Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Software Department

Data Analysis and Management of Business Intelligence Using Developed Deep Learning

A Dissertation

Submitted to the Council of the College of Information Technology at University of Babylon in Partial Fulfillment of the Requirements for the Degree of Doctorate of Philosophy in Information Technology/ Software

By

Ameer Alhaq Adil Sahib Ali

Supervised by

Prof. Dr. Eman Salih Al-Shamery

2022 A.C. **1443 A.H.**

بِسْمِ اللَّهِ الرحمن الرَّحِيمِ

﴿ قال اجعلني على خزائن الأرض إني حفيظ عليم﴾

صدق الله العلي العظيم

سورة يوسف\آية 55

Supervisor Certification

I certify that this dissertation was prepared under my supervision at the Department of Software / Collage of Information Technology / Babylon University, by **Ameer Al-Haq Adel Sahib** as a partial fulfillment of the requirements for the degree of **Ph.D. in Information Technology**.

Signature:

Name: **Dr. Eman Salih Al-Shamery**

Title: **Professor**

Date: / / 2022

The Head of the Department Certification

In view of the available recommendation, we forward this dissertation for debate by the examining committee.

Signature:

Name: **Dr. Ahmed Saleem Abbas**

Title**:** **Assist. Professor**

Date: / / 2022

Declaration

I hereby declare that this dissertation, **Data Analysis and Management of Business Intelligence Using Developed Deep Learning** submitted to University of Babylon in partial fulfillment of requirements for the degree of Doctorate of Philosophy in Information Technology-Software has not been submitted as an exercise for a similar degree at any other University. I also certify that this work described here is entirely my own except for reports and summaries whose sources are appropriately cited in the references

Signature:

Name: **Ameer Al-Haq Adil Al-Shamery**

Date: / / 2022

Acknowledgements

All praise be to ALLAH Almighty who helped me complete this task successfully and my utmost respect to his last Prophet Mohammad PBUH and to the hero of Islam born in the heart of Kaaba Amir-Al-Mo'mineen, Al-Imam Ali Ibn Abi Talib.

First, I would like to express my deep appreciation for the wonderful woman, my supervisor **Prof. Dr. Eman Salih Al-Shamery**, for her invaluable guidance, supervision and untiring efforts during the course of this work.

I must express my very profound gratitude to my mother and my wife for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this dissertation.

I would like to declare my deep thanks and appreciation to my sisters & friends for their patience, encouragement and help during the work.

Thanks and appreciation to the staff of the College of Information Technology, especially **Dr. Ahmed Habeeb Al-Azawi** and **Dr. Ahmed Al-Ajeli** for the great assistance they provided me.

Last but not least, most importantly, I would like to thank all the kind, helpful and lovely people who helped me directly or indirectly to complete this work and apologize to them for not being able to mention them by name here, though they are in my heart.

Ameer Al-Haq Adel

Dedication

To the savior of mankind …

Al-Imam Al-Mehdi (peace be upon him)

Who will fill the earth with justice and equity, after it has been filled with injustice and oppression. The survival of religion in the earth.

To My Father …

My soul, who left me alone for several years.

Who taught me that hard work does not take place except with patience and determination.

To my mother …

You have always been a source of kindness and compassion; I grew up on the shoulders of your patience.

To my wife and children

Who are the secret of my happiness in life.

Abstract

Developing and managing an effective investment portfolio and forecasting the stock market are one of the most complicated issues facing the financial sector today, especially in light of the increasing competition and global economic changes at the national and international levels. Thus, portfolio management and optimization is an important parts of the Business Intelligent (BI) system.

Although deep learning has addressed a broad range of issues in many fields, it still faces problems in certain areas of finance such as portfolio management. This dissertation, therefore, investigates the use of deep learning in solving the issues of portfolio optimization and forecasting stock prices. It also aims at understanding whether traditional portfolio theory principles should be totally replaced by a comprehensive integrated deep neural network or included in architectures.

To deal with such issues, this dissertation proposes an integrated deep neural network system to provide clear guidance for fund managers and individual investors. The proposed system can predict price in the stock market. It can also be used to select the right strategies for making investment decisions, and this, in turn, can enhance revenue during uncertainty periods. The proposed system includes three models namely, Deep Learning Classification (DLC), Deep Learning Regression (DLR), and Deep Portfolio Optimization (DPO). These three models are automatically applied from the input to the decision-making.

The DLR and DLC models improve the convergence speed in the training of full dense layers by using the quasi-newton method instead of gradient descent. Accordingly, these models can predict the closing price of the stocks and provide a decision to sell, buy, or hold stocks and the models can produce high accuracy in a reasonable time. DPO model is designed to address two limitations in the traditional portfolio theory which are the mixed-integer quadratic problem and the short-term trading. The first issue is solved by using the general reduced gradient optimizer, whereas the second is addressed by using the predicted expected return by the DLR model instead of the historical mean return. Moreover, such solutions are reconstructed as a deep learning model which has not been applied in earlier research.

Real global and local datasets which are the New York Stock Exchange (NYSE) and the Iraq Stock Exchange (ISX) are used to determine the behavior of the proposed system. Twenty-five well-known stocks are selected from the NYSE between 2010 and 2016 and forty-three stocks from ISX between 2013 and 2020. The ISX dataset is built and structured in this dissertation.

The experimental results support the robustness of the proposed models. Firstly, the DLC model proved its efficiency in terms of accuracy and execution time. Thus the accuracy rate reached 97.942% and 98.116% for the ISX and NYSE datasets, respectively. Secondly, the DLR model has also outperformed the prediction accuracy of earlier research based on the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) metrics as they reached 0.32094 and 0.49347 for NYSE, whereas both measures reached 0.084 and 0.1879 for the ISX dataset. Thirdly, the DPO model produced the largest Sharpe ratio, maximized profits, and reduced risks in the portfolio. It forms the portfolio with a Sharpe Ratio of 1.406 and 0.944 for the ISX and NYSE datasets, respectively.

The key advantages of the proposed system that relate to the practical aspect are: 1) employing the multithread principle in most layers of the deep neural network, 2) the application of this principle had an effective role in reducing the time taken from n to 1, 3) employing the second derivative method (quasi newton) in the prediction layer has the merit of reducing the convergence time for network training from 45 minutes to 0.75 seconds, and 4) the suggestion of an approximate activation function based on the principle of the Toiler series also led to a reduction in the execution time by almost 100%.

Table of Contents

Section No.	Subject	Page No.
	Chapter One : General Introduction	
1.1	Introduction	**1**
1.1.1	Stock Market Prediction	**2**
1.1.2	Portfolio Optimization	**4**
1.2	Problem Statement	**7**
1.3	Research Questions	**8**
1.4	Aims of Dissertation	**8**
1.5	Dissertation Contributions	**9**
1.6	Related Works	**10**
1.6.1	Related Work for Stock Market Prediction	**10**
1.6.2	Related Work for Portfolio Optimization	**15**
1.7	Dissertation Outline	**20**
	Chapter Two : Theoretical Background	
2.1	Introduction	**21**
2.2	Business Intelligence	**21**
2.3	Stock Markets	**23**
2.3.1	Stock Market Hypothesis	**23**
2.4	The Portfolio Management Problem	**26**
2.4.1	Portfolio Concept	**26**
2.4.2	Portfolio Theory	**26**
2.4.3	Diversification and Portfolio Principle	**28**
2.4.4	Risk and Return in MPT	**29**
2.4.5	Asset Return	**29**
2.4.6	Asset Risk	**30**
2.4.7	Portfolio Risk and Return	**31**

Section No.	Subject	Page No.
2.4.8	Efficient Frontier and Optimal Portfolio	33
2.4.9	Limitations of the MVO Model	33
2.5	The Research Dataset	34
2.5.1	The New York Stock Exchange	35
2.5.2	The Iraq Stock Exchange	35
2.6	Data Preprocessing	36
2.6.1	Interpolation	36
2.6.2	Technical Indicators Generation	37
2.6.3	Normalization or Standardization	40
2.7	Stock Market Prediction Techniques	40
2.7.1	Decision tree	41
2.7.2	K-nearest-neighbor	42
2.7.3	Naive Bayes	42
2.7.4	Artificial Neural Network	43
2.8	New Trends of Neural Networks	47
2.8.1	Deep Neural Networks	47
2.8.2	Optimization of ANN	49
2.9	Evaluation Measures	61
2.9.1	Performance Metrics	61
2.9.2	K-Cross Validation	63
2.9.3	Portfolio Metric Evaluation	64
Chapter Three : Design of the Proposed BI System		
3.1	Introduction	65
3.2	Proposed System Architecture	66
3.2.1	Preprocessing Stage	70
3.2.2	Prediction Stage	74
3.2.3	Evaluation Model Stage	89

Section No.	Subject	Page No.
3.2.4	Portfolio Construction and Optimization Stage	89
	Chapter Four : The Results and Discussion	
4.1	Introduction	94
4.2	System Requirement	95
4.3	Description of Stock Market Dataset	95
4.3.1	The New York Stock Exchange Dataset	95
4.3.2	The Iraq Stock Exchange Dataset	97
4.4	Preprocessing Stage	101
4.5	Evaluating the Classification Model	101
4.5.1	The Results of ISX	102
4.5.2	The results of the NYSE	112
4.6	Evaluating the Regression Model	118
4.6.1	The Results of ISX	118
4.6.2	The Results of NYSE	120
4.7	Evaluating The Portfolio Optimization Model	125
4.7.1	The Results of ISX	125
4.7.2	The results of NYSE	128
4.8	Discussion and Justification	130
4.8.1	Training Model	130
4.8.2	Technical Indicators	130
4.8.3	Activation function	131
4.8.4	The Prediction Results	132
4.8.5	Portfolio Optimization Results	133
4.8.6	Implications of theses	134
	Chapter Five : Conclusions and Future Works	
5.1	Conclusions	136
	Recommendations	137

Section No.	Subject	Page No.
5.3	Limitations	139
5.4	Future Works	140
References		141

List of Algorithms

Algorithm No.	Title	Page No.
3.1	Preprocessing stock market dataset	70
3.2	TIs Generation Layer	76
(3.3)	Standardization Layer	82
(3.4)	Optimization DLR model using BFGS method	86
(3.5)	Optimization DLC model using BFGS method	88
(3.6)	Deep Portfolio Optimization Layer	91

List of Procedures

Procedure No.	Title	Page No.
3.1	Interpolation (X, Y, r, period)	72
(3.2)	AROON (s, Close, i, k)	77
(3.3)	RSI (s, Close, i, k)	78
(3.4)	CCI (s, Close, High, Low, i, k)	79
(3.5)	MACD (s, Close, i, k)	80
(3.6)	%R (s, Close, High, Low, i, k)	81

List of Figures

Figure No.	Title	Page No.
(2.1)	The BI functions	22
(2.2)	Total portfolio risk	29
(2.3)	The correlation coefficient is between +1 and -1	32
(2.4)	Efficient frontier and optimal portfolio	33
(2.5)	Multilayer perceptron network	44
(2.6)	The linear activation function	45
(2.7)	The sigmoid activation function	46
(2.8)	The hyperbolic activation function	47
(2.9)	Deep learning in the context of artificial intelligence.	48
(2.10)	The loss function	49
(2.11)	One-dimension optimization	51
(2.12)	Steps of gradient descent	52
(2.13)	The search directions of Gradient descent	53
(2.14)	The steps of Newton's method	54
(2.15)	The search directions of Newton's method	55
(2.16)	The steps of conjugate gradient	56
(2.17)	The Quasi-Newton method	58
(3.1)	Overview of the proposed system	66
(3.2)	The architecture of proposed system	67
(3.3)	The structure of the proposed deep neural network	69
3.4	Nominal to Numeric	71
(3.5)	Interpolation	73
(3.6)	The topology DLR model	75

Figure No.	Title	Page No.
(3.7)	The topology DLC Model	87
(3.8)	Cross-Validation with 10-Folds for SM Dataset	89
(3.9)	The topology portfolio optimization layer	90
(3.10)	The characteristics of portfolio optimization in IDL system	92
(3.11)	The characteristics of enhancing accuracy in IDL system	93
(3.12)	The characteristics of enhancing execution time in IDL system	93
(4.1)	The NYSE historical close prices	96
(4.2)	The organizational view of ISX	97
(4.3)	The feature extraction step	98
(4.4)	ISX stock's closing Price	99
(4.5)	ISX60 closing Price	101
(4.6)	The bar chart accuracy of 43 stocks in ISX	110-111
(4.7)	The bar chart accuracy of 25 stocks on the NYSE	118
(4.8)	The bar chart MAE of 25 stocks on the NYSE	124
(4.9)	The bar chart MAE of 25 stocks on the NYSE	124
(4.10)	The weights of the assets in a portfolio for ISX	127
(4.11)	The weights of the assets in a portfolio for NYSE	129

List of Tables

Table No.	Title	Page No.
(1.1)	Summary of the related works.	**14**
(2.1)	Confusion Matrix	**61**
(3.1)	Standard features of SM	**71**
(3.2)	Fundamental features of SM	**73**
(4.1)	The New York Stock Exchange dataset	**96**
(4.2)	No. of trading stock in ISX	**100**
(4.3)	Evaluation criteria results of ISX60	**102**
(4.4)	Evaluation criteria results for ISX dataset	**104**
(4.5)	Analysis the results of Table 4.4	**105**
(4.6)	The DLC results with and without TIs generation layer for ISX	**106**
(4.7)	Analysis the results of Table 4.6	**107**
(4.8)	The DLC results with approximation activation function for ISX	**108**
(4.9)	The accuracy comparison results for ISX	**109**
(4.10)	Evaluation criteria results for NYSE dataset	**112**
(4.11)	Analysis the results of Table 4.10	**113**
(4.12)	The DLC results with and without TIs generation layer for NYSE	**114**
(4.13)	Analysis the results of Table 4.12	**115**
(4.14)	The DLC results with approximation activation function for NYSE	**116**
(4.15)	The DLC results with six activation function for JNJ stock	**117**
(4.16)	The DLC accuracy of the NYSE	**117**
(4.17)	The evaluation results of DLR model with and without TIs generation layer on ISX	**119**
(4.18)	Analysis the results of Table 4.17	**120**

