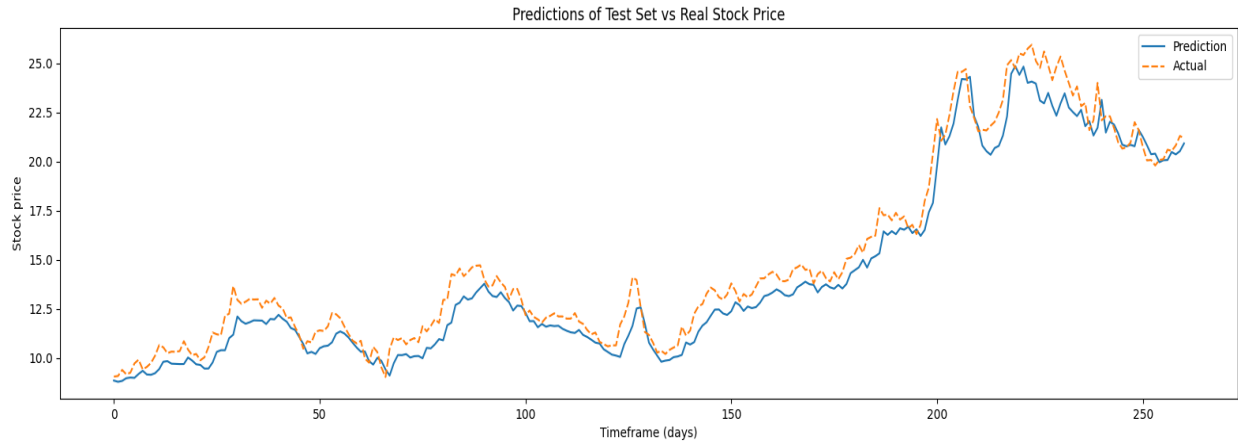


Figure 4.16. Plot for Real value vs Predicted value for ISCTR.IS using BiLSTM

A prediction was made regarding the future price of KCHOL.IS, THYAO.IS, FROTO.IS, TUPRS.IS, and ISCTR.IS shares, based on an analysis of historical data. The resulting stock price prediction graph, which illustrates the performance of the CNN-LSTM model, is presented in the figure below. The abscissa represents the number of days, while the ordinate depicts the share price. The algorithm has been successfully plotted with the predicted test price (blue) and the actual price (red). In the aggregate, the discrepancy between the estimated test price (blue) and the actual price (red) is minimal. The data presented in Fig. 4.17, Fig. 4.18, Fig. 4.19, Fig. 4.20, and Fig. 4.21 span a 250-day interval. As illustrated in Figures 4.17, 18, 19 and 20, the CNN-LSTM method demonstrates the capacity to discern shifts in trend over time intervals between 200 and 250. However, Fig. 4.21 illustrates that the network did not demonstrate the same degree of success in identifying trends and dynamics within the 200-250 days.