Table No.	Title	Page No.
(4.19)	The evaluation results of DLR model with and without TIs generation layer on NYSE	121
(4.20)	Analysis the results of Table 4.19	121
(4.21)	The DLR results with six activation function for three stocks in NYSE	122
(4.22)	The Comparison results of RMSE and MAE for NYSE	123
(4.23)	The weights of the assets in portfolio for ISX	126
(4.24)	Sharp Ratio Results for ISX	126
(4.25)	Weights for stocks after applying three models for NYSE	128

List of Abbreviations

Abbreviation	Meaning
ACO	Ant Colony Optimization
AI	Artificial Intelligence
ANN	Artificial Neural Network
ASE	Amman Stock Exchange
BB	Bollinger Band
BFGS	Broyden, Fletcher, Goldfarb, and Shanno
BI	Business intelligence
BP	Back Propagation
BSE	Bucharest Stock Exchange

Abbreviation	Meaning
CCI	Commodity Channel Index
CG	Conjugate Gradient
CM	Confusion Matrix
CNN	Convolutional Neural Network
CV	Cross Validation
DLC	Deep Learning Classification
DLR	Deep Learning Regression
DM	Data Mining
DNN	Deep Neural Network
DPO	Deep Portfolio Optimization
DT	Decision Tree
EF	Efficient Frontier
EMA	Exponential Moving Average
EMH	Efficient Market Hypothesis
EPS	Earnings Per Share
EW	Equal Weighted
FN	False Negative
FP	False Positive
GA	Genetic Algorithm
GD	Gradient Descent
GRG	General Reduce Gradient
GWO	Grey Wolf Optimization

Abbreviation	**Meaning**
ISX	Iraqi Stock Exchange
KNN	K Nearest Neighbor
LR	Linear Regression
LSTM	Long-Short Term Memory
MACD	Moving Average Convergence Divergence
MAE	Mean Absolute Error
MCS	Monte Carlo Simulation
MIQP	Mixed-Integer Quadratic Problem
ML	Machine Learning
MLFF	Multi-Layer Feed Forward
MLP	Multilayer Perceptron
MPT	Modern Portfolio Theory
MSE	Mean Square Error
MVO	Mean Variance Optimization
NASDAQ	National Association of Securities Dealers Automated Quotations
NN	Neural Network
NSE	National Stock Exchange
NYSE	New York Stock Exchange
OBV	On Balance Volume
PO	Portfolio Optimization
PSO	Particle Swarm Optimization

Abbreviation	Meaning
QN	Quasi Newton
RMSE	Root Mean Square Error
RNN	Recurrent Neural Networks
ROA	Return On Asset
ROC	Rate of Change
ROE	Return On Equity
RP	Regression Predictor
RSI	Relative Strength Index
SD	Standard Deviation
SLFF	Single Layer Feed Forward
SM	Stock Market
SMA	Simple Moving Average
SVM	Support Vector Machine
SVR	Support Vector Regression
TI	Technical Indicator
TN	True Negative
TP	True Positive
TPT	Traditional Portfolio Theory
TR	True Range
TSF	Time Series Forecast
WMA	Weighted Moving Average

List of Dissertation Related Publications

(First Paper)

- **Name of Journal:** Karbala International Journal of Modern Science.
- **Paper Title:** Prediction of Iraqi Stock Exchange using Optimized Based-Neural Network
- **Scopus: Q1 site score 4**
- **Authors:**
    - Ameer Alhaq Adel Al-Shamery.
    - Prof. Dr. Eman Salih Al-Shamery.

Software Department, Information Technology College, Babylon University.

Email: ameeralhaq@itnet.uobabylon.edu.iq.

(Second Paper)

- **Name of Journal:** Periodicals of Engineering and Natural Sciences.
- **Paper Title:** A Superior Active Portfolio Optimization model for Stock Exchange.
- **Scopus: Q2 site score 2.3**
- **Authors:**
    - Ameer Alhaq Adel Al-Shamery.
    - Prof. Dr. Eman Salih Al-Shamery.

Software Department, Information Technology College, Babylon University.

Email: ameeralhaq@itnet.uobabylon.edu.iq.

(Third Paper accepted with minor correction)

- **Name of Journal:** CMC-Computers, Materials & Continua.
- **Paper Title:** An Integrated Deep Neural Network Model for Stock Market Prediction and Portfolio Optimization.
- **Clarivate: Impact factor: 3.772, Scopus: Q1 site score 4.6**
- **Authors:**
    - Ameer Alhaq Adel Al-Shamery.
    - Prof. Dr. Eman Salih Al-Shamery.

Software Department, Information Technology College, Babylon University.

Email: ameeralhaq@itnet.uobabylon.edu.iq.

Chapter One

General Introduction

General Introduction

1.1 Introduction

Business Intelligence (BI) is a system designed to empower better decision making. BI and its subsets, business analytics and data analytics, are data management solutions used to understand historical and contemporary data and create insights (Delen & Ram, 2018). Under the umbrella of BI, this dissertation is focused specifically on the concept of financial analysis. Which examines two separate issues in this area. The first issue is related to Stock Market (SM) prediction, and the second issue is related to portfolio management and optimization.

The SM environment is dynamic and noisy in behavior. Traditional methodologies, such as fundamental and technical analysis, may offer investors with tools for stock management and price forecasting. However, these strategies are unable to identify every conceivable relationship between stocks, necessitating a new method that provides a more in-depth examination.

On this foundation, the scientist Markowitz (Markowitz, 1952) built the Modern Portfolio Theory (MPT) to determine the relationship between stocks. Numerous developments have been made since then to improve the original techniques of portfolio optimization. The current dissertation focuses on building deep learning models that design the entire portfolio in a single step. Despite the fact that deep learning has helped address a broad range of issues in a variety of fields, it has yet to show itself in certain areas of finance, such as portfolio management. As a result, this dissertation investigates how to use deep learning to solve the issue of portfolio optimization, as well as whether traditional portfolio theory principles should be totally replaced by comprehensive Deep Neural Networks (DNNs) or included into architectures.

1.1.1 Stock Market Prediction

The SM prediction is a hot area of research because the connection between inputs and outputs are variable, nonlinear, and volatile in nature. Thus the SM prediction price still represents an issue in the financial time series that needs to be addressed (Ukrit et al., 2020). Choosing a suitable training and prediction method is also still a very critical problem (Navale et al., 2016). SM is an effective part of any country's economy. It plays a significant role in the financial and this, in turn, affects the country's economy growth. SM is a public place where companies are allowed to trade money through the sale or purchase of shares and stocks after determining the price agreed upon in advance (Ferni Ukrit et al., 2020).

The SM prediction is a process of concluding the future value of a stock's company based on its historical data. Two ways can be followed to meet this aim. First, fundamental analysis can be determined by mathematical data of a particular company (Edwards et al., 2018). Second, technical analysis can be conducted by Technical Indicators (TIs) and Machine Learning (ML) (Gosens & Lu, 2013). Numerical factors such as daily ups and downs, the volume of stock, tendency pointers, the highest and lowest prices of a day, directories, simple moving average, etc. can be used. Previous literature has attempted to discover some accurate arithmetic models that can allocate these inputs to predict the desirable outputs (Yefimochkin, 2011). Although there are many examples of these models, Deep Neural Network (DNN) technique have been widely adopted (LeCun et al., 2015).

The DNN has been used in several different areas such as financial forecasting, signal processing, and pattern recognition. It has been widely used due to its successful classification and regression in SM predictions (Abiodun et al., 2018). However, DNN algorithms may possibly fail to predict SM precisely. This is because it may be stuck in local minima, which can generally use learning rate to avoid this problem. On the other hand, choosing the incorrect value of the learning rate can lead to another problem. If the value of the learning rate is large, the overshooting problem will appear, whereas if its value is small, the convergence speed will be very slow (Pang et al., 2020). Based on this discussion, other several methods are available to address such issues in which the Quasi-Newton (QN) optimization approach is one of them. This approach depends on the second derivative to avoid falling into local minima. Moreover, learning rate parameters can be automatically adapted (Leonard & Kramer, 1990).

Building a predicting model that can consistently predict the future value of stocks with greater accuracy. It is expected that this can lead to better business decisions and earn money. To meet this objective, this dissertation improves the convergence speed in the training of deep multilayer perceptron that use the Back-Propagation (BP) algorithm, by using the QN method instead of Gradient Descent (GD). Accordingly, this dissertation designs a new Deep Learning Prediction models, which can predict the closing price of the stocks as well as giving a decision to sell, buy, or hold for stocks and produce high accuracy in a reasonable time.

1.1.2 Portfolio Optimization

Finance Portfolio Optimization (PO) is a subject that has been extensively researched in statistics, mathematics, financial, and computer science literature (Grinold & Kahn, 2000). It pertains to establishing the ideal mix of weights linked with a portfolio's financial assets (Zaman et al., 2020). PO is difficult in reality because of the wide range of parameters, mathematical formulations, business restrictions, and complicated financial instruments. Additionally, with the huge investment opportunities available in the business environment in general and the financial market environment in particular, the methods of selecting stocks and building investment portfolios are multiple and increasingly complex (Oztemel & Gursev, 2020). A portfolio is defined as a group of investment assets, which are selected based on specific constraints and economic objectives to achieve best profits for investors (Almahdi & Yang, 2017). Portfolio management, on the other hand, refers to the investment decision process based on tactical investment strategies that are tailored to match maximization of return for each investment time horizon (Zellweger, 2007).

Until the 1950s, investing in SM was a random process because there was no sufficient financial data and there was a lack of people's awareness about the concept of investment management. Investors had usually focused on studying the behavior of each stock separately, rather than investigating the relationship between the return and risk (Waweru et al., 2008). In 1952, however, this situation was fundamentally changed by the Nobel Prize winner Markowitz who proposed a new portfolio selection technique which is so-called “MPT” (Michaud & Michaud, 2008).

The Mean Variance Optimization (MVO) model of Markowitz is one of the most important theories in finance research (Levy & Levy, 2014). To reduce the variance of the investment portfolio in this model, the diversification principle is suggested. It assumes that investing in different types of stocks can reduce the possible risks in comparison to the investment in one stock only. The final stage of Markowitz model is the Efficient Frontier (EF). This key notion of the EF stage is selecting the efficient portfolio in line with the investors' preferences. The EF is represented by the curves of risks (variance) and return (mean) to determine the efficient portfolio (Bailey & Lopez de Prado, 2012).

Although the MVO model is important for profit maximization (Markowitz & Grossman, 1998), the PO problem based on the MVO model involves solving the quadratic programming problem for return and risk tradeoffs in order to achieve the EF. For this purpose, many metaheuristics proposed exact solutions to solve the PO problem. However, such approaches require extensive CPU time to find a solution (Skolpadungket et al., 2007). With real-life constraints, the problem is formulated as a Mixed-Integer Quadratic Problem (MIQP) and hence the problem belongs to the NP-Complete class of problems. To deal with this issue, it has been suggested that hybrid local search techniques may be used to reduce the computing complexity of this type of problem. Approaches based on heuristics have emerged as a viable alternative (Doering et al., 2019). Local search techniques seek for suboptimal answers to NP-Complete problems in a fair period of time rather than an ideal solution in an excessive amount of time.

Another issue with MVO model is that it is suitable for long-term investment only. The expected return for such portfolio models is usually estimated using the historical stock return mean (Waweru et al., 2008). However, it cannot deal with the short term trading. Recently, many types of ML models have been employed to forecast short-term financial markets (Michaud & Michaud, 2008). This may provide a potential path in predicting future returns. Thus, the benefits of ML approaches can be integrated with the modern portfolio theory to optimize contemporary portfolio models for short-term investing (Ta et al., 2020).

This dissertation, therefore, aims at predicting the expected return and optimizing portfolio. To achieve this aim, a new DNN topology is proposed. This topology includes two phases. The first is the prediction of the expected return. This phase encompasses three layers namely, the TIs generation layer, the standardization layer, and the prediction layer. The second phase is the optimization of the portfolio. The rationale behind the integration of the proposed topology is that earlier research relied on a manual data preprocessing (Boenninghoff et al., 2019), while the first and second layers of proposed topology automates this process. Moreover, traditional DNNs may be do not reach the optimal solution because of they are stuck in the local minimum (Bengio et al., 2005), whereas layer three addresses this issue by adopting the second derivative approach which is represented by the QN optimization method. Finally, although the key principle of DNN is either classification or regression, the proposed topology adapts DNN in optimizing portfolios. This is achieved by exploiting the General Reduce Gradient (GRG) method to update the layers' weights in order to maximize the objective function (Sharp Ratio) and this, in turn, leads to allocating these weights to asset in portfolios. To the best of the researcher's knowledge, this dissertation is the first that proposes such an integrated DNN topology that helps enhance the prediction of the expected return, maximize profits, and reduce risks.

Based on the above discussion, this dissertation is motivated by the lack of an integrated BI system in the Iraqi Stock Exchange (ISX). Moreover, the SM is one of the interesting areas at the moment that can affect the countries' economy. Finally, where and how people can invest their money is a problem that many investors still face.

1.2 Problem Statement

Investors in the financial market face multiple investment opportunities, in light of a large number of stocks and the corresponding multiple entrances to building investment portfolios. This makes the task of selecting best investment portfolio to be a difficult task, unless the investor identifies clear preferences regarding the expected return and the extent of avoiding the potential risks (Melicher & Norton, 2013).

Based on the above discussion, the Markowitz's model attempts to build an optimized portfolio and allocate weight for each stock to maximize the profit and reduce the possible risks (Markowitz, 1952). However, this model faces two problems which are the mixed-integer quadratic problem and the short-term trading (Akbay et al., 2020).

Previous literature attempts to address the issue of the mixed-integer quadratic problem by the heuristic optimization techniques (Dhaini & Mansour, 2021), whereas the second issue was addressed based on ML techniques to predict the future expected return and exploit it in the MVO model instead of the historical average return (Ma et al., 2020).

Earlier research, however, attempts to address either the first or the second issues. Regarding the use of heuristic techniques to solve the first Markowitz’s limitation, the execution time requires extensive CPU use to find a solution (Skolpadungket et al., 2007). The other issue of the MVO model is that it is suitable for long-term investment only using the historical stock return mean (Waweru et al., 2008). However, it cannot deal with the short term trading. Although machine learning techniques were used to address this issue (Michaud & Michaud, 2008), previous studies did not solve the first limitation of the MVO model.

1.3 Research Questions

The main question in this dissertation is how to build an integrated system that can improve the profit and reduce the possible investment risks. Based on this question, many sub-questions can be drawn:

- Is it possible to build an efficient portfolios using DNN?
- Can technical analysis and its mechanisms be implemented in building active portfolios that can outperform the corresponding Iraqi general index?
- How can stock prices be predicted with high accuracy based on DNN?
- Can DNN be used to recommend best decision to buy, sell, and/or hold stocks?
- How can the execution time of the proposed system be reduced?

1.4 Dissertation Aims

This dissertation aims to design an integrated DNN system to provide clear guidance for fund managers and individual investors. The proposed system can help choose the right strategies for making investment decisions and this, in turn, can enhance profits during the uncertainty periods and reduce risks. This is achieved through several sub-objectives:

- Building and optimizing an efficient portfolio using DNN.
- Building an active portfolio to outperform the Iraqi general index: This objective is achieved by designing the Deep Portfolio Optimization model.
- Improving the prediction of general index price, daily stock price, and daily expected return in the SM: This is met based on designing a Deep Learning Regression model.
- Increasing the accuracy in the decision-making process (selling, buying, and waiting) stocks: The Deep Learning Classification model is proposed in order to correspond to this objective.

- Reducing the execution time in the price prediction stage and the portfolio optimization stage: This is achieved by applying the optimization technique, the approximate activation function, and multi-thread principle.

1.5 Dissertation Contributions

This dissertation analyzes the financial sector based on portfolio management and the prediction of possible investment in SM using deep learning. Here, an integrated system is proposed and built to support BI decision-making. The proposed system includes three models namely, Deep Learning Classification (DLC), Deep Learning Regression (DLR), and Deep Portfolio Optimization (DPO). These three models are automatically applied from the input to the decision making, whereas previous work applies each step independently. Accordingly, this dissertation adds many contributions in comparison to previous literature.

1. It is the first research that built and organized a benchmark for the ISX. This depends on building a dataset for eighty-nine Iraqi companies that are categorized under eight sectors namely, Banks, Communications, Investment, Insurance, Services, Agriculture, Industry, and Hotels.
2. The prediction accuracy and convergence speed of neural networks are improved using QN instead of the traditional training method GD.
3. This dissertation is the first that contributes to the enhancement of the prediction performance of neural networks for fluctuation companies by generating technical indicators and adding them as a new layer to the neural networks based on the principles of statistical with neural networks.
4. It also proposes a new approximation activation function to reduce the execution time of the proposed model training. The produced function is used instead of the sigmoid activation function which provides an approximate accuracy in comparison with the sigmoid function, but with less execution time.

5. The traditional implementation of technical indicators, standardization, and prediction layers is sequential. This dissertation, however, uses the principle of parallel computing to synchronize the application of these three layers based on the multi-thread principle. This reduces the implementation time by n:1 in which n represents the number of stocks.
6. The dissertation also addresses two limitations in the MVO model which are the mixed-integer quadratic problem and the short-term trading. The first issue is solved by using the general reduced gradient optimizer, whereas the second is addressed by using the predict expected return instead of the historical mean return. Moreover, such solutions are reconstructed as a deep learning model which has not been applied in earlier research.
7. The outcomes of this dissertation should be significant for investors, fund companies, decision-makers because they can rely on such outcomes to make the right decision of investment.

1.6 Related Work

A review of previous studies applied to BI and in particular to SM is conducted. According to the issues addressed in this dissertation, this section is divided into two parts:

1.6.1 Related Work for Stock Market Prediction

The relevant papers are divided into several groups in terms of relevant datasets, inputs, pre-processing methods, techniques and methodologies, prediction methods, and evaluation criteria. The investigation also presented the limitations of most technologies used to predict SM. It is indicated that NNs have poor learning on data patterns or their performance may be unexpected or inconsistent due to the complexity of stock exchange data. As such, some of the previous literature that attempts to improve or outperform NNs for forecasting SM is presented.

1- Althelaya et al. (2018) evaluated and compared Long-Short Term Memory (LSTM) deep learning architectures specifically Stacked and Bidirectional LSTM with shallow NNs and unidirectional LSTM for long-term and short-term prediction of SM. The results showed that both Bidirectional LSTM and Stacked LSTM networks performed better in short-term price prediction than long-term forecasting. The outcomes also showed that deep learning is better than shallow NNs in predicting SM.

2- Zhang et al. (2018) designed a stock price prediction system based on a Random Forest model that predicts both the movement of the stock price and the rate of growth (or decline) at intervals within predefined forecast periods in the China market. The historical data of each stock was cut into multiple clips with a predetermined prediction length. This process was done by using a sliding window method. Then, the unsupervised heuristic algorithm was utilized to classify the shapes that appear in the close prices of these clips into four main classes which are Down, Up, Flat, and Unknown. The results showed an improvement in the forecasting of market volatility as well as better accuracy and return per trade.

3- According to Hiransha et al. (2018), a group of four kinds of DNNs named LSTM, Convolutional Neural Network (CNN), Recurrent Neural Networks (RNN), and Multi-Layer Perceptron (MLP) was implemented. Two stock markets were taken to check the performance of the price prediction. From National Stock Exchange (NSE) market, the collected dataset includes three various sectors: IT, Automobile, and Banking, whereas from the market of the New York Stock Exchange (NYSE), the Petroleum and Finance sectors were selected. The obtained results were compared with the Auto-Regressive Integrated Moving Average (ARIMA) model. It was observed that DNNs are superior to the current linear model ARIMA.

4- Alalaya et al. (2018) proposed a model based on the integration of NNs with Fuzzy logic ANFIS. The Amman Stock Exchange (ASE) index prices data were used in this dissertation. Different models were compared in which Relative Strength Index (RSI), Stochastic Oscillator (SO), Moving Average Convergence/Divergence (MACD), and On Balance Volume (OBV) as input indicators were used in this dissertation. The results proved the enormous capabilities of NN and fuzzy logic architecture such as ANFIS to implement time series forecasting as a task for forecasting the direction of stock prices in the ASE.

5- Ahmed et al. (2019) used Ant Colony Optimization (ACO) to train the NN instead of the Gradient Descent optimization in predicting stock prices in the Nigerian Stock Exchange. The ACO was also compared with three other algorithms which were Moving Average, Price Momentum Oscillator, and Stochastic. The dissertation findings reached the superiority of the ACO method in accuracy over the other methods implemented.

6- Sahoo & Mohanty (2020)evaluated and compared a hybrid model consisted of merging NN technology with Grey Wolf Optimization (NN-GWO) and a standard NN. These models have been applied to the Bombay Stock Exchange dataset in a period time from 2005 to 2018. After using different evaluation error measures, the results showed that the hybrid NN-GWO model is better than the traditional NN model.

7- Adnan & Isma'eel (2021) proposed a prediction model based on artificial NNs to predict stock returns for thirty-eight companies listed on the ISX during the period from 2010 to 2019. After training the network using the BP algorithm, they found a weakness in its performance and its inability to distinguish between stock returns data patterns when it was used as individual inputs to the network. The reason was that return rates had a large fluctuation, which led to a discrepancy in proportions in different directions, positively and negatively.

8- khalid Chyad (2021)applied the ARIMA model to predict the ISX Index. Statistical methods were also used such as arithmetic average, percentage change, T-test, scatter plot, Q plot, and the autocorrelation coefficient. The research dataset consisted of 102 companies listed in ISX. These companies were divided into seven sectors Communications, Banks, Insurance, Industry, Services, Agriculture, and Hotels. The study period was from 1/1/2019 to 12/31/2019. It was found that the model was very good for predicting ISX Index with accuracy and reliability level equal to (95%).

9- Mohamed et al. (2021) applied an adaptive fuzzy- neuro model to fifty-eight companies listed on the Dubai Financial Market and Abu Dhabi Securities Exchange for the period between 2014 and 2018. After examining four predictors: Return On Equity (ROE), P0ofit Margin (PM), Return On Asset (ROA), and Earning Per Share (EPS), the results showed that ROE was the most significant predictor and ROA was the least.

Table 1.1 briefly explains the techniques and methods, stock exchange data, periods, and evaluation metrics used in the previously reviewed literature. This leads to the conclusion that most of the earlier studies attempted to build a single model of prediction, either for classification or regression purposes. In this dissertation, however, the proposed model aims at predicting the real price of the stock (regression) as well as providing a decision to buy, wait, or sell (classification).

Table 1.1. Summary of the related works.

References	Prediction Techniques	Stocks/Index	Data	Period time	Accuracy (%)	RMSE (%)
Althelaya et al. (2018)	MLP, LSTM, Stacked LSTM (SLSTM), Bidirectional LSTM (BLSTM)	S&P 500	Historical daily stock prices (closing price)	01/01/2010 to 30/11/2017	-	Short-term: SLSTM: 0.01248 MLP: 0.03875 BLSTM: 0.00947 LSTM: 0.01582 Long-term: SLSTM: 0.06637 MLP: 0.09369 BLSTM: 0.06055 LSTM: 0.08371
Zhang et al. (2018)	unsupervised heuristic algorithm and Random forest(RF), SVM, ANN, k-NN	Shenzhen Growth Enterprise Market in China	prices and technical indices	January 25, 2010, to October 1, 2016.	RF: 72.2 SVM: 61.5 ANN: 57.0 k-NN: 43.9	-
Hiransha et al. (2018)	ARIMA, RNN, LSTM, CNN, MLP	National Stock Exchange (NSE) of India and New York Stock Exchange (NYSE).	day-wise closing price	NSE: 1 JAN 1996 TO 2015 June 30 NYSE: 3rd January 2011 to 30th December 2016	-	ARIMA: 19.64 RNN: 5.40 LSTM: 4.88 CNN: 4.05 MLP: 3.85
Alalaya et al. (2018)	ANN, ANFIS, Wavelet ANFIS	Amman Stock Exchange (ASE)	Index of Banking Sector	2000-2014	-	ANN: 0.0614 ANF1S: 0.0098 ANF1S with wavelet: 0.0721
Ahmed et al. (2019)	Ant colony optimization (ACO), Price Momentum Oscillator (PMO), Stochastic (St.), Moving Average (MA)	Nigerian stock exchange	Historical daily stock prices (closing price)	100 days	ACO: 0.812500 PMO: 0.677778 St.: 0.791667 MA:0.516854	-
Sahoo & Mohanty (2020)	ANN and grey wolf optimization (GWO), ANN	Bombay Stock Exchange (BSE)	Historical daily stock prices (closing price)	25 August 2004 to 24 October 2018	-	ANN: 0.539518 GWO: 0.534562
Adnan & Isma'eel (2021)	ANN	Iraq Stock Exchange (ISX)	Historical daily stock prices (stock return)	2010 to 2019	-	0.65
khalid Chyad (2021)	integrated regression model of moving averages (ARIMA)	Iraq Stock Exchange (ISX)	ISX60	(1/1/2019) to (1/31/2019)	95%	-
Mohamed et al. (2021)	ANN, Fuzzy logic	Abu Dhabi Securities Exchange and the Dubai Financial Market	Historical daily stock prices (closing price)	2014–2018	-	1.56

1.6.2 Related Work for Portfolio Optimization

Throughout the last several decades, different approaches and theories have been proposed to build PO models. Such models use a variety of ways to choose stocks, and then adopt an optimization model to evaluate the stocks in the portfolio to maximize earnings. Standard PO methods, in term of trading type, are often unsuitable for real short-term investing. As a consequence, it is required to develop a more efficient strategy that incorporates PO algorithms and predictive results of return. Here, previous literature has been classified into four categories based on the desired methodology for building and optimizing portfolios:

- **A Portfolios Construction Based on ML Predictive Performance**

In this category, earlier research used ML techniques to predict expected returns and then use the prediction accuracy to build a portfolio. Krauss et al. (2017) used a statistical arbitrage to implement and assess the performance of MLP, gradient-boosted tree, random forest, and certain ensembles models. Portfolios were constructed using the predicted outcomes of several models by the top k and the bottom k stocks. The findings demonstrated that the constructed portfolios using an equal-weighted ensemble model with MLP, gradient-boosted tree, and random forest provided daily returns of more than 0.45% before deducting the transaction fees. In another research study, Fischer and Krauss(2018) applied LSTM neural networks, random forests, MLP, and logistic regression for forecasting future stock returns. It was found that the LSTM neural network achieved higher performance in comparison with the other memory-free models. Another research study conducted by Yang et al.(2019) used an extreme learning machine to forecast future stock returns and then the stock scoring mechanism is built as a linear combination of the expected stock returns and fundamental factors using computational intelligence-based for optimization weights, and the top-ranked stocks are picked for an equally weighted portfolio. Lee and Yoo (2020) examined the stock return prediction performance of RNN, gated recurrent units, and LSTM neural networks. The results suggested that the LSTM neural network outperformed the other models. After that, a predictive threshold-based portfolio was proposed using the LSTM neural network's prediction outputs to build the portfolio.