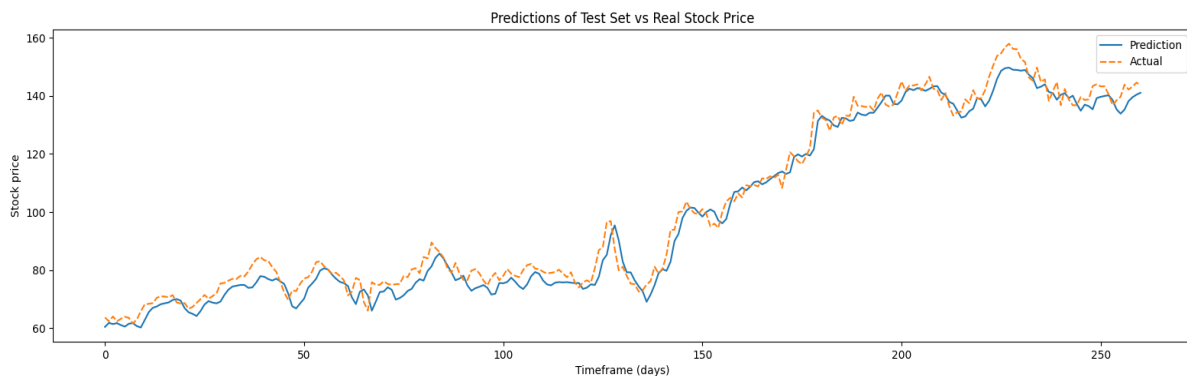


Figure 4.17. Plot for Real value vs Predicted value for KCHOL.IS using CNN-LSTM

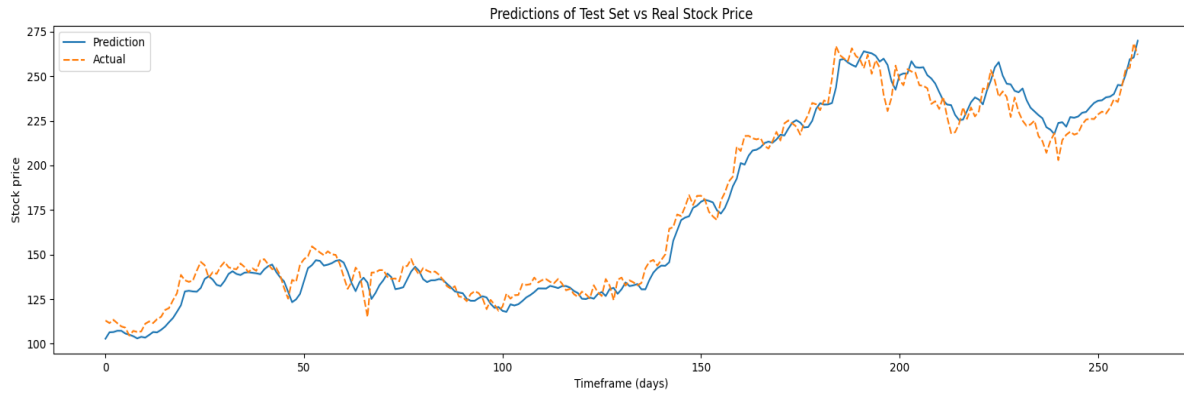


Figure 4.18. Plot for Real value vs Predicted value for THYAO.IS using CNN-LSTM

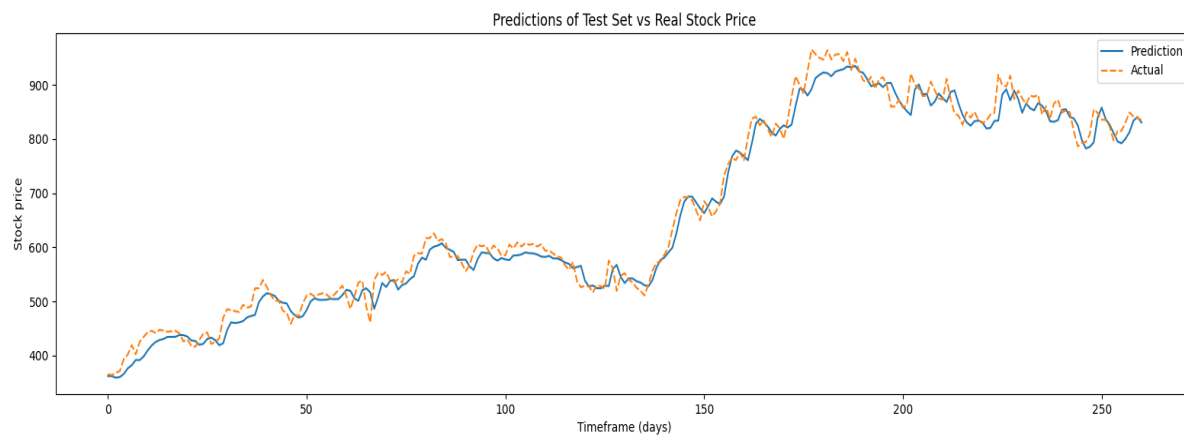


Figure 4.19. Plot for Real value vs Predicted value for FROTO.IS using CNN-LSTM

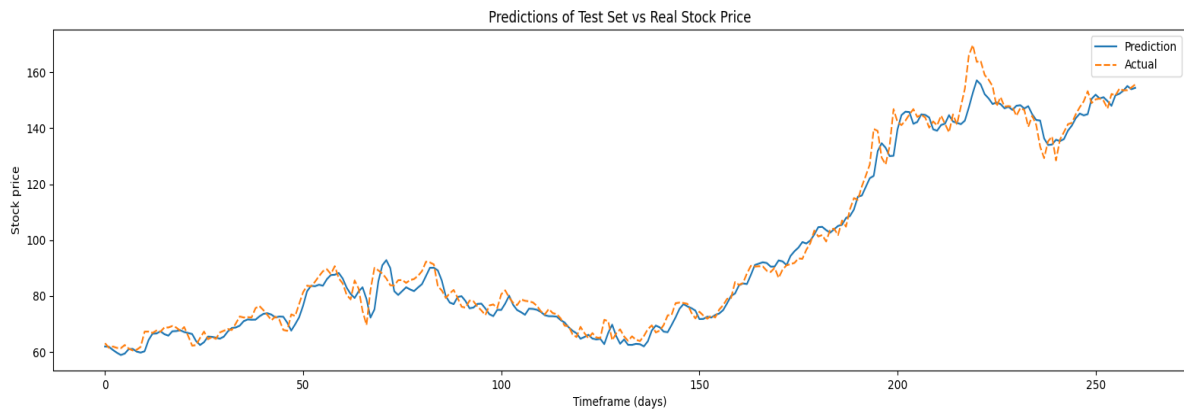


Figure 4.20. Plot for Real value vs Predicted value for TUPRS.IS using CNN-LSTM

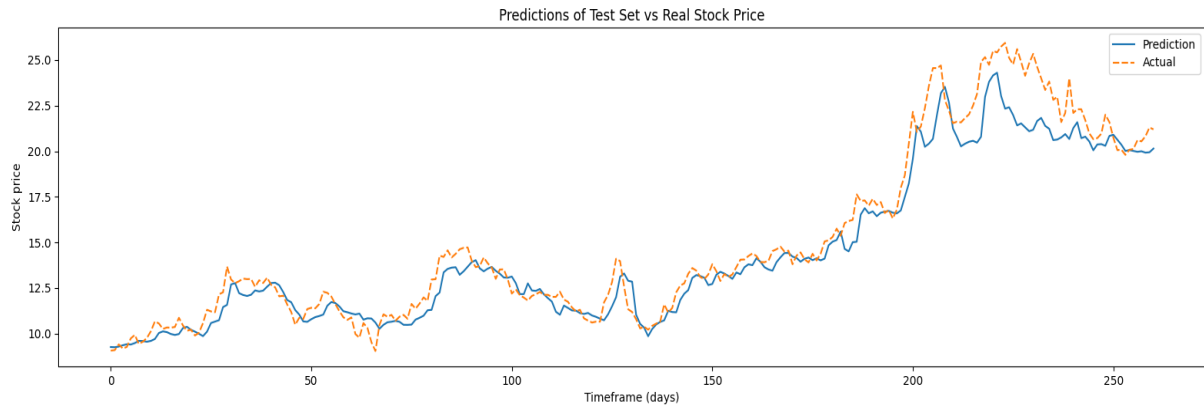


Figure 4.21. Plot for Real value vs Predicted value for ISCTR.IS using CNN-LSTM

A prediction was made regarding the future price of KCHOL.IS, THYAO.IS, FROTO.IS, TUPRS.IS, and ISCTR.IS shares, based on an analysis of historical data. The resulting stock price prediction graph, which illustrates the performance of the 3L-LSTM model, is presented in the figure below. The abscissa represents the number of days, while the ordinate denotes the share price. The algorithm has been successfully plotted with the predicted test price (blue) and the actual price (red). In the aggregate, the discrepancy between the estimated test price (blue) and the actual price (red) is minimal. The data presented in Fig. 4.22, 23, 24, 25, and 26 represent a 250-day interval. The three-layer LSTM method demonstrates a clear ability to capture trends and dynamics throughout the examined time period, as well as to identify and represent changes in trend.

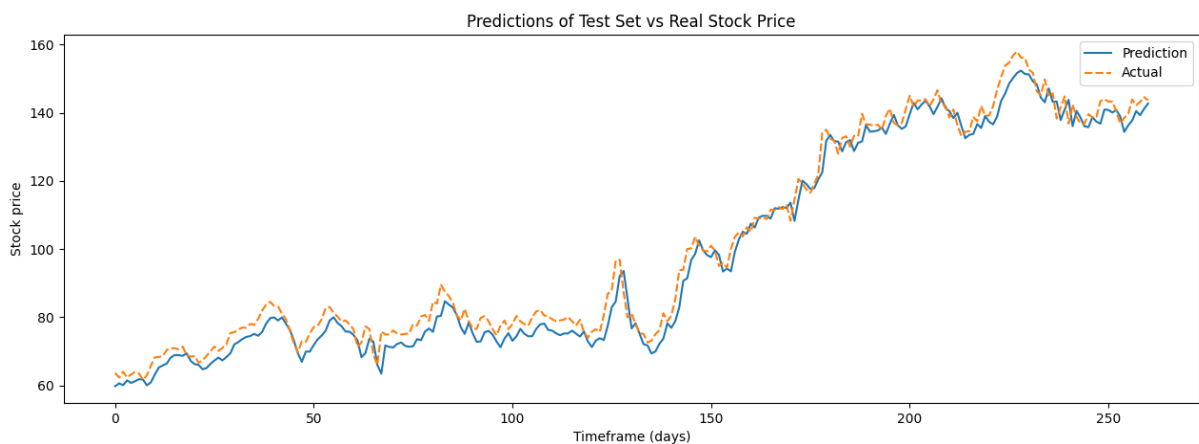


Figure 4.22. Plot for Real value vs Predicted value for KCHOL.IS using Three-Layer-LSTM

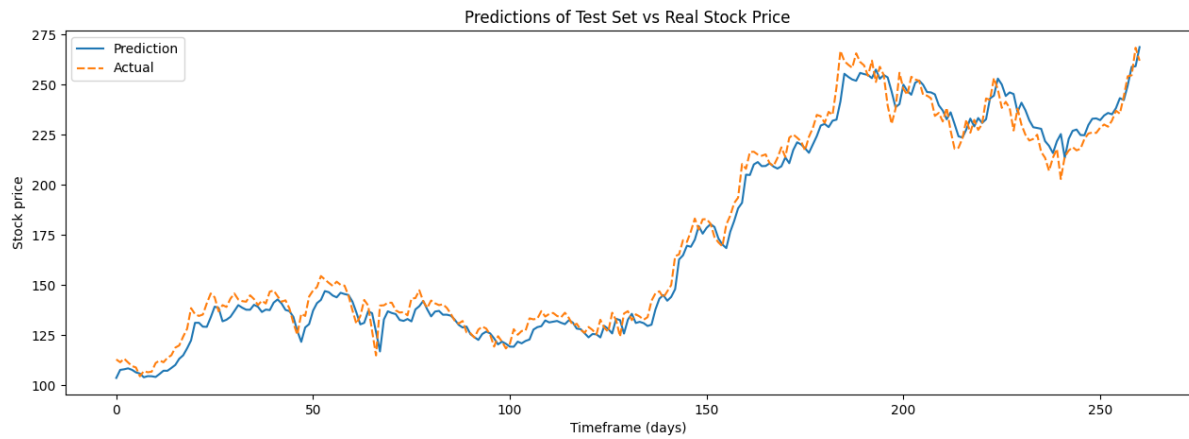


Figure 4.23. Plot for Real value vs Predicted value for THYAO.IS using Three-Layer-LSTM