The above discussed empirical studies are solely based on ML models for predicting stock returns and constructing portfolios. Such portfolio management techniques, however, are unable to balance the relationship between return and risk efficiently. Accordingly, the contemporary portfolio theory has been developed to address this issue. Thus, ML prediction models have been integrated with MVO for investing.

- **The Construction of Portfolios using ML Algorithms and MVO**

Another direction of research uses the MVO model to optimize portfolios based on expected historical results. Lin et al.(2006) proposed a dynamic portfolio model to address the portfolio selection problem. The Elman neural network was used to learn the dynamic SM behavior and forecast future returns, whereas the cross-covariance matrix was applied to calculate the covariance matrix of stocks. An optimal dynamic portfolio selection model was obtained following the proposed model. The results showed that this model outperformed the vector auto regression model and produced superior outcomes for the dynamic PO issue. Paiva et al.(2019)proposed a decision-making model for day trading SM investments. It was based on the fusion of the support vector machine (SVM) technique as a classifier and the MVO for portfolio selection. The proposed model was applied for the Random plus MVO and the SVM plus EQ. The model was assessed using the So Paulo Stock Exchange's index. Another research study used SVM to forecast expected returns (NekoeiQachkanloo et al., 2019). Fuzzy logic and the MVO model were used to design a portfolio. This portfolio can recommend how much an investor should put in different current equities. The proposed model can show an optimum expected return based on the degree of risk tolerance. The NYSE equities from 2010 to 2016 were chosen for the evaluation and comparison of the forecast. Ma et al.(2020) used three deep neural networks, namely LSTM, deep multilayer perceptron (DMLP), and CNN to construct a prediction-based portfolio optimization models. These models use both the modern portfolio theory and deep learning techniques. DNNs models were used to forecast the return of each stock. Subsequently, the predicted errors of DNNs were used to calculate the stock's risk. By adopting the semi-absolute deviation of predicting errors and predicted returns, PO models were created. The China Stock Exchange 100 Index was used to evaluate the proposed model. The findings indicated that DMLP outperformed the other models in forecasting the return of inventory. This is because it is more consistent with the characteristics of the entry after using multiple measures such as MAE, MSE, HR, HR+, and HR- for a complete assessment of capabilities. Based on LSTM network predictions of stock movement, Ta et al. (2020) created an effective portfolio by utilizing a variety of models, including Monte Carlo Simulation (MCS), Equal Weighted (EQ), and MVO. After applying the proposed model on the Standard and Poor 500 (S&P 500) benchmark, the results suggested an efficient prediction accuracy and portfolio optimization.

The MVO model assumes that the mean of historical returns is equal to its average and this, in turn, means that the historical returns follow a normal distribution. In reality, however, the historical return distributions often deviate from normality, showing kurtosis and skewness (Fama, 1965; KON, 1984).

- **The Portfolio Prediction based on Optimization models**

This direction of research was based on predicting PO according to the application of several different optimization techniques. Freitas et al.(2006) introduced a PO approach in which expected returns were predicted using autoregressive neural networks, whereas PO was performed using prediction errors. The suggested model achieved better results than the MVO model and showed a higher rate of return for the same risk. Hao et al.(2014) built a model for portfolio selection based on forecasting future stock returns. It uses Support

Vector Regression (SVR) and the variance of predictive errors as the risk for portfolio optimization. The proposed model was compared with (Freitas et al., 2006). The findings suggested that the improvement in forecasting accuracy of future stock returns can positively affect the performance of the PO model.

On the other hand, the problem of PO based on the classical model requires solving the quadratic programming problem for return and risk trade-offs. To tackle the PO problem, several heuristics and exact methods are introduced.

- **A Portfolios Construction Based on Optimization Models**

According to Maringer and Kellerer(2003), different techniques were investigated to overcome the computational difficulty of the above-mentioned issue. This includes recommending multi-objective heuristic and hybrid local search methods. Heuristic techniques have risen to prominence as a viable option. heuristics are becoming increasingly popular and are being used successfully in a variety of study domains (Doering et al., 2019). heuristic techniques for NP-Complete issues strive for suboptimal solutions in a fair amount of time rather than an ideal answer in an excessive amount of time. The research of Chen and Zhou(2019) looks at how the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) and the Genetic Algorithm (GA) might be used to optimize portfolios. MOEA/D and GA have both been successful in locating portfolios. The suggested framework is capable of successfully producing weighted portfolios. The portfolio optimization issue based on the extended MVO model utilized a new multi-objective algorithm based on Particle Swarm Optimization (PSO) and a suggested adaptive ranking mechanism (Silva et al., 2019). The suggested technique, dubbed Adaptive Ranking Multi-Objective PSO (ARMOPSO), was compared to many metaheuristic algorithms. The ARMOPSO's competitiveness in finding quality outcomes in the challenges studied was shown by comparative findings. Lim et al.(2020) developed an ensemble technique to identify optimum portfolios based on a GA. This approach is unique in that it employs several models, including MVO, Capital Asset Pricing Model (CAPM), and Momentum Strategy, all of which are optimized using a GA. The ensemble approach was examined using the Return on Investment over 5 years, and the findings were positive. Kalayci et al. (2020) proposed a hybrid metaheuristic algorithm based on important components from GAs, Continuous Ant Colony Optimization (CACO), and Artificial Bee Colony (ABC), with computational results demonstrating the hybrid algorithm's efficacy when compared to state-of-the-art algorithms. In (Akbay et al., 2020), the authors developed a unique hybrid strategy combining parallel Variable Neighborhood Search (VNS) with Quadratic Programming (VNS + QP).

The proposed model in this present dissertation, however, is developed based on several earlier studies (Alrabadi, 2016; Ma et al., 2020; NekoeiQachkanloo et al., 2019; Paiva et al., 2019; Ta et al., 2020). Although ML, particularly DNN techniques have been used to forecast expected returns and stock prices, they have not been utilized to enhance portfolio performance. Accordingly, this dissertation uses DNN to predict expected returns and improve portfolio optimization. It integrates DNN for prediction and optimization in which the DNN model encompasses four consecutive layers that are run automatically.

1.7 Dissertation Outline

The rest of the dissertation is structured as the following:

Chapter Two: presents an extensive description of the main principles of BI in the term of predictive analysis and portfolio optimization, which are used later in this dissertation.

Chapter Three: clarifies the main steps of designing an integrated Deep Neural Network (DNN) system for SM problem and its properties.

Chapter Four: shows the employment of the proposed system on real SM dataset and experimental results of this implementation.

Chapter Five: lists the derived conclusions of this dissertation and provides suggestions for future work.

Chapter Two

Theoretical Background

Theoretical Background

2.1 Introduction

This chapter introduces the theoretical concepts in this dissertation by presenting the concept of BI and discussing important portfolio improvement concerns. The PO model based on Markowitz's theory is discussed, along with its associated restrictions. Additionally, a variety of SM hypotheses are examined. Then, fundamental and technical analysis are presented to offer techniques for predicting the values of current stocks, specifically, DNNs. Furthermore, detailed descriptions of the dataset utilized for computational analysis in this dissertation are provided. Finally, assessment measures for forecasting stock prices are offered, as well as metrics for portfolio optimization.

2.2 Business Intelligence

In the quick development and change in the business environment, companies with the ability to exploit information about their marketplace, customers, company processes and suppliers will be able to capitalize business opportunities and maintain a competitive edge. Company's business operations can be effectively monitored and evaluated if it has access to information and an efficient data management system. Therefore, the concept of BI has grown, which includes the strategies and technologies used by enterprises for the data analysis and management of business information. In another meaning, the technique of understanding and visualizing data for the purpose of making meaningful business choices is known as BI (Starace, 2008). Common functions of BI technologies include online analytical processing, reporting, analytics, data mining, dashboard development, process mining, business performance management, complex event processing, text mining, benchmarking, prescriptive analytics and predictive analytics (Fombellida et al., 2020) as shown in Figure 2.1.

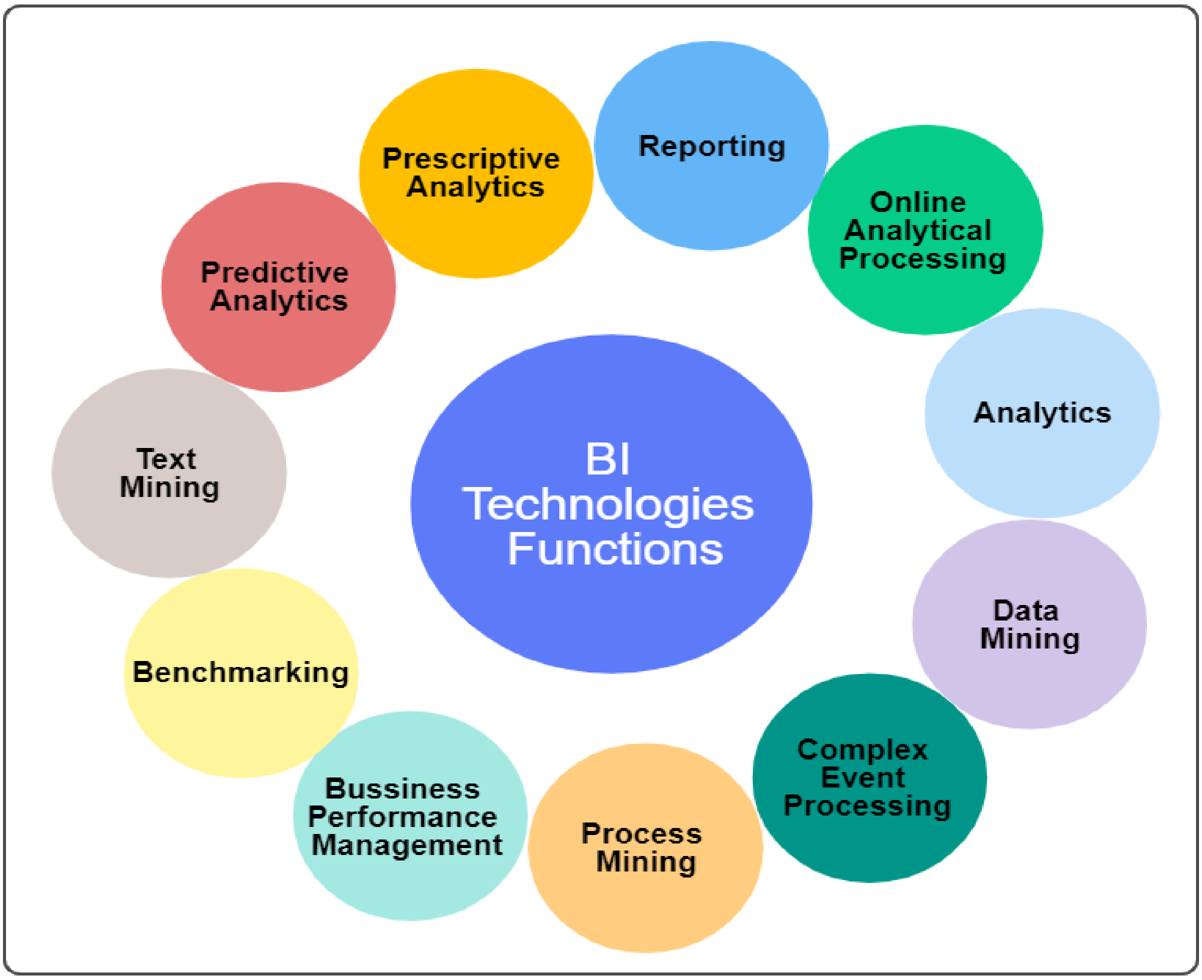


Figure 2.1. The BI functions

BI is defined similarly in the literature and by academics. In (Baschab & Piot, 2007), the authors describe BI as the capacity to deliver an information edge to the company; means that BI allows companies to accomplish what they have always done, but more efficiently. BI is regarded as a value proposition that enables organizations to have access to decision-making information that ordinary reporting does not deliver. BI has been defined by Singer (2001) as a set of techniques and tools were aimed to improving the quality of decision-making. BI encourages consistent decision-making and decreases the risk associated with discrepancies in the decision-making process (Rubin & Rubin, 2013).

Data exploration and analysis, dashboard monitoring, and sharing insights are the three main BI activities. However, There are also seven levels to the BI process: Information Technology (IT) infrastructure, data collection, data integration, data organization, data storage, data analytics, and data display (Glaser & Stone, 2008). Based on Figure 2.1, there are eleven functions of BI, this dissertation tries to design an intelligent trading system that employs most of the mentioned tasks except prescriptive analytics, text mining, and online analytical processing.

2.3 Stock Markets

Stock Market (SM) is a market in which financial securities, commodities, and other fungible things of value are traded at low transaction costs and at prices that reflect supply and demand (Aggarwal, 2017). Stocks and bonds are considered securities, whereas precious metals and agricultural items are considered commodities (Aggarwal, 2017). In economics, the phrase "market" refers to the aggregation of potential buyers and sellers of a certain commodity or service, as well as the transactions that occur between them. A securities market is used in an economy to attract fresh capital, convert physical assets to financial assets, establish a price that balances demand and supply and offer a way for short and long-term investment (Khanna & Palepu, 2010).

2.3.1 Stock Market Hypothesis

The efficient market hypothesis in finance has been disputed by economists who highlight psychological and behavioral components of stock-price determination and econometricians who assert that stock returns are predictably foreseeable to a large degree. In this context, many important market hypotheses have emerged in any financial analysis system such as:

- **Efficient Markets Hypothesis**

Efficient Market Hypothesis (EMH) confirms that market efficiency operations involve all related evidences, and it causes all available information to be absorbed at any given time (Laffont & Maskin, 1990). Moreover, the theory of market efficiency Fama can produce three levels: weak, semi-strong, and strong (Malkiel, 2003). These levels refer to the information set used in the statement "prices reflect all available information". The weak EMH assumes that weak stock prices already incorporate all historical data prices. The semi-strong level reflects current available public information and all historical data prices. The efficient strong market hypothesis is supposed to include private information that is hidden within the stock price. According to EMH principles, by including both public and private historical information side by side with the current price, it is difficult for investors to buy inexpensive equities on the SM or sell overpriced stocks. Therefore, it is difficult to beat the entire market by skilled stock selection or market timing, and the only way an investor can earn larger returns is by selecting riskier assets (Fama & French, 2021).

- **Random Walk Theory**

This theory depicts price prediction differently. Stock prices are inconsistence in this approach in which they are based on other features, although they have the same distribution. Thus, the past movements or trends in asset prices cannot be used to predict future behavior. Overall, random walk theoretical basis hypothesis of the theory looks like the semi-strong efficient market in which public information is supposed to reach to everyone. This theory says that costumer’s natural behavior of the market and the most possible one is a random walk, which would suggest that the predictions are not possible (Fama, 1965).

- **Analysis Philosophies**

Research currently revealed that the market prices do not follow the principle of random behavior and predicting financial market is possible (Klioutchnikov et al., 2017). In the world of finance, there are two major trading philosophies. The fundamental analysis philosophy is determined by mathematical data of a company. It may include sales data, financial status of the company, import/export volume, audit reports, strength and investment of company, plant capacity, the competition, terminal balance slips, and production indexes (Edwards et al., 2012). On the other hand, the technical analysis philosophy predicts future prices by using historical time series data. The favorite idea investors tend to follow is preceding investor behavior. Therefore, technical analyses are based on this idea. Comparing volumes and historical prices, average actions with the current volume and price can complete profitable opportunities. Technical analysis also generates TIs such as Moving Averages, Relative Strength, Rate of Change Index, Moving Average Convergence Divergence, Commodity Channel Index, etc. These indicators have helped traders to determine if the assets are oversold or purchase, or if the trend is weak or strong. Recently, TIs have been widely used in the prediction of systems as an input based on the ML techniques (Edwards et al., 2012). This technique knows how to identify data complex patterns and predict prices and trends in the future (Edwards et al., 2012).

- **Financial Forecasting using Machine Learning**

ML provides a unique opportunity for finance executives aiming to enhance their forecasting accuracy and procedures to fundamentally revolutionize financial forecasting. ML technology may considerably enhance the accuracy of predictions if it is properly applied. In addition, these technologies may be used to automate predicting models and conduct rapid calculations on massive dataset. By automating the work aspects of forecasting and enhancing forecasts, analysts are able to concentrate on providing more value to decision-makers.

2.4 The Portfolio Management Problem

The portfolio management issue and more precisely Harry Markowitz's Mean Variance Optimization (MVO) model is presented. Also an overview of the fundamental formulae that comprise the issue is shown. after that, the most important concepts, like as return and risk, are defined both for a single investment and for a portfolio of stocks in general. Additionally, the basic notion of diversification is discussed, how diversity impacts portfolio risk by examining the variables that necessitate the adoption of a diversified approach. Finally, the ideas of efficient portfolios and efficient frontiers is offered.

2.4.1 Portfolio Concept

The business field is full of many investment and financial activities, which contain multiple and diversified methods, including in particular investment portfolios in the financial markets. The concept of an investment portfolio is defined as a variety of financial and non-financial assets held by an investor. It can be defined more precisely as " a group of assets created for the purpose of achieving one or more investment objectives" (Norton & Tenenbaum, 1993).

Although many elements influence an investment portfolio, such as preferences, wealth, and experience, it is built in settings they are familiar with, beginning with stock selection, creating the portfolio, maintaining and styling the portfolio, and finally evaluating it (Grinold & Kahn, 2000).

2.4.2 Portfolio Theory

The investment portfolio and its technical procedures did not appear overnight but rather they are developed over time in tandem with the formation of financial markets and the development of investing strategies (Markowitz, 2010). The suggestions of Markowitz's in 1952 are credited with building the foundations of MPT (Elton & Gruber, 1997). This represents the most significant turning point.

The period before this was known as the period of Traditional Portfolio Theory (TPT).

TPT is a descriptive technique that relies on balancing the portfolio by acquiring a diverse range of equities or bonds for various industries and sectors. TPT emphases that on adding large and well-known firms' shares can perform well and reduce the risk. Moreover, because firms ‘securities are highly cash flow, the transaction costs can be lowered as well. It is worth noting that this is also due to huge investors buying these shares, which are then sought by small investors in the same way due to the herd effect. This technique is naive because it follows the adage principle "don't put all your eggs in one basket" and it does not focus as much emphasis on the link between stocks (Platanakis & Urquhart, 2020).

In 1952, however, this situation was fundamentally changed by the Nobel prize winner Markowitz who proposed a new portfolio selection technique which is so-called “Modern Portfolio Theory”, which is a quantitative and scholarly approach (Markowitz, 1952). The MVO model of Markowitz is one of the most important theories in finance research (Levy & Levy, 2014). This model clarified the way of measuring risk through the use of variance returns (Maringer & Kellerer, 2003). It is a moral and appropriate method and this, in turn, leads to its use not only to demonstrate portfolio diversification efficiency, but also to diversify the portfolio efficiently in the first place. According to Markowitz's suggestions, the strategy return-risk or mean-variance is based on "appropriate diversification", which is the practice of investing in stocks or assets with low joint variance (correlation) (Markowitz, 1952). The MVO model is based on a set of assumptions that explain investor behavior:

a) Investors take into consideration each investment alternative through the probabilistic distribution of its returns in a certain holding period.

b) Investors maximize their utility over a period of time, and their utility curves demonstrate diminishing marginal utility of wealth.

c) Investors estimate the portfolio's risk based on the change in expected returns.

d) An Investor bases his/her decisions only on the basis of the expected return and the expected risk (standard deviation) for returns only.

e) At a certain level of risk, the investor prefers the highest return over the lowest, and at a given level of return, the investor prefers the lowest risk over the highest.

2.4.3 Diversification and Portfolio Principle

Diversification is the working principle of MPT. The key concept of this principle is that the investor eliminates or at least limits excessive levels of risk in the assets which represents the shared concern of all types of investors. Although the risk is distinct from uncertainty, the financial literature uses the terms interchangeably. Thus, the risk is defined for the majority of investors as the condition of uncertainty about future returns, or the chance of earning returns that are less than anticipated and undesired (Abreu & Mendes, 2010).

There are two parts to investment risks. The first one is the systemic risk, which is also known as non-diversifying or market risk. This type represents the amount of risk that all assets in the market face at the same time, but at different rates in which it cannot be eliminated through diversification such as global financial crises and wars (Démurger et al., 2010). The second one is an unsystematic risk which is also called private, unique, or diversified risk. It affects the assets of certain organizations in the market as a result of emergency conditions such as worker strikes, technical failures, and other uncontrollable events. This type of risk can be avoided by diversifying across a group of assets within the portfolio. Accordingly, the total risk will be the sum of the two risks as shown in Figure 2.2 (Démurger et al., 2010).

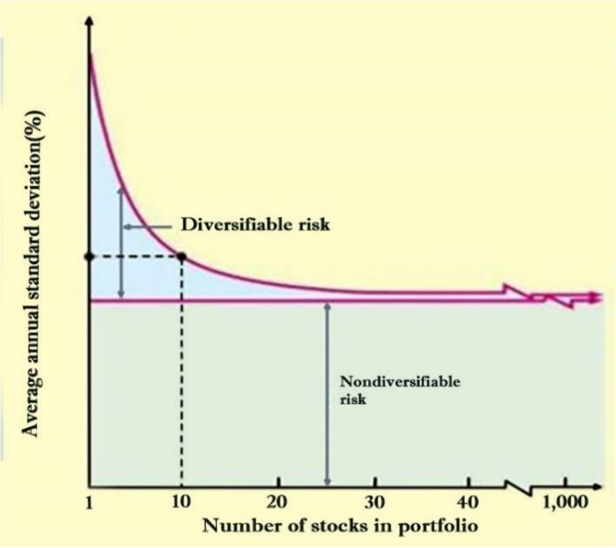


Figure 2.2. Total portfolio risk (Démurger et al., 2010)

2.4.4 Risk and Return in MPT

The process of assessing the portfolio's return and risk begins with an examination of the return and risk of individual securities and it concludes with the portfolio as a whole. As a result, after fulfilling the criteria for the portfolio model's inputs, the research will refer to how to calculate the return and risk of individual assets before moving on to the computation of the return and risk of the portfolio (Omisore et al., 2011).

2.4.5 Asset Return

The return is defined as the amount received by the investor during the period in which the asset is held, and it is the summation of the capital and periodic returns. As a result, it is computed based on a single holding period, and this form of return is known as the realized return (Diebold & Yilmaz, 2009), and it is calculated using Eq. 2.1:

$$\boldsymbol{R_i} = \left(\frac{\boldsymbol{P_1 - P_0}}{\boldsymbol{P_0}}\right) \tag{2.1}$$

Where:

P_0: The share's previous price when purchased

P_1: The current price or liquidation price.

As for the return used in the portfolio accounts, it is the expected return and it obtains from the asset in light of the current information on the price behavior of the asset. The expected return is computed by analyzing historical data for the asset's returns, in which the arithmetic mean of the asset's previous returns is calculated. Eq. 2.2 reflects the value of the expected return (Friedlob & Plewa Jr, 1996).

$$\boldsymbol{E(R)} = \left(\frac{\sum_{i=1}^{n} \boldsymbol{R_i}}{\boldsymbol{n}}\right) \qquad \textbf{(2.2)}$$

Where R_i is the return on the retention period, and n is the number of holding periods used to calculate the expected return.

2.4.6 Asset Risk

The risk is the degree of uncertainty about the expected output of the investment, and thus its metrics vary. Some measurements are used to measure the magnitude of the return and others are measuring the possibility of dropping below the projected value. Moreover, the standard deviation measure is the most prominent and extensively used metric in the portfolio and the rest of the asset pricing models (Chesney et al., 2020). It is one of the measures of statistical dispersion, since it reflects the level of dispersion of results from the anticipated value. The standard deviation of expected returns is obtained by using Eq. 2.3 (Ogneva, 2012).

$$\boldsymbol{\sigma^2} = \left(\frac{\sum_{i=1}^{n} (\boldsymbol{R_i} - \boldsymbol{\bar{R}})^2}{\boldsymbol{n-1}}\right) \qquad \textbf{(2.3)}$$

where: σ^2 is asset variance, R_i is historical returns of the asset, $\bar{R}$ is Historical average returns, and n is the number of historical outputs measured by year, month, or day.

2.4.7 Portfolio Risk and Return

The portfolio's return is determined using the weighted rate of expected returns for its various assets or investments, as shown in Eq. 2.4 (Fama & French, 2021).

$$\boldsymbol{E(R_P) = \sum_{i=1}^{n}(E(R_i).W_i)} \qquad \textbf{(2.4)}$$

where: $E(R_i)$ is the return asset *i* in the portfolio, and W_i is the weight of asset *i* in the portfolio.

Each asset has a specific weight when forming the portfolio, and the weight represents the value of the invested portion of the asset $\boldsymbol{i}$ relative to the total value of the portfolio as a whole. As demonstrated in Eq. 2.5, the total weights of the assets = 1. The weight of each asset is also calculated by the ratio of the value of the asset to the total value of the portfolio (Norton et al., 2019).

$$\boldsymbol{\sum_{i=1}^{n} W_i = 1} \qquad \textbf{(2.5)}$$

In contrast to the method of calculating portfolio return, the total risk of the portfolio varies greatly because it is not including only the weighted rate of risk (variance) of the assets, but also the common variance (Covariance) between assets. It plays a significant role in reducing risk and making it much less than the risk of the same assets if combined directly together, and this method of calculating risk is the most important product of MPT. The portfolio risk is computed by determining the variance of a portfolio of n assets, as shown in Eq. 2.6 (Norton et al., 2019).

$$\boldsymbol{\sigma_P^2 = \sum_{i=1}^{n} W_i^2\sigma_i^2 + \sum_{i=1}^{n}\sum_{j=1}^{n} W_iW_jCov_{ij}} \qquad \textbf{(2.6)}$$

where: σ_P^2 is the portfolio variance, W_i is the weight of asset i in the portfolio. σ_i^2 variance of asset *i*, and Cov_{ij} is the covariance between returns.

The significance of Markowitz's propositions in calculating risk based on variance is that the covariance has been considered the most important component, which indicates the degree of two variables move together relative to the single values of their averages over time and it is calculated in two ways: the first way is based on historical data as show in Eq. 2.7, whereas the second way is by Eq. 2.8 (Norton et al., 2019).

$$\boldsymbol{Cov_{ij}} = \left(\frac{\sum(\boldsymbol{R_i}-\overline{\boldsymbol{R_\iota}})(\boldsymbol{R_j}-\overline{\boldsymbol{R_J}})}{\boldsymbol{n-1}}\right) \quad \textbf{(2.7)}$$

$$\boldsymbol{Cov_{ij}} = \boldsymbol{\sigma_i\,\sigma_j\rho_{ij}} \quad \textbf{(2.8)}$$

where: $\sigma_i\,\sigma_j$ is The standard deviation of asset *i* and *j* respectively, and ρ_{ij} is the correlation coefficient of the assets *i* and *j*.

As a result, the research concludes that the correlation (the correlation coefficient) is the most essential component in the covariance combination since it measures the degree of the link between two things. The correlation coefficient plays a role in the variance law of the portfolio by reducing the covariance limit (it affects the effectiveness of diversification in the portfolio and the correlation ranges between (+1 and -1)). If its value is (+1), this indicates perfect correlation, both assets are travelling in the same direction in terms of return movement, rendering diversification ineffective and wasteful, because it will not help risk reduction. If it is (-1), the returns are inversely proportional, and diversification appears in its most efficient form. In principle, the portfolio's risk is (zero), as illustrated in Figure 2.3 (Robertson, 1959).