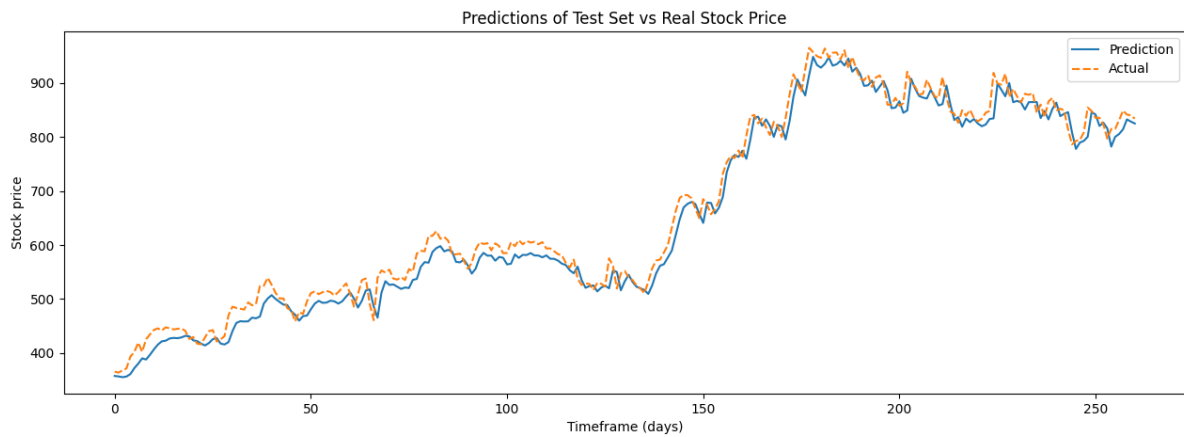


Figure 4.24. Plot for Real value vs Predicted value for FROTO.IS using Three-Layer-LSTM

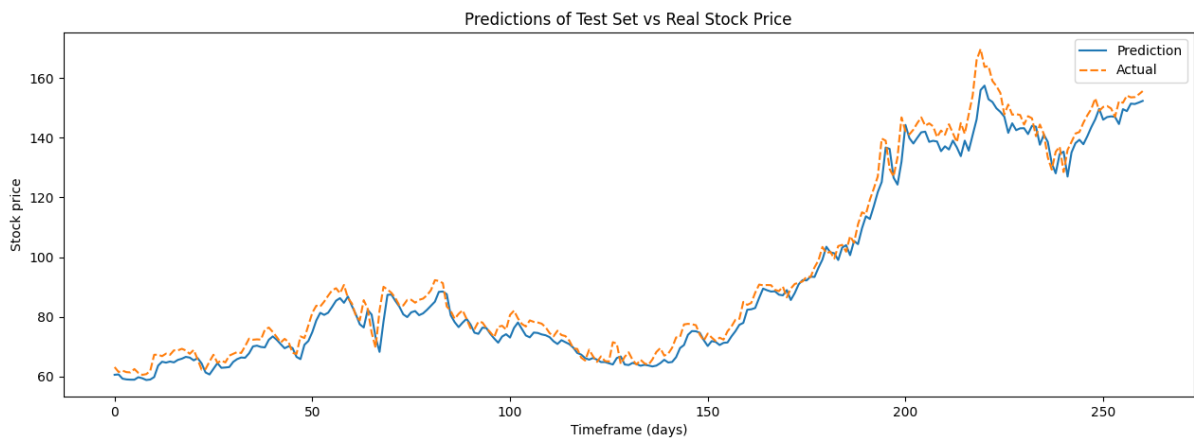


Figure 4.25. Plot for Real value vs Predicted value for TUPRS.IS using Three-Layer-LSTM

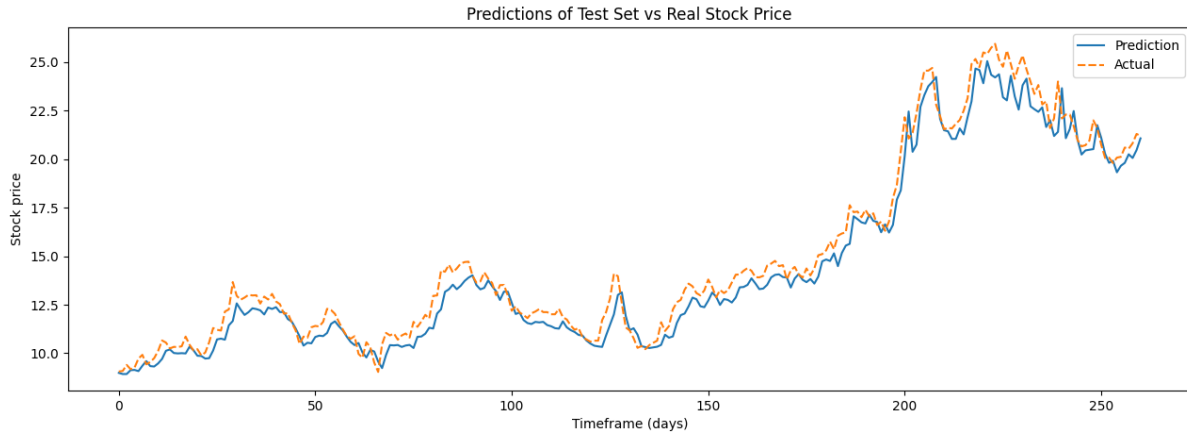


Figure 4.26. Plot for Real value vs Predicted value for ISCTR.IS using Three-Layer-LSTM

A prediction was made regarding the future price of KCHOL.IS, THYAO.IS, FROTO.IS, TUPRS.IS, and ISCTR.IS shares, based on an analysis of historical data. The figure presents a stock price prediction graph illustrating the performance of the Single Layer-LSTM model. The abscissa represents the number of days, while the ordinate denotes the share price. The algorithm has been successfully plotted with the predicted test price (blue) and the actual price (red). In the aggregate, the discrepancy between the estimated test price (blue) and the actual price (red) is minimal. The data presented in Fig. 4.27, 4.28, 4.29, 4.30, and 4.31 represent a 250-day interval. As illustrated in Figures 4.27, 4.28, 4.29, 4.30, the Single Layer-LSTM method demonstrates the capacity to discern and replicate the fluctuations in the trend over the specified time interval, spanning 200 to 250 days. However, the network depicted in Figure 4.31 was unable to achieve the same degree of success in capturing trends and dynamics within the specified period between 200 and 250 days.