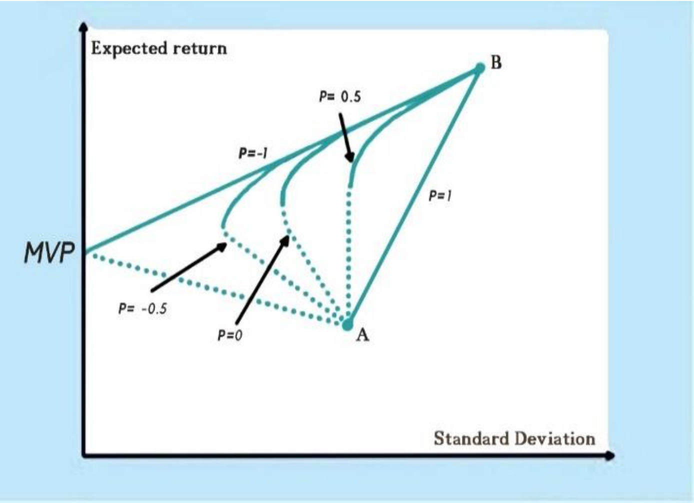


Figure 2.3. The correlation coefficient is between +1 and -1 (Robertson, 1959)

2.4.8 Efficient frontier and optimal portfolio

In final stages, the efficient portfolio theory is embodied by selecting the efficient portfolio according to the investor's preferences. The efficient portfolio can be represented in the space (return - risk) by drawing the horizontal axis representing risk and the vertical axis representing expected return. For example, the research used several assets, and it is clear from Figure 2.4 that the curve (BYOZC) represents the efficient frontier, which contains efficient portfolios and various asset combinations, all of which provide the highest possible return at a certain level of return or the lowest risk at a certain level of risk (Zhou & Li, 2000).

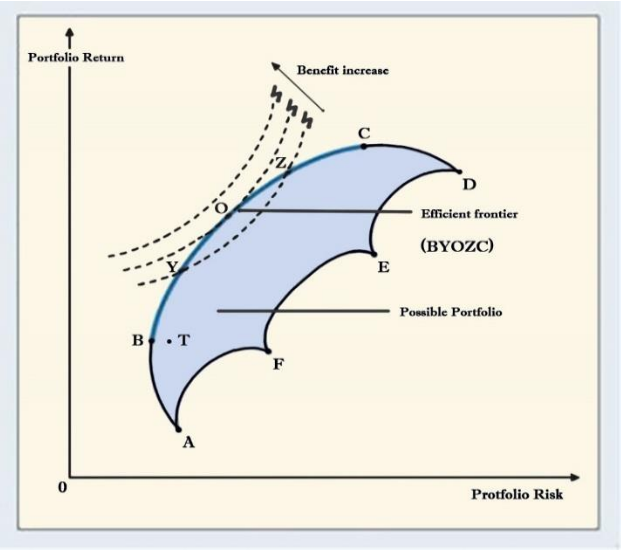


Figure 2.4. Efficient frontier and optimal portfolio (Zhou & Li, 2000)

2.4.9 Limitations of the MVO Model

Since MVO inception, the majority of investment businesses and mutual funds throughout the world have used the MVO approach as their principal instrument for portfolio diversification. However, despite its widespread use, the MVO model has a number of major limitations that restrict its usefulness. The primary drawbacks of the MVO model are briefly described as follows (Gökgöz & Atmaca, 2012):

1- The MVO approach implies that investors concentrate on a single time horizon and will never modify their asset allocation once it is decided.

2- It is difficult to determine the return of the market portfolio or even to represent it accurately.

3- Even if the efficient portfolio is determined, the transaction cost will be very high and small investors cannot afford it.

4- The market's structure and forces change dynamically, causing the risk-free return to fluctuate, followed by the market return, and finally the efficient limit, making the efficient portfolio unstable and thus difficult to obtain. This is why some refer to it as the (Holy Grail), implying that everyone wants it but it is difficult to obtain.

5- The statistical methodologies employed to calculate the return and risk have a statistical error rate that renders the findings completely inaccurate.

6- With large portfolios, the portfolio input under Markowitz's propositions becomes extremely complicated, as owning a portfolio of N assets necessarily requires N returns, N standard deviations, and (N(N-1)/2) common variances, resulting in data entry confusion and the occurrence of clearly incorrect results.

2.5 The Research Dataset

In finance, SM data consists of the price and other relevant data for a financial instrument, as supplied by a trading venue such as a stock exchange. The SM data enables investors and traders to see the current price and historical trends for assets such as stocks, fixed-income securities, currencies and derivatives. Historical data for global and local stock exchange indices and different companies can be used in this dissertation.

2.5.1 The New York Stock Exchange

The NYSE is a stock exchange in New York city that is the world's biggest by market capitalization of its listed stocks. Previously, a private company, the NYSE became a public company on March 8, 2006, when it acquired electronic trading exchange Archipelago. In 2007, a merger with Euronext—biggest Europe's stock exchange—created NYSE Euronext, which was eventually bought by Intercontinental Exchange, Inc. (ICE), the New York Stock Exchange's current parent company. There are 82 % of S&P 500 companies listed on the New York Stock Exchange (NYSE), as well as 70 of the top companies on the planet. There are over nine million business equities and investments exchanged every day on these publicly listed corporations (Wood et al., 1985; Pandey & Kumari, 2020).

2.5.2 The Iraq Stock Exchange

The Baghdad stock exchange is established in 1991 by the Iraqi government which started and shared trades for the period from 1992 to 2003. This market is managed by the Ministry of Finance in Iraq. Therefore, it is considered as a governmental market which indicates as a part of the Iraqi government policy. This market becomes more attractive in the last years where about 113 listed Iraqi companies, are distributed between the private sector and the mixed private-public sector (Asaad, 2014).

The Baghdad stock exchange reaches more than seventeen million dollars as an annual average. On March 19, 2003, the Iraqi government decided to close this market before the start of the war operations of the international coalition against the Iraqi regime (Hyndman & Koehler, 2006).

The ISX is established and the governors according to order seventy-four known as the Stock Exchange Law formed the Board of Directors of the ISX. The market started its activity in trading operations in June 2004 under the direct supervision of the Securities Commission in Iraq. This is an independent institution that was formed as a self-regulating organization which is similar to the New York Stock Exchange, owned by fifty or so members of the brokerage (Annual reports of the Iraq Stock Exchange, 2004-2020) (Talmon, 2013).

2.6 Data Preprocessing

Data-preprocessing techniques are used to process data that has low quality (García et al., 2015). Data-preprocessing techniques are classified into two kinds; The first kind concerns with cleaning data from noisy, lost and redundant data, while the second kind relates to feature construction by generation of technical indicators, data reconstruction by features normalization, and select the best features and eliminate irrelevant features by feature selection techniques (García et al., 2015). This dissertation employs most techniques of these two kinds because they can improve the efficiency of the forecasting process.

2.6.1 Interpolation

Linear interpolation is the simplest type of interpolation; it simply links each data point with a straight line to account the missing values. Linear regression (LR) is one of the most known methods. It is a statistical technique for determining the relationship between two variables by fitting a linear equation to the data. The explanatory variable and the dependent variable are both referred to as variables. LR models may, for example, be used to relate the weights of individuals to their heights. The explanatory variable $\boldsymbol{X}$ and dependent variable $\boldsymbol{Y}$ are both represented by Eq. 2.9.

$$\boldsymbol{Y = a + bX} \tag{2.9}$$

The linear regression line is customary to use $\boldsymbol{b}$ to represent the slope and $\boldsymbol{a}$ to represent the intercept when describing a line (Browne, 1975).

2.6.2 Technical Indicators Generation

The key notion behind the technical indicator refers to the results of the mathematical computation that applied on charts to direct investors towards long or short trading on different time frame configurations. This can confirm price movement and allow investors to forecast the development of equities (Wu et al., 2018).

There are two kinds of TIs, which are leading and lagging. Leading TIs are designed to track a market's price fluctuations and provide prediction capabilities. A guiding signal is created prior to the reversal or establishment of a new trend (Gosens & Lu, 2013). On the other hand, lagging TIs are constructed to track price movement. They are based on trend TIs, which are provide a signal after the initiation of trends (Gosens & Lu, 2013). In this dissertation, three leading TIs and two lagging TIs are used to generate five TIs features to be the input with the standard features to the standardization layer (layer two). The three leading TIs features are:

- **Relative Strength Index (RSI)**

RSI is a momentum indicator with a range of 0 to 100. The RSI indicator is used to measure the magnitude of recent price changes to evaluate overbought or oversold conditions in the price of a stock or other asset (Luo et al., 2011) as shown in Eq. 2.10.

$$RSI = 100 - \frac{100}{1+RS} \tag{2.10}$$

where:

$$RS = \frac{EMA\ (U,n)}{EMA\ (D,n)}$$

$$U = C_i\ (t) - C_j\ (t - 1)$$

$$D = C_i\ (t - 1) - C_i\ (t)$$

where U: trading periods in upward change. **D:** trading periods in downward change. **C_i (t):** close price of time (**t**). and Exponential Moving Average (EMA) can be calculated according to Eq. 2.11:

$$\boldsymbol{EMA = K\,(C_j - EMA\,(t - 1)) + EMA\,(t - 1)} \qquad \textbf{(2.11)}$$

where:

$\boldsymbol{K = 2/((1+k))}$

*And **k** is the time period*

Traditional interpretation and usage of the RSI is that values of 70 or above indicate that a security is becoming overbought or overvalued and may be primed for a trend reversal or corrective pullback in price. An RSI reading of 30 or below indicates an oversold or undervalued condition.

- **Commodity Channel Index (CCI)**

From the statistical techniques that is used to determine the market's cyclical direction, the CCI estimates the variation of the market with a range of [-∞ to ∞]. The CCI indicator measures the difference between the current price and the Simple Moving Average (SMA) price. This is followed by a scaling factor based on the mean absolute deviation of this price as illustrated in Eq. 2.12 (Schlossberg, 2006).

$$\boldsymbol{CCI = \frac{Ci(t) - SMA_{Ci}(k)}{(0.015)\,\sigma_{mad}\,(Ci)}} \qquad \textbf{(2.12)}$$

Where $\boldsymbol{\sigma_{mad}}$ is the mean of absolute deviation $\boldsymbol{C_i}$ and $\boldsymbol{k}$ is the time period.

- **Williams %R**

Williams % R (O'Grady et al., 1989) is used to illustrate the connection between the current price and the high and low prices during the previous n-day time period with a range of 0 to -100 . The % R is calculated as in Eq. 2.13.

$$\boldsymbol{\%R = \frac{Hi(k) - Ci(t)}{Hi(k) - Li(k)}} \qquad \textbf{(2.13)}$$

Where **H_i (t):** high price of time (**t**) and **L_i (t):** low price of time (**t**).

The other two lagging TIs are the moving average convergence divergence (MACD) and AROON.

- **Moving Average Convergence Divergence (MACD)**

From this indicator, it is possible to derive three different signals. The first is MACD which is defined as the difference between 26-day and 12-day EMA with a range of [-∞ to ∞]. The MACD's 9-day EMA is the second indication (Signal). The last signal is the histogram, which represents the difference between the signal and MACD. When the Moving Average (MA) of the MACD varies from one period to the other, and the indicator refers to the price oscillator. These signals are presented in equations (2.14) to (2.16), and these provide information about direction, momentum, and strength respectively (Appel, 2005).

$$MACD = EMA\ (12) - EMA\ (26) \qquad \textbf{(2.14)}$$

$$Signal = EMA\ (MACD,\ 9) \qquad \textbf{(2.15)}$$

$$Histogram = MACD - Signal \qquad \textbf{(2.16)}$$

- **AROON**

AROON assesses if a stock is going uptrend or downtrend as well as the strength of the trend with a range of 0 to 100. In the present dissertation, the price is estimated based on the highest and lowest prices over the previous fourteen days. AROON is calculated based on equations 2.17 and 2.18 (Gold, 2015):

$$AROON_{up} = ((14 - H_j\ (14))/14) * 100 \qquad \textbf{(2.17)}$$

$$AROON_{down} = ((14 - L_j\ (14))/14) * 100 \qquad \textbf{(2.18)}$$

2.6.3 Normalization or Standardization

Normalization is necessary to avoid a difference with the large values that dominate the calculation results. The term standardization is used to make a normal variable (Gaussian). The goal of normalization or standardization is to ensure that a whole set of values have a certain property. A traditional example is "standardize a variable" in statistics. If **μ** is the average value of the attribute values, and **σ** is the standard deviation, Eq. 2.19 generates a new variable X' with a mean value of zero and a standard deviation of one (Farrell & Saloner, 1985).

$$X' = \frac{(x-\mu)}{\sigma} \qquad \textbf{(2.19)}$$

2.7 Stock Market Prediction Techniques

The prediction task is a two-step procedure that begins with the construction of the prediction model and ends with the prediction of the target characteristic for the provided data (Nazerfard & Cook, 2015). Much effort is being done to forecast future stock price movement using artificial intelligence approaches. In general, this dissertation primarily focuses on two types of these techniques which are classification and regression.

The input data for a prediction task is a collection of records. Each record is represented by a tuple (**X, y**), where **X** is the attributes set and **y** is the target of prediction. The type of target attribute **y** detects the required technique; if values of **y** are binary, binary-class classification techniques can be used, multi-class classification is used if **y** has more than two categories. If values of y are continuous, regression techniques can be used for predictive task (Tan et al., 2016).

Here, there are three possible outputs which are:

- 1 denotes an upward moving future price.
- -1 shows a downward trending future price.
- 0 denotes no trending future price in the SM classification issue.

To make a decision, investors may look at the results and determine whether they want to sell, buy, or keep the stock. Although this method does not offer the error margin for the predicted future price increase or decrease in comparison to the preceding price, it is widely used. However, determining whether to sell or buy a stock is difficult without knowing the whole change margin. Hence, regression may be used to classify data (Tan et al., 2016).