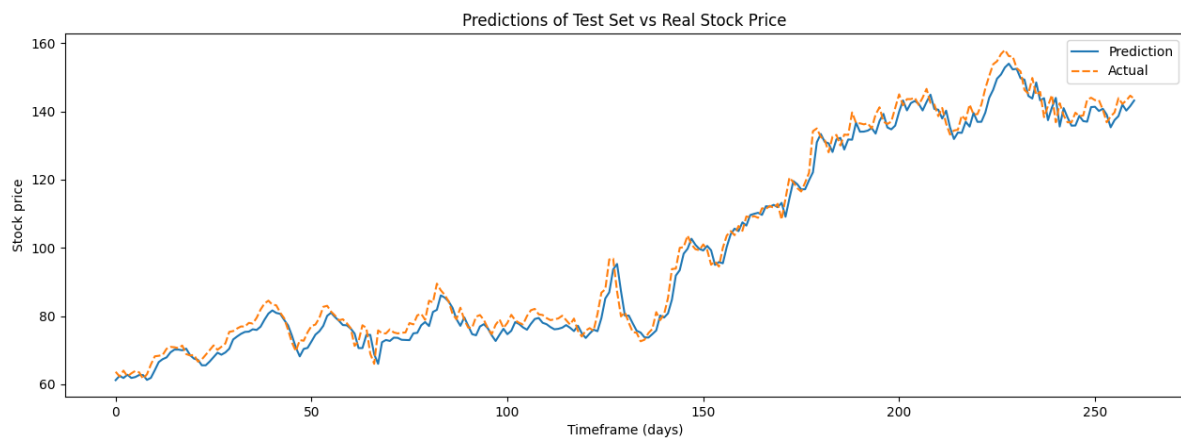


Figure 4.27. Plot for Real value vs Predicted value for KCHOL.IS using SingleLayer-LSTM

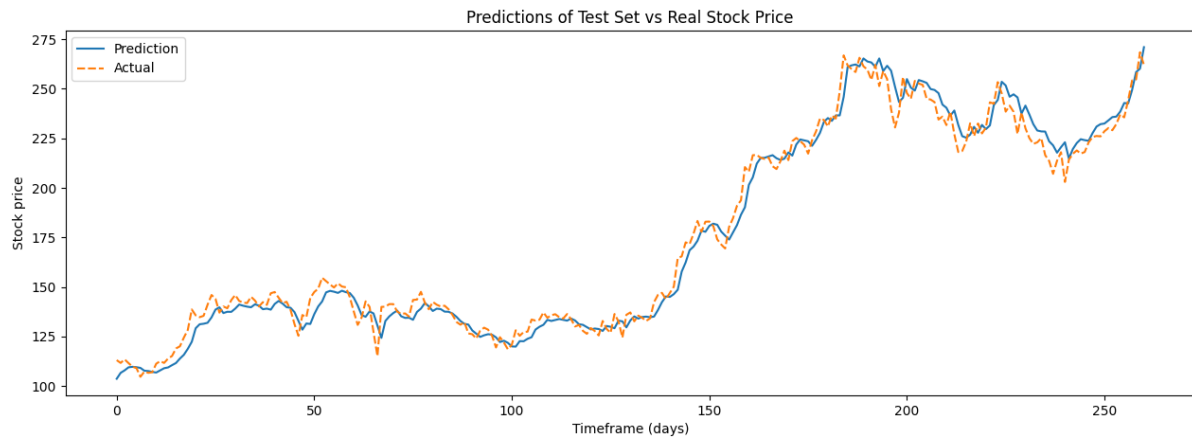


Figure 4.28. Plot for Real value vs Predicted value for THYAO.IS using SingleLayer-LSTM

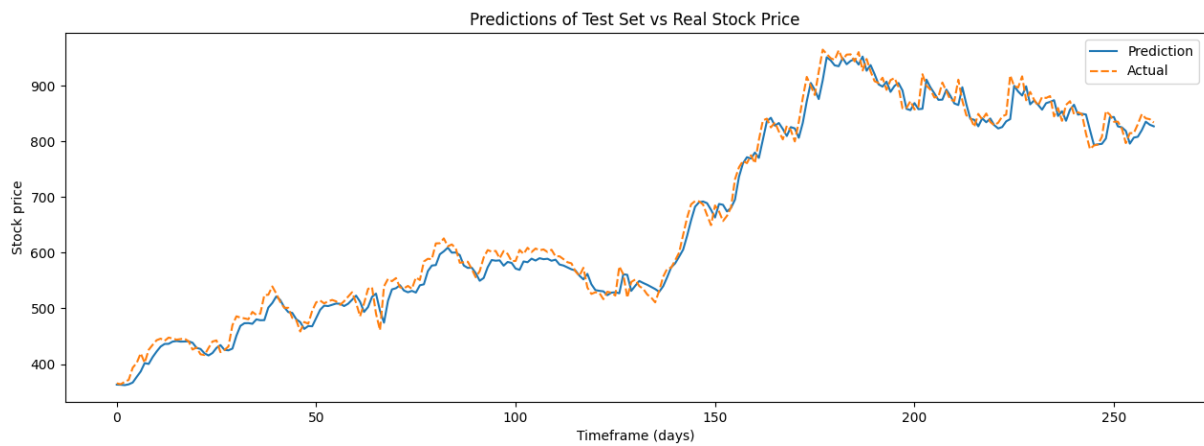


Figure 4.29. Plot for Real value vs Predicted value for FROTO.IS using SingleLayer-LSTM

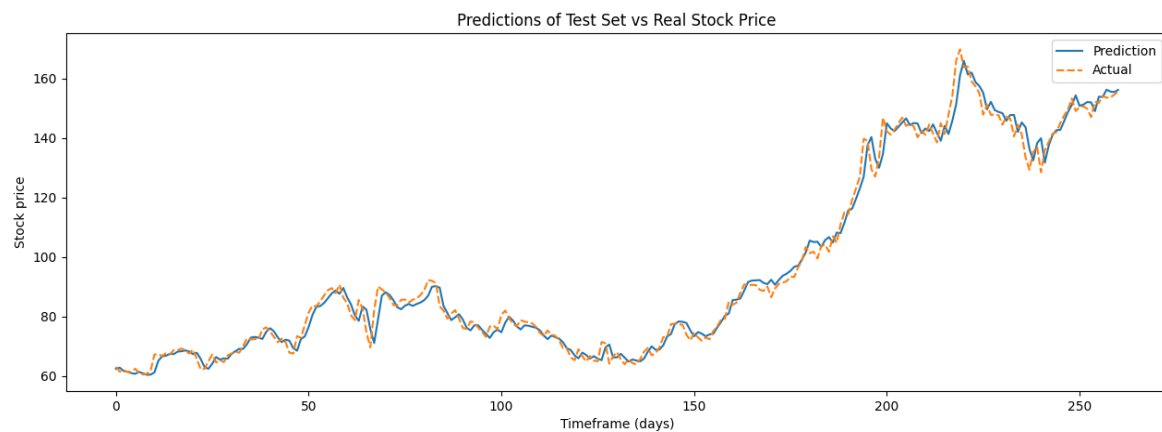


Figure 4.30. Plot for Real value vs Predicted value for TUPRS.IS using SingleLayer-LSTM

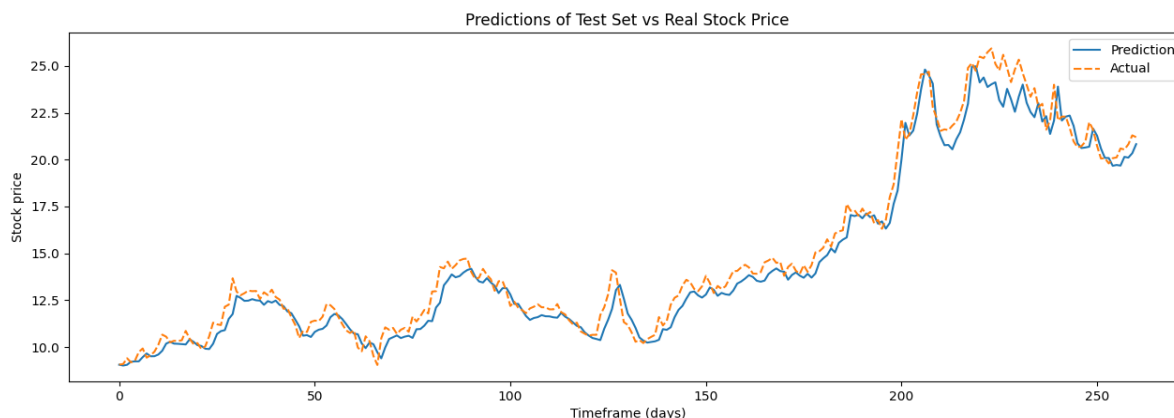


Figure 4.31. Plot for Real value vs Predicted value for ISCTR.IS using SingleLayer-LSTM

Performance evaluation metrics are employed to assess the accuracy and efficacy of output values derived from input values within a given model. The values of the performance evaluation metrics and CPU time values for each model considered in the thesis are provided in the Table 4.7. A lower MSE value indicates a superior model performance. As evidenced by the stocks FROTO.IS, ISCTR.IS, KCHOL.IS, THYAO.IS, and TUPRS.IS, the most optimal model is the single-layer LSTM method. A lower mean MAPE indicates a superior model. The Single Layer-LSTM method yielded the optimal MAPE result when evaluated using the FROTO.IS, ISCTR.IS, KCHOL.IS, THYAO.IS, and TUPRS.IS stocks. The coefficient of determination (R^2) is a measure of how well the model fits the data. A value of 1 indicates an optimal model (Chen, 2021). As evidenced by the stocks FROTO.IS, ISCTR.IS, KCHOL.IS, THYAO.IS, and TUPRS.IS, the optimal model is the single-layer LSTM method.

Studies comparing different DL algorithms have been examined in the literature. Similar to this study, different machine algorithms were used, but no study was found that paid more attention to pre-processing to normalize the data. The results obtained in the findings section provide useful advice for investors.

Table 4.7. Comparison of the errors of the prediction results of the traditional stock market indices by various models.

FROTO.IS	EVS	MAE	MAPE	MDA	MSE	MSLE	RMSE	r2_score	CPUtime
Single Layer-LSTM	0,9884	0,0205	0,0482	0,5320	0,0007	0,0003	0,0266	0,9865	22,6202
Three-Layer LSTM	0,9881	0,0259	0,0638	0,5312	0,0010	0,0005	0,0317	0,9808	69,1803
Single Layer BiLSTM	0,9877	0,0261	0,0665	0,5305	0,0010	0,0005	0,0320	0,9804	111,8846
CNN-LSTM	0,9854	0,0234	0,0557	0,5316	0,0009	0,0004	0,0304	0,9824	60,8400
ISCTR.IS	EVS	MAE	MAPE	MDA	MSE	MSLE	RMSE	r2_score	CPUtime
Single Layer-LSTM	0,9815	0,0279	0,0796	0,4927	0,0013	0,0006	0,0362	0,9729	22,4726
Three-Layer LSTM	0,9832	0,0307	0,0888	0,4927	0,0015	0,0007	0,0386	0,9692	68,0938
Single Layer BiLSTM	0,9808	0,0386	0,1160	0,4906	0,0022	0,0011	0,0466	0,9552	113,6686
CNN-LSTM	0,9548	0,0375	0,0954	0,4925	0,0030	0,0012	0,0545	0,9386	45,2971
KCHOL.IS	EVS	MAE	MAPE	MDA	MSE	MSLE	RMSE	r2_score	CPUtime
Single Layer-LSTM	0,9903	0,0209	0,0542	0,5179	0,0007	0,0003	0,0265	0,9868	23,0434
Three-Layer LSTM	0,9895	0,0260	0,0718	0,5151	0,0010	0,0005	0,0316	0,9811	67,2800
Single Layer BiLSTM	0,9891	0,0222	0,0611	0,5163	0,0008	0,0004	0,0277	0,9856	122,9685
CNN-LSTM	0,9864	0,0264	0,0721	0,5162	0,0011	0,0005	0,0327	0,9799	45,5374
THYAO.IS	EVS	MAE	MAPE	MDA	MSE	MSLE	RMSE	r2_score	CPUtime
Single Layer-LSTM	0,9819	0,0240	0,0602	0,5248	0,0010	0,0004	0,0311	0,9818	22,4991
Three-Layer LSTM	0,9815	0,0264	0,0674	0,5263	0,0011	0,0005	0,0333	0,9792	67,5548
Single Layer BiLSTM	0,9825	0,0264	0,0662	0,5264	0,0011	0,0005	0,0336	0,9788	116,9388
CNN-LSTM	0,9751	0,0297	0,0757	0,5255	0,0014	0,0006	0,0368	0,9746	48,8059

TUPRS.IS	EVS	MAE	MAPE	MDA	MSE	MSLE	RMSE	r2_score	CPUtime
Single Layer-LSTM	0,9886	0,0165	0,0458	0,4967	0,0005	0,0002	0,0231	0,9886	22,4607
Three-Layer LSTM	0,9880	0,0244	0,0676	0,4973	0,0010	0,0005	0,0317	0,9785	68,2111
Single Layer BiLSTM	0,9884	0,0240	0,0681	0,4979	0,0010	0,0004	0,0309	0,9795	116,3628
CNN-LSTM	0,9843	0,0201	0,0578	0,4971	0,0008	0,0004	0,0281	0,9831	43,8241



5. CONCLUSIONS

In this study, the stock price prediction was conducted using DL model. The task of predicting stock prices is inherently challenging due to the dynamic, non-linear, complex, non-parametric, and chaotic nature of the stock market. Stock market movements are influenced by a multitude of macroeconomic factors, including political events, corporate policies, and prevailing economic conditions. In this sector, it is not always feasible to obtain realistic predictions due to the non-linearity of the data. Consequently, developing AI systems for such predictions necessitates an iterative process of knowledge discovery and system improvement through data mining, knowledge engineering, theoretical and data-driven modeling, as well as trial and error experiments. The primary objective of investors is to attempt to anticipate future stock prices or returns. It is often the case that a single method is insufficient. Consequently, more comprehensive predictions are made by combining multiple analyses. Given the numerous external factors that influence stock forecasting, it is crucial to implement a robust risk management strategy to navigate these uncertainties. This is why a variety of analytical techniques and methods are employed for prediction purposes. In this study, the shares of KCHOL.IS, THYAO.IS, FROTO.IS, TUPRS.IS, and ISCTR.IS, which is among the five largest companies in the BIST100 index according to market capitalization, was selected for analysis. Initially, it was hypothesized that utilizing raw data for stock price prediction would result in a lower success rate. To prevent the models from negatively affecting the learning process and reducing the prediction performance, technical indicators were calculated without the use of the data directly for prediction. This approach yielded more meaningful and descriptive features.

The proposed hypothesis is that the use of CNN, LSTM networks, and bidirectional LSTM networks as time series-based prediction models based on DL can facilitate more accurate stock price prediction. In the present era, the aforementioned models are employed with greater frequency than other regression models for stock price prediction. The preference for DL algorithms in stock prediction is predicated on their robust data processing capabilities, capacity to model intricate relationships, and high prediction accuracy. One of its key features is its capacity to extract meaningful features from data sets of considerable size, with the ability to identify and process these features automatically. Furthermore, they offer superior accuracy and performance compared to other traditional modeling techniques. Among the DL models,

the single-layer LSTM, three-layer LSTM, three-layer BiLSTM, and CNN-LSTM methods were employed. By appropriately adjusting the hyperparameters in the models, the results were updated and the values most closely aligned with reality were obtained. The optimal values for the model were determined through a process of trial and error. The evaluation and improvement of model performance hinges on the use of appropriate error metrics. These metrics are employed to measure the degree of accuracy exhibited by the model predictions. In the context of error metrics, a lower MSE and RMSE indicate a superior prediction outcome. Conversely, a value of R^2 that is closer to 1 signifies an enhanced prediction result. The results of the prediction models indicate that they are capable of making successful predictions at a certain level. Among the models recommended for each stock, the least error is observed in the MSE, RMSE, and R^2 metrics when compared to the other three models for single-layer LSTM. Furthermore, the least favorable outcomes were observed in the single-layer CNN-LSTM methods for KCHOL.IS, THYAO.IS, and ISCTR.IS stocks, and in the single-layer BiLSTM models for FROTO.IS and TUPRS.IS stocks. It has been demonstrated that predictions can be made using historical data.