Forecasting raw price values by regression is known as regressionism. It is possible to predict how much a stock's price will rise or fall in the future. Given a current index price of P_t, and it is required to know the future price of that index at time t+1. This scenario may be done by using prior index values P_{t-n}, P_{t-n+1}, ..., P_{t-1}, P_t. The prediction function is expressed in Eq. 2.20.

$$\mathbf{P_{t+1} = f\,(P_t, P_{t-1} \ldots P_{t-n})} \qquad \mathbf{(2.20)}$$

Several familiar ML algorithms could be used to predict a given problem with a set of features (Tan et al., 2016).

2.7.1 Decision Tree

Decision Tree (DT) is a commonly classification technique, which depends on creating a structure as tree with each branch representing an association between the values of feature and a class label. The results are very interpretable because DT generates rules, which easy to understand, but the outcomes are represented in categorical data. Hence, DT is less efficient for prediction when the features are numerical (Panigrahi & Mantri, 2015).

The most famous typical amongst DTs is the C4.5 tree. The mechanism of C4.5 tree is done by recursively partitioning the training dataset according to tests on the possible feature values in separating the classes. The most important question is how can the most important features be selected first? entropy or information gain is used to select the most important features, which is automatically considered feature of the lowest entropy or of the highest information gain (X. Chen & He, 2015).

2.7.2 K-Nearest-Neighbor

The K-Nearest-Neighbor (KNN) method is introduced in the early 1950s. It is applied in the most common applications named pattern recognition. There are many ML methods, nevertheless lazy learning algorithm is the simplest one because it does not need for any model in training. Therefore, KNN is one of the lazy learning types, which predicts classes of entity based on the K nearest training instances in the feature space. The KNN classifiers are based on learning work by calculating the similarity between the training objects and a specific test object. Furthermore, the training objects consist of N features. Each object defines a point in *n*-dimensions space. To classify an unidentified object, a KNN classifier observes for the *K* training objects that are nearest to that object in the *n*-dimensions space. This *K* training objects represent the KNNs of the unidentified example (Han, Kamber, et al., 2011).

To classify a KNN entity, the majority votes need to be gathered from the object's immediate neighbors. In this strategy, determining the proper value of K is considered a challenge.

2.7.3 Naive Bayes

The Naive Bayes Classifier is one of the unpretentious probabilistic models that depends on the Bayes Rule along with a powerful assumption of independence. It involves simplifying the assumption of conditional independence. This means that negative or positive classes are conditionally independent from each other. For its simplicity, this model is named “naive”. The accuracy of the classification is not greatly affected by this assumption, but it makes the classification algorithms fast and applicable to the problem (Orsolic et al., 2017).

2.7.4 Artificial Neural Network

Artificial Neural Networks (ANNs) are a kind of computer model that consists of a network of linked nodes or neurons that is designed to simulate the network found in the brain. Due to its capacity to learn complicated patterns, ANNs are commonly utilized to solve difficult issues. Such networks are robust against noise, ambiguity, and missing data, as well as capable of learning and adapting to its environment (Han, Kamber, et al., 2011).

In ANNs, the training is classifying into two broad categories: supervised and unsupervised training. With the assistance of supervised training, the NN determines the intended output and the synaptic weight that is modified in such a manner that the desired and estimated outputs are as similar as feasible (e.g. multilayer feed forward NNs). Unsupervised training implies that the network is given a collection of facts (patterns) and then allowed to settle down (or not) to a stable state after a certain number of repetitions ((Kohonen, 1988;Svozil et al., 1997).

ANNs are classed according to their architecture as Single Layer Feed Forward (SLFF), Multi-Layer Feed Forward (MLFF), or Recurrent in which their essential notion is fairly similar. Each input neuron is capable for receiving and processing signals before transmitting them into an output signal. Each neuron is connected to at least one other neuron, and each connection can be measured using a real number called the synaptic weight, which represents the significance of the connection in NNs (Wu & Banzhaf, 2010).

An example of MLFF is Multi-Layer Perceptron (MLP) approach, which is composing of neurons that have differentiable functions. The network involves one or more hidden layers. They consume a high degree of connectivity which is determined by synaptic weights to analysis a large part of the difficulty of these networks as a non-linearity and high connectivity of their neurons (Haykin, 1994).

Figure 2.5 depicts the three main layers in the MLP. The input layer (1, 2.,…., n) is the first layer that inserts a signal to feed the network nodes. The hidden layer (1, 2, …, h) is the second layer, whereas the output layer (1, 2…,0) is the third one. The complete calculations of the system are done in the hidden layer. The results for that network are obtained from the third layer.

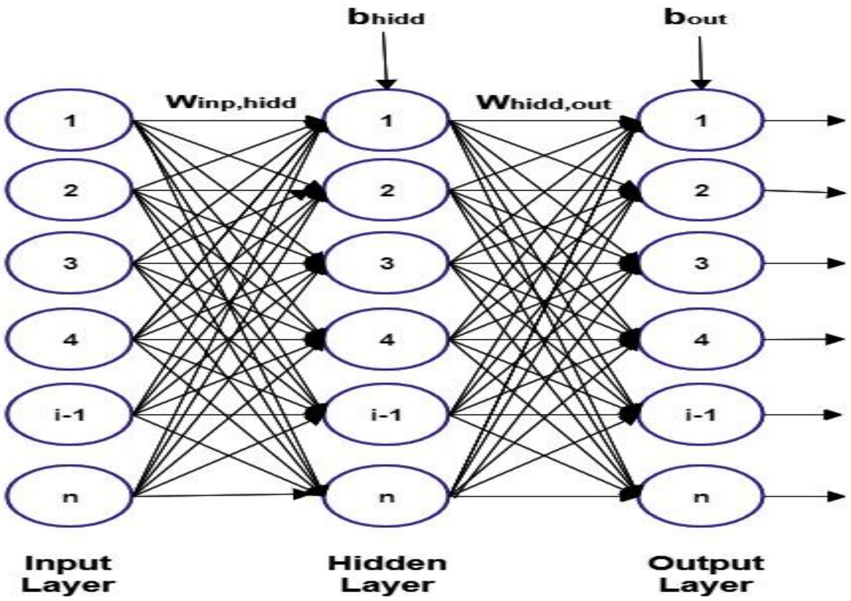


Figure 2.5. Multilayer perceptron network (Chakraborty, 2012).

The network contains bias nodes (b_{hidd}) and (b_{out}). These are linked with the hidden layer and output layer respectively. Moreover, the network contains synaptic weight values $W_{in,hidd}$ which is the assigned among the hidden layer and input layer of that network, whereas $W_{hi,\ out}$ is the synaptic weight values assigned among the output layer and a hidden layer of the network. A supervised learning method called the back propagation algorithm is often used to train MLP. The overall network error was controlled and reduced by adjusting these weight values (Chakraborty, 2012).

The back propagation algorithm has two main tasks which are the computation of feed forward and error back propagation to minimizing the total error learning of the network. During work, back propagation algorithm controls error by correcting all values of synaptic weight (Riedmiller & Braun, 1993).

An activation function ***f*** is a mathematical operation that is performed on the signal. The activation functions are determined in accordance with the nature of the issue that the network attempts to solve (Nwankpa et al., 2018). The most frequently used activation functions are as follows:

i. **Linear activation function:** The output of a linear activation function perceptron is just the combination of that neuron's inputs as shown in Figure 2.6.

$$f\,(x) = \textit{combination} \qquad \textbf{(2.21)}$$

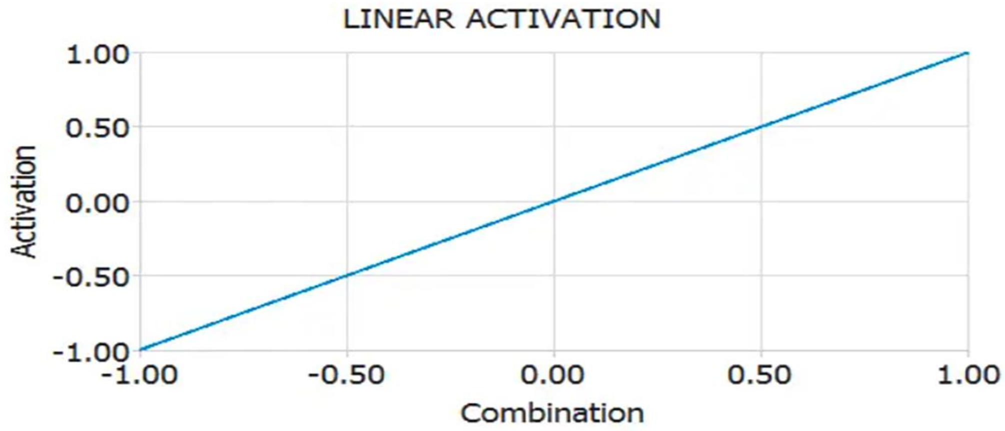


Figure 2.6. The linear activation function (Nwankpa et al., 2018)

ii. **Sigmoid function:** it is one of the most widely used types of activation functions in neural network modelling. It is a strictly growing function with a combination of linear and nonlinear behavior (Nwankpa et al., 2018).

– **Bipolar**: The term "bipolar" refers to the fact that neurons generate both positive and negative responses. The output range is [-1,1], which may be computed as follows:

$$f(x) = \frac{2}{1 + e^{-(x)}} - 1 \qquad (2.22)$$

– **Unipolar**: The term "unipolar" refers to the fact that neurons generate solely positive responses as presented in Figure 2.7. The output range is [0,1], which may be determined as follows:

$$f(x) = \frac{1}{1 + e^{-(x)}} \qquad (2.23)$$

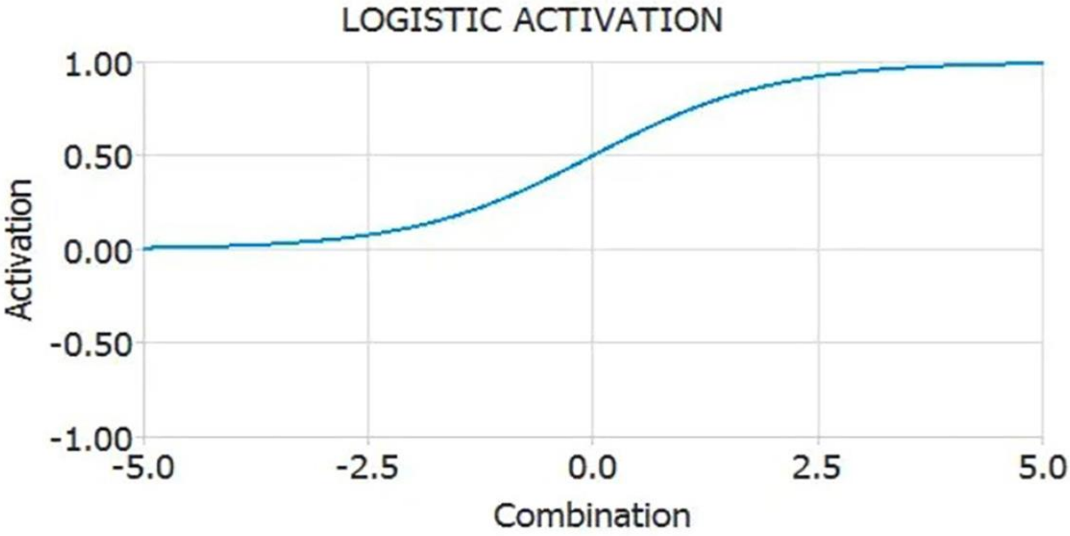


Figure 2.7. The sigmoid activation function (Nwankpa et al., 2018)

iii. **Hyperbolic functions:** They have names that are similar to those of trigonometric functions, but are defined in terms of the exponential function (Nwankpa et al., 2018). These functions are classified into three categories. *f(x)* equals sinh *x*, cosh *x* and tanh *x*.

f(x) = tanh *x* is implemented in this dissertation as viewed in Figure 2.8.

i. **Tanh x**: This function was pronounced 'tansh' or 'than' in speech. The formula that defines the function is as shown in Eq. 2.24.

$$f(x) = \tanh x = \frac{\sinh x}{\cosh x} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \qquad (2.24)$$

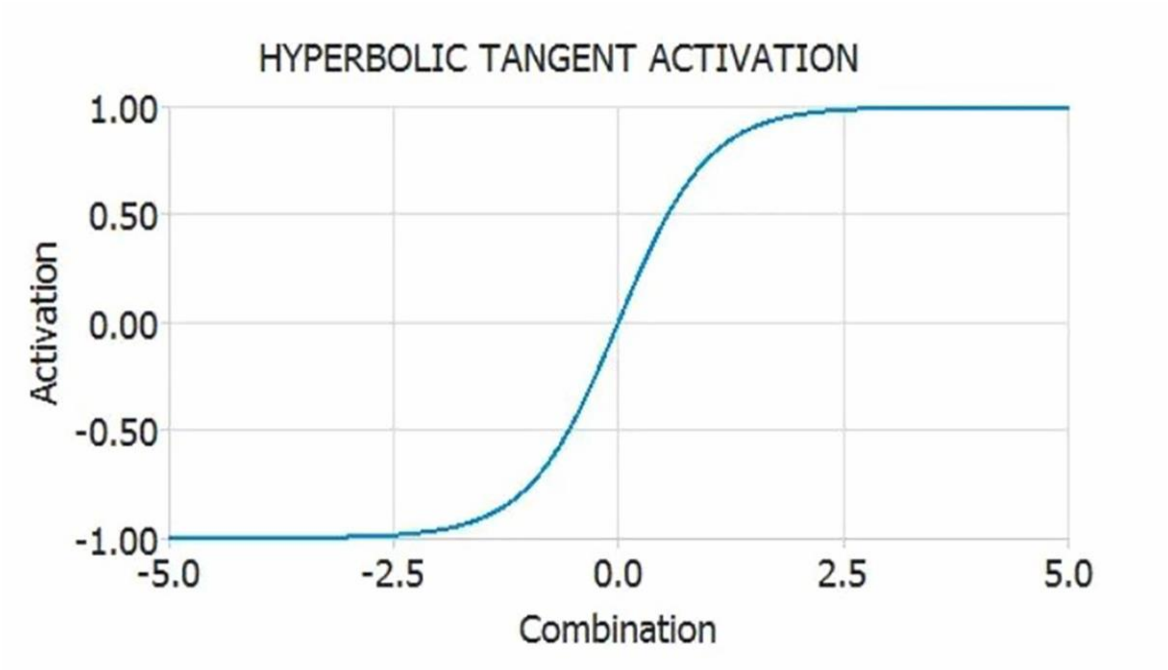


Figure 2.8. The hyperbolic activation function (Nwankpa et al., 2018)

2.8 New Trends of Neural Networks

There are several methods that can be applied to improve these weighting vectors by reducing the overall network error.