Consequently, stock price prediction represents a crucial strategic instrument for both investors and companies. The findings of this study indicate that DL algorithms have considerable potential for use in stock prediction. Moreover, the sophisticated techniques and methodologies employed will confer substantial benefits in economic decision-making processes, enhancing forecast precision. The application of sophisticated analytical techniques in the financial sector enables investors to enhance risk management and make more informed strategic decisions. Further research in this field may establish new benchmarks for market analysis, enhancing the precision and dependability of financial modeling.

REFERENCES

- Adebiyi, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Comparison of ARIMA and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics*, 2014(1), 614342.
- Agatonovic-Kustrin, S., & Beresford, R. (2000). Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *Journal of pharmaceutical and biomedical analysis*, 22(5), 717-727.
- Akcan, A., & Kartal, C. (2011). İMKB Sigorta Endeksini Oluşturan Sirketlerin Hisse Senedi Fiyatlarının Yapay Sinir Ağları İle Tahmini. *Muhasebe ve Finansman Dergisi*, (51), 27-40.
- Alexandropoulos, S. A. N., Kotsiantis, S. B., & Vrahatis, M. N. (2019). Data preprocessing in predictive data mining. *The Knowledge Engineering Review*, 34, e1.
- Almeida, L., & Vieira, E. (2023). Technical analysis, fundamental analysis, and Ichimoku Dynamics: A Bibliometric analysis. *Risks*, 11(8), 142.
- Althelaya, K. A., El-Alfy, E. S. M., & Mohammed, S. (2018, April). Evaluation of bidirectional LSTM for short-and long-term stock market prediction. In 2018 9th international conference on information and communication systems (ICICS) (pp. 151-156). IEEE.
- Anand, C. (2021). Comparison of stock price prediction models using pre-trained neural networks. *Journal of Ubiquitous Computing and Communication Technologies*, 3(2), 122-134.
- Ayyildiz, N., & Iskenderoglu, O. (2024). How effective is machine learning in stock market predictions?. *Heliyon*, 10(2).
- Bakas, I., & Kontoleon, K. J. (2021). Performance evaluation of artificial neural networks (Ann) predicting heat transfer through masonry walls exposed to fire. *Applied Sciences*, 11(23), 11435.

Bilen, K., & Bahar, O. (2023). Teknik Analiz İndikatörlerinin Etkinlik Analizi: Borsa İstanbul Turizm Sektörü Üzerine Bir Uygulama. *Anatolia: Turizm Araştırmaları Dergisi*, 34(2), 83-94. <https://doi.org/10.17123/atad.1291666>

Birgili, M. E. (2013). Teknik analiz yöntemini kullanan yatırımcıların davranışsal finans modelleri ile açıklanması: Türkiye'de bir araştırma (Master's thesis, Adnan Menderes Üniversitesi).

Briceno-Mena, L. A., Nnadili, M., Benton, M. G., & Romagnoli, J. A. (2022). Data mining and knowledge discovery in chemical processes: Effect of alternative processing techniques. *Data-Centric Engineering*, 3, e18. doi:10.1017/dce.2022.21

Bustos, O., & Pomares-Quimbaya, A. (2020). Stock market movement forecast: A systematic review. *Expert Systems with Applications*, 156, 113464.

Chauhan, R., Ghanshala, K. K., & Joshi, R. C. (2018, December). Convolutional neural network (CNN) for image detection and recognition. In 2018 first international conference on secure cyber computing and communication (ICSCCC) (pp. 278-282). IEEE.

Chen, J., Wen, Y., Nanekaran, Y. A., Suzaiddola, M. D., Chen, W., & Zhang, D. (2023). Machine learning techniques for stock price prediction and graphic signal recognition. *Engineering Applications of Artificial Intelligence*, 121, 106038.

Chen, S., & Ge, L. (2019). Exploring the attention mechanism in LSTM-based Hong Kong stock price movement prediction. *Quantitative Finance*, 19(9), 1507-1515.

Chen, S., Lan, X., Hu, Y., Liu, Q., & Deng, Y. The time series forecasting: from the aspect of network (2014). arXiv preprint arXiv:1403.1713.

Chen, Y., Fang, R., Liang, T., Sha, Z., Li, S., Yi, Y., ... & Song, H. (2021). Stock Price Forecast Based on CNN-BiLSTM-ECA Model. *Scientific Programming*, 2021(1), 2446543.

Chen, Y., Wu, J., & Wu, Z. (2022). China's commercial bank stock price prediction using a novel K-means-LSTM hybrid approach. *Expert Systems with Applications*, 202, 117370.

Çetinyokuş, T., & Gökçen, H. (2002). Borsada göstergelerle teknik analiz için bir karar destek sistemi. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 17(1), 43-58.

Çınaroğlu, E., & Avcı, T. (2020). THY hisse senedi değerinin yapay sinir ağları ile tahmini. *Atatürk Üniversitesi İktisadi ve İdari Bilimler Dergisi*, 34(1), 1-19.

Dey, S. K., Hossain, A., & Rahman, M. M. (2018, December). Implementation of a web application to predict diabetes disease: an approach using machine learning algorithm. In 2018 21st international conference of computer and information technology (ICCIT) (pp. 1-5). IEEE.

Di Persio, L., & Honchar, O. (2016). Artificial neural networks architectures for stock price prediction: Comparisons and applications. *International journal of circuits, systems and signal processing*, 10, 403-413.

Dike, H. U., Zhou, Y., Deveerasetty, K. K., & Wu, Q. (2018, October). Unsupervised learning based on artificial neural network: A review. In 2018 IEEE International Conference on Cyborg and Bionic Systems (CBS) (pp. 322-327). IEEE.

Doğan, F., & Türkoğlu, İ. (2018). Derin öğrenme algoritmalarının yaprak sınıflandırma başarımlarının karşılaştırılması. *Sakarya University Journal of Computer and Information Sciences*, 1(1), 10-21.

Doğan, F., & Türkoğlu, İ. (2019). Derin öğrenme modelleri ve uygulama alanlarına ilişkin bir derleme. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 10(2), 409-445.

Erden, C. (2021). Python ile Veri Madenciliği. KODLAB YAYIN DAĞITIM YAZILIM LTD. ŞTİ.

Feng, W., Guan, N., Li, Y., Zhang, X., & Luo, Z. (2017, May). Audio visual speech recognition with multimodal recurrent neural networks. In 2017 International Joint Conference on neural networks (IJCNN) (pp. 681-688). IEEE.

Fidelity Investments. (n.d.). Ichimoku Cloud. Fidelity. <https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/Ichimoku-Cloud>

Göçken, M., Özçalıcı, M., Boru, A., & Dosdoğru, A. T. (2016). Integrating metaheuristics and artificial neural networks for improved stock price prediction. *Expert Systems with Applications*, 44, 320-331.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10), 2222-2232.

Gülcü, Ş. (2022). Training of the feed forward artificial neural networks using dragonfly algorithm. *Applied Soft Computing*, 124, 109023.

Gülmez, B. (2023). Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm. *Expert Systems with Applications*, 227, 120346.

Güneş, H. (2023). Altın Fiyatı Üzerinde MACD İndikatörünün Test Edilmesi. *Ekonomi ve Finansal Araştırmalar Dergisi*, 5(2), 139-149.

Haq, M. A., Ahsan, A., & Gyani, J. (2023). Implementation of CNN for plant identification using UAV imagery. *International Journal of Advanced Computer Science and Applications*, 14(4).

Hossain, M. A., & Sajib, M. S. A. (2019). Classification of image using convolutional neural network (CNN). *Global Journal of Computer Science and Technology*, 19(2), 13-14.

Hossain, M. A., Karim, R., Thulasiram, R., Bruce, N. D., & Wang, Y. (2018, November). Hybrid deep learning model for stock price prediction. In 2018 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 1837-1844). IEEE.

Islam, M. R., & Nguyen, N. (2020). Comparison of financial models for stock price prediction. *Journal of Risk and Financial Management*, 13(8), 181.

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, 31(3), 685-695.

Jarrett, J. E., & Kyper, E. (2011). ARIMA modeling with intervention to forecast and analyze Chinese stock prices. *International Journal of Engineering Business Management*, 3, 17.

Jeeanunta, C., Chaysiri, R., & Thong, L. (2018, May). Stock price prediction with long short-term memory recurrent neural network. In *2018 International Conference on Embedded Systems and Intelligent Technology & International Conference on Information and Communication Technology for Embedded Systems (ICESIT-ICICTES)* (pp. 1-7). IEEE.

Jia, M., Huang, J., Pang, L., & Zhao, Q. (2019, July). Analysis and research on stock price of LSTM and bidirectional LSTM neural network. In *3rd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2019)* (pp. 467-473). Atlantis Press.

Kanwal, A., Lau, M. F., Ng, S. P., Sim, K. Y., & Chandrasekaran, S. (2022). BiCuDNNLSTM-1dCNN—A hybrid deep learning-based predictive model for stock price prediction. *Expert Systems with Applications*, 202, 117123.

Kaytan, M., Aydılek, İ. B., Yeroğlu, C., & Karci, A. (2022). Sigmoid-Gumbel: Yeni Bir Hibrit Aktivasyon Fonksiyonu. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 11(1), 29-45.

Kehinde, T. O., Chan, F. T., & Chung, S. H. (2023). Scientometric review and analysis of recent approaches to stock market forecasting: Two decades survey. *Expert Systems with Applications*, 213, 119299.

Keskenler, M. F., & Keskenler, E. F. (2017). Geçmişten günümüze yapay sinir ağları ve tarihçesi. *Takvim-i Vekayi*, 5(2), 8-18.

Khan, Z. H., Alin, T. S., & Hussain, M. A. (2011). Price prediction of share market using artificial neural network (ANN). *International Journal of Computer Applications*, 22(2), 42-47.

Kılıç, İ. (2023, May 15). Perceptron model: The foundation of neural networks. Medium. <https://medium.com/@ilyurek/perceptron-model-the-foundation-of-neural-networks-4db25b0148d>.

Kumar, P., Batra, S., & Raman, B. (2021). Deep neural network hyper-parameter tuning through twofold genetic approach. *Soft Computing*, 25(13), 8747-8771.

Kumbure, M. M., Lohrmann, C., Luukka, P., & Porras, J. (2022). Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, 197, 116659.

Laboissiere, L. A., Fernandes, R. A., & Lage, G. G. (2015). Maximum and minimum stock price forecasting of Brazilian power distribution companies based on artificial neural networks. *Applied Soft Computing*, 35, 66-74.

Li, C. (2023). Common Stock Indicators to Use for Beginner Traders.

Li, M., Zhu, Y., Shen, Y., & Angelova, M. (2023). Clustering-enhanced stock price prediction using deep learning. *World Wide Web*, 26(1), 207-232.

Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN-LSTM-based model to forecast stock prices. *Complexity*, 2020(1), 6622927.

Lu, W., Li, J., Wang, J., & Qin, L. (2021). A CNN-BiLSTM-AM method for stock price prediction. *Neural Computing and Applications*, 33(10), 4741-4753.

Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR)*. [Internet], 9(1), 381-386.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5, 115-133.

Mehtab, S., & Sen, J. (2020). Stock price prediction using convolutional neural networks on a multivariate timeseries. *arXiv preprint arXiv:2001.09769*.

Melli, G. (2017). Neural network output layer. Retrieved August 1, 2024, from https://www.gabormelli.com/RKB/Neural_Network_Output_Layer.

Mitchell, T. M., & Mitchell, T. M. (1997). *Machine learning* (Vol. 1, No. 9). New York: McGraw-hill.

Mostafa, M. M. (2010). Forecasting stock exchange movements using neural networks: empirical evidence from Kuwait. *Expert systems with applications*, 37(9), 6302-6309.

Muhammad, A.(2022). ForexBee. KDJ indicator. ForexBee. <https://forexbee.co/kdj-indicator/>

Muhammad, T., Aftab, A. B., Ibrahim, M., Ahsan, M. M., Muhu, M. M., Khan, S. I., & Alam, M. S. (2023). Transformer-based deep learning model for stock price prediction: A case study on Bangladesh stock market. *International Journal of Computational Intelligence and Applications*, 22(03), 2350013.

Nasiri, H., & Ebadzadeh, M. M. (2023). Multi-step-ahead stock price prediction using recurrent fuzzy neural network and variational mode decomposition. *Applied Soft Computing*, 148, 110867.

Nguyen, T. T., & Yoon, S. (2019). A novel approach to short-term stock price movement prediction using transfer learning. *Applied Sciences*, 9(22), 4745.

Nikou, M., Mansourfar, G., & Bagherzadeh, J. (2019). Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms. *Intelligent Systems in Accounting, Finance and Management*, 26(4), 164-174.

Onan, A., & Korukoğlu, S. (2016). Makine öğrenmesi yöntemlerinin görüş madenciliğinde kullanılması üzerine bir literatür araştırması. Pamukkale University Journal of Engineering Sciences, 22(2).

Özçalıcı, M. (2016). Yapay sinir ağları ile çok aşamalı fiyat tahmini: BIST30 senetleri üzerine bir araştırma. Dokuz Eylül Üniversitesi İktisadi İdari Bilimler Fakültesi Dergisi, 31(2), 209-227.

Öztemel, E. (2003). Yapay sinir ağları. PapatyaYayincilik, İstanbul.

Öztürk, H. (2016). Teknik analizde alım-satım sistemi oluşturma: Sistemin geçmişe yönelik testleri. Finansal Araştırmalar ve Çalışmalar Dergisi, 8(15), 469-493.

Öztürk, K., & Şahin, M. E. (2018). Yapay sinir ağları ve yapay zekâ'ya genel bir bakış. Takvim-i Vekayi, 6(2), 25-36.

Palit, A. K., & Popovic, D. (2006). Computational intelligence in time series forecasting: theory and engineering applications. Springer Science & Business Media.

Park, K., Jung, Y., Kim, K., & Park, S. K. (2020). Determination of deep learning model and optimum length of training data in the river with large fluctuations in flow rates. Water, 12(12), 3537.

Patel, J., Patel, M., & Darji, M. (2018). Stock price prediction using RNN and LSTM. Journal of Emerging Technologies and Innovative Research, 5(11), 1069-1079.

Pawar, K., Jalem, R. S., & Tiwari, V. (2019). Stock market price prediction using LSTM RNN. In Emerging Trends in Expert Applications and Security: Proceedings of ICETEAS 2018 (pp. 493-503). Springer Singapore.

Pratiwi, N. C., Ibrahim, N., Fu'adah, Y. N., & Masykuroh, K. (2020, December). Computer-aided detection (CAD) for COVID-19 based on chest X-Ray images using convolutional neural network. In IOP Conference Series: Materials Science and Engineering (Vol. 982, No. 1, p. 012004). IOP Publishing.

Rakesh, H., (2018, November 6). Introduction to neural networks. Data Driven Investor. <https://medium.datadriveninvestor.com/introduction-to-neural-networks-a0fe9ec0a947>.

Sakınç, S. Ö. (2018). Hisse senedine yatırım yaparken finansal analiz ile teknik analiz yöntemlerinin birlikte kullanılmasının önemi ve Bist’de bir uygulama. Sosyal Araştırmalar ve Davranış Bilimleri, 4(5), 134-153.

Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017, September). Stock price prediction using LSTM, RNN and CNN-sliding window model. In 2017 international conference on advances in computing, communications and informatics (icacci) (pp. 1643-1647). IEEE.

Ser, G., & Bati, C. T. (2019). Derin sinir ağları ile en iyi modelin belirlenmesi: mantar verileri üzerine keras uygulaması. Yuzuncu Yıl University Journal of Agricultural Sciences, 29(3), 406-417.

Shah, D., Campbell, W., & Zulkernine, F. H. (2018, December). A comparative study of LSTM and DNN for stock market forecasting. In 2018 IEEE international conference on big data (big data) (pp. 4148-4155). IEEE.

Sunny, M. A. I., Maswood, M. M. S., & Alharbi, A. G. (2020, October). Deep learning-based stock price prediction using LSTM and bi-directional LSTM model. In 2020 2nd novel intelligent and leading emerging sciences conference (NILES) (pp. 87-92). IEEE.

Şakrak, K. (2022). Hisse senedi gruplandırma duyurularının hisse senedi piyasa fiyatına etkisi Borsa İstanbul üzerine bir uygulama (Master's thesis, Hitit Üniversitesi Lisansüstü Eğitim Enstitüsü).

Şeker, A., Diri, B., & Balık, H. H. (2017). Derin öğrenme yöntemleri ve uygulamaları hakkında bir inceleme. Gazi Mühendislik Bilimleri Dergisi, 3(3), 47-64.

Şener, C., & Şener, U. (2019). Monte Carlo simülasyonu ile hisse senedi fiyat tahminleri. *Beykoz Akademi Dergisi*, 7(2), 294-306.

Şişmanoğlu, G., Koçer, F., Önde, M. A., & Sahingoz, O. K. (2020). Derin öğrenme yöntemleri ile borsada fiyat tahmini. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 9(1), 434-445.

Tahersima, H., Tahersima, M., Fesharaki, M., & Hamed, N. (2011, June). Forecasting stock exchange movements using neural networks: A case study. In *2011 International Conference on Future Computer Sciences and Application* (pp. 123-126). IEEE.

Tektaş, A., & Karataş, A. (2004). Yapay sinir ağları ve finans alanına uygulanması: Hisse senedi fiyat tahminlemesi. *Atatürk Üniversitesi İktisadi ve İdari Bilimler Dergisi*, 18(3-4).

Tekyaygil, F., (2020, May 18). Başarabilirsin: Makine öğrenmesine giriş — 3. Modelimizde hata, hata optimizasyonu ve hiperparametreler. *Deep Learning Türkiye*. <https://medium.com/deep-learning-turkiye/basarabilirsin-makine-ogrenmesine-giris-3-modelimizde-hata-hata-optimizasyonu-ve-hiper-3dfcb7251edc>.

Tosunoğlu, E., Yılmaz, R., Özeren, E., & Sağlam, Z. (2021). Eğitimde makine öğrenmesi: Araştırmalardaki güncel eğilimler üzerine inceleme. *Ahmet Keleşoğlu Eğitim Fakültesi Dergisi*, 3(2), 178-199.

Usmani, M., Adil, S. H., Raza, K., & Ali, S. S. A. (2016, August). Stock market prediction using machine learning techniques. In *2016 3rd international conference on computer and information sciences (ICCOINS)* (pp. 322-327). IEEE.

Uyar, U., Kelten, G. S., & Moralı, T. (2020). YATIRIMCILAR İÇİN TEKNİK ANALİZ: BITCOIN VE ETHEREUM UYGULAMALARI. *Finansal Araştırmalar ve Çalışmalar Dergisi*, 12(23), 653-671.

Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock closing price prediction using machine learning techniques. *Procedia computer science*, 167, 599-606.

Vui, C. S., Soon, G. K., On, C. K., Alfred, R., & Anthony, P. (2013, November). A review of stock market prediction with Artificial neural network (ANN). In 2013 IEEE international conference on control system, computing and engineering (pp. 477-482). IEEE.

Wang, J., Liu, J., & Jiang, W. (2024). An enhanced interval-valued decomposition integration model for stock price prediction based on comprehensive feature extraction and optimized deep learning. *Expert Systems with Applications*, 243, 122891.

Wöllmer, M., Eyben, F., Schuller, B., & Rigoll, G. (2011, May). A multi-stream ASR framework for BLSTM modeling of conversational speech. In 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4860-4863). IEEE.

Wu, J. M. T., Li, Z., Herencsar, N., Vo, B., & Lin, J. C. W. (2023). A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. *Multimedia Systems*, 29(3), 1751-1770.

Yadav, A., Jha, C. K., & Sharan, A. (2020). Optimizing LSTM for time series prediction in Indian stock market. *Procedia Computer Science*, 167, 2091-2100.

Yakut, Y. B. E. T. Y., Yakut, E., & Yavuz, S. (2014). Yapay Sinir Ağları Ve Destek Vektör Makineleri Yöntemleriyle Borsa Endeksi Tahmini. *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 19(1), 139-157.

Yang, S., Ding, Y., Xie, B., Guo, Y., Bai, X., Qian, J., ... & Ren, J. (2023). Advancing Financial Forecasts: A Deep Dive into Memory Attention and Long-Distance Loss in Stock Price Predictions. *Applied Sciences*, 13(22), 12160.

Yılmaz, E., & Aslan, T. (2016). Finansal risklerin yönetilmesinde türev ürünlerin kullanımı: Borsa İstanbul (BİST) 100 Endeksi'ndeki şirketler üzerine bir araştırma. *İşletme Araştırmaları Dergisi*, 8(1), 663-678.

Yiğiter, Ş. Y. (Ed.). (2024). Sosyal, Beşeri ve İdari Bilimler Alanında Uluslararası Araştırmalar XXIX. EĞİTİM YAYINEVİ.

Yoo, S., KYUNGJOO, S., & John Jongdae, J. I. N. (2007). Neural network model vs. SARIMA model in forecasting Korean stock price index (KOSPI). *Issues in Information Systems*, 8(3), 372-378.

Zahedi, J., & Rounaghi, M. M. (2015). Application of artificial neural network models and principal component analysis method in predicting stock prices on Tehran Stock Exchange. *Physica A: Statistical Mechanics and its Applications*, 438, 178-187.

Zhang, J., Ye, L., & Lai, Y. (2023). Stock price prediction using CNN-BiLSTM-Attention model. *Mathematics*, 11(9), 1985.

Zhao, C., Hu, P., Liu, X., Lan, X., & Zhang, H. (2023). Stock market analysis using time series relational models for stock price prediction. *Mathematics*, 11(5), 1130.

Zhong, X., & Enke, D. (2019). Predicting the daily return direction of the stock market using hybrid machine learning algorithms. *Financial innovation*, 5(1), 1-20.