2.8.1 Deep Neural Networks

According to Figure 2.9, deep learning is a ML method involving the use of DNN under the umbrella of artificial intelligence (Sze et al., 2017). Just as the human brain consists of nerve cells or neurons which process information by sending and receiving signals, the deep neural network learning consists of layers of 'neurons' which communicate with each other and process information (Aggarwal, 2018). The 'deep' in Deep Learning points out the number of layers within the network; more the number of layers, deeper is the network.

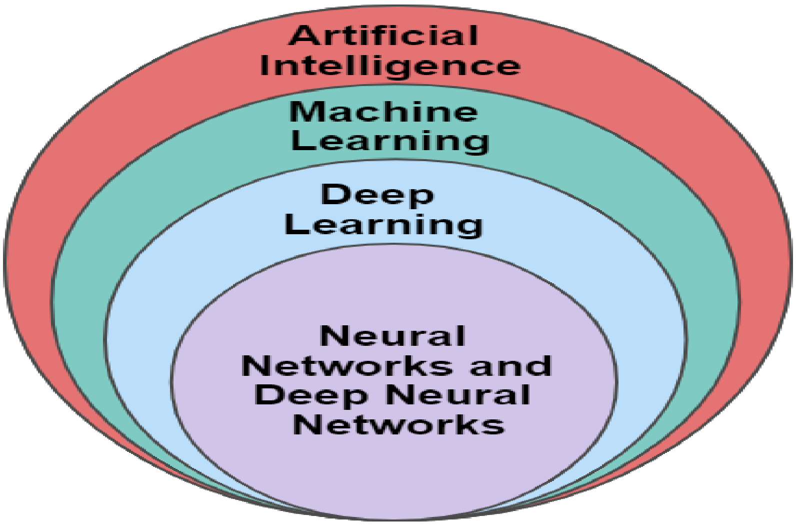


Figure 2.9. Deep learning in the context of artificial intelligence.

DNNs serve as the basis for the majority of contemporary artificial intelligence applications (LeCun et al., 2015). DNNs have been applied in many applications such as voice-recognition (Deng et al., 2013) and image-recognition (Krizhevsky et al., 2012). In several of these categories, DNNs can currently outperform humans in terms of accuracy (Sze et al., 2017). DNNs' exceptional performance is a result of its capacity to extract high-level features from raw sensory data using statistical learning on a vast quantity of data to get an accurate representation of an input space. This differs from previous techniques that used hand-crafted features or rules created by professionals.

Hence, returning to finance, artificial intelligence in general is heavily used in the financial business. For instance, it assists identifying issues such as unexpected debit card usage or large transfers to the account by using DNN. Thus, DNN as a concept contributes to the protection of individuals from fraudulent operations. Additionally, DNN is utilized to facilitate and improve trading by facilitating more structured and rapid decision-making based on a variety of market parameters (Kraus & Feuerriegel, 2017).

As for the stock market, DNNs are chosen over other statistical and ML approaches because of their representation-learning capabilities. The transformations of raw input data can help the process of learning complex functions (LeCun et al., 2015). For example, CNN automatically extracts features during such transformations in order to map input data to output data. This adaptability may also be used to extract complicated patterns from nonlinear SM data. Features may be broken down into numerous levels, which correspond to various layers of a DNN architecture. These levels reflect abstract features obtained from the previous levels and increase abstraction. With each additional layer, a deep network may extract more features than a shallow network with only a single hidden layer (Sun et al., 2014; Yong et al., 2017). In order to discover the underlying trends in a volatile SM, it is essential to use an appropriate data format. Thus, the adaptation of DNNs for nonlinear financial markets is the key driver behind the selection of DNN integration with SM data.

2.8.2 Optimizing ANNs

The issue of learning is defined as the minimization of a loss function ***f***. It is a function that determines how well a neural network performs on a given dataset (Aggarwal, 2018). In general, the loss function consists of an error and a regularization term. The error term quantifies the fit of a neural network to a data collection. By limiting the model's complexity, the regularization term avoids overfitting (Wang et al., 2016). The loss function depends on the neural network's adaptive parameters (biases and synaptic weights). They can be concatenated into a single n-dimensional weight vector **w**. Figure 2.10 illustrates the loss function *f(w)*.

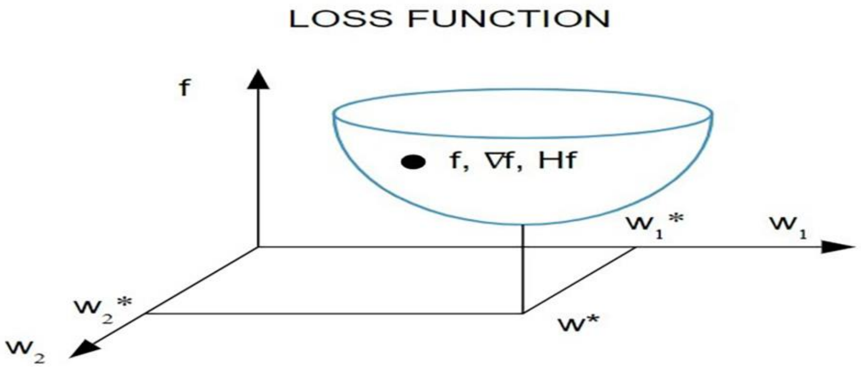


Figure 2.10. The loss function (Aggarwal, 2018)

As can be seen, the loss function's minimum occurs at the position $\mathbf{w}^*$. The research can compute the loss function's first and second derivatives at any point $\boldsymbol{A}$. The gradient vector is used to group together the first derivatives (Aggarwal, 2018).

$$\boldsymbol{\nabla}_{\boldsymbol{i}}\boldsymbol{f}(\boldsymbol{w}) = \frac{\boldsymbol{\partial f}}{\boldsymbol{\partial w_i}} \qquad \textbf{(2.25)}$$

where i=0,1,2, …

In a similar fashion, the Hessian matrix identifies the second derivatives.

$$\mathrm{H}_{\boldsymbol{i,j}}\boldsymbol{f}(\boldsymbol{w}) = \frac{\boldsymbol{\partial^2 f}}{\boldsymbol{\partial w_i \partial w_j}} \qquad \textbf{(2.26)}$$

where i, j=0,1,2, …

Significant attention has been paid to the issue of minimizing differentiable and continuous functions of a large number of variables. Numerous established approaches for this problem may be simply applied to neural network training (LeCun et al., 2015).

i. One-dimension optimization techniques

Despite the fact that the loss function is dependent on several factors, one-dimensional optimization techniques are crucial in this case. Indeed, they are utilized often in the training of neural networks. Several training methods firstly compute a training direction **d**, and then the learning rate **η** that minimizes loss in that direction, **f(η)**. This one-dimensional function is shown in Figure 2.11.

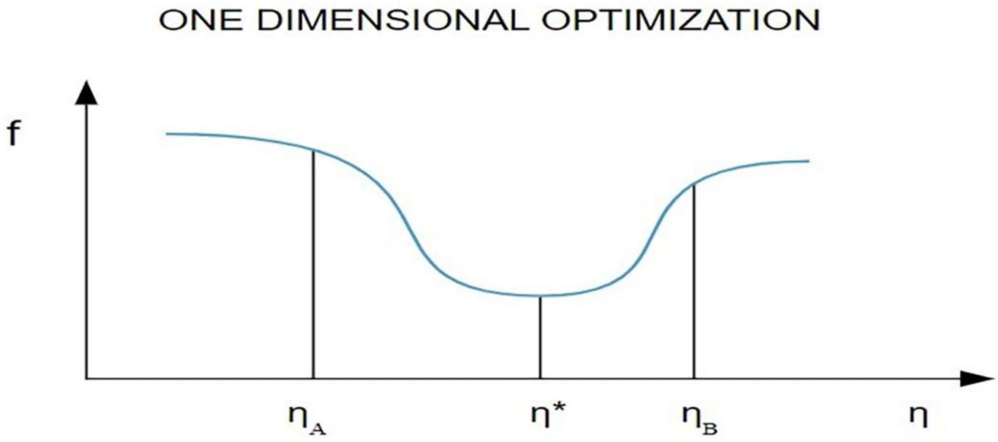


Figure 2.11. One-dimension optimization (Aggarwal, 2018)

An interval that includes the minimal value of *f*, η^*, is defined by points η_A and η_B. In this respect, one-dimensional optimization techniques seek the lowest one-dimensional functions. Brent's and golden section approach are the most prevalent as both are lower the minimum bracket until the distance between the two extremities falls below a certain tolerance (Johnson et al., 2010).

ii. Multi-dimension optimization

Based on the main concepts of neural networks, the search for the least loss function value in the parameter vector w^* is referred to as learning. It is essential that the gradient vector be zero if the neural network's loss function is reduced. For the most part, the loss function is a nonlinear function. As a consequence, it is difficult to identify closed training strategies for minima. Instead, in this dissertation, a step-by-step exploration of the parameter space is applied. At each level, the neural network parameters may be changed to minimize the loss. To training a neural network, a parameter vector is often chosen at random. The next step is to iteratively refine the settings in order to reduce the loss function. It is the difference in losses between two phases. The training process ends when a certain condition or stopping criterion is met (LeCun et al., 2015).

- **Gradient Descent (GD)**

GD is the simplest approach for training, it is a first-order approach since it needs information from the gradient vector.

Assume that $f\left(w^{(i)}\right) = f^{(i)}$ and $\nabla f(w^{(i)}) \ = g^{(i)}$. The procedure starts at $w^{(0)}$ and progresses in the training direction $d^{(i)} = -g^{(i)}$ from $w^{(i)}$ to $w^{(i+1)}$ until a stopping requirement is fulfilled. As a result, the GD algorithm iterates as illustrated in Eq. 2.27 (Ruder, 2016).

$$\boldsymbol{w^{(i+1)} = w^{(i)} - g^{(i)}\eta^{(i)}} \qquad \textbf{(2.27)}$$

where i=0,1,2, …

The factor η denotes the learning rate. This value may be set to a constant value or determined at each step using one-dimensional optimization along the training path. At each successive step, it is typically desired to obtain an optimal value for the training rate by line minimization. Nevertheless, a large number of software applications continue employing a set value for the training rate (Ruder, 2016).

Figure 2.12 depicts the activity diagram for the GD training process. As can be seen, the method optimizes parameters in two stages: It begins by determining the GD training direction, then it determines an appropriate training rate.

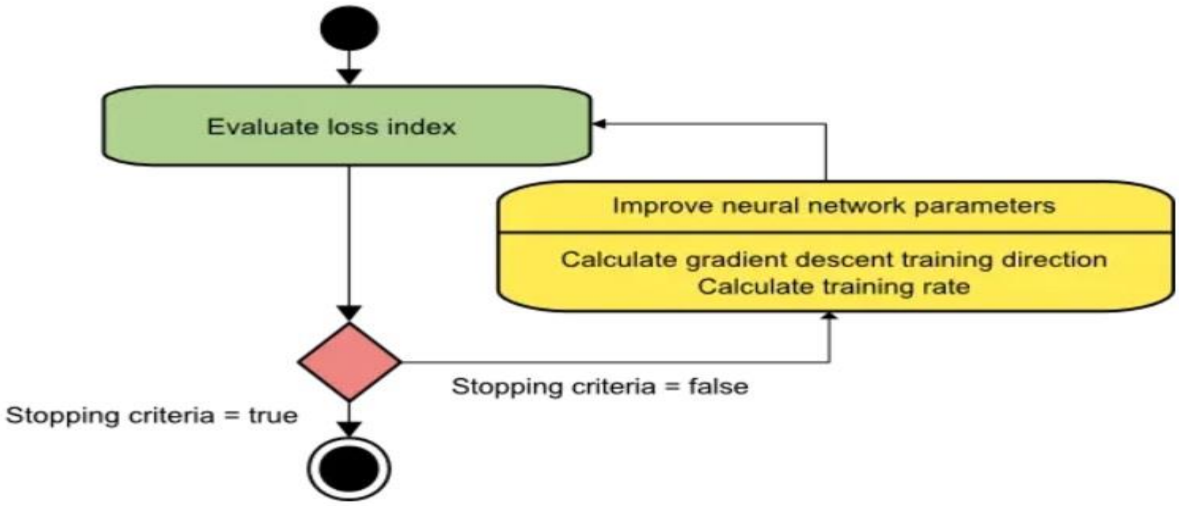

Evaluate loss index
Improve neural network parameters
Calculate gradient descent training direction
Calculate training rate
Stopping criteria = false
Stopping criteria = true

Figure 2.12. Steps of Gradient descent (Ruder, 2016)

In functions with long and narrow valley structures, GD training has the drawback of requiring an excessive number of iterations. Despite the fact that the loss function drops the fastest on a downhill gradient, this does not necessarily lead to the fastest convergence. This argument is shown in Figure 2.13 (Du & Swamy, 2013).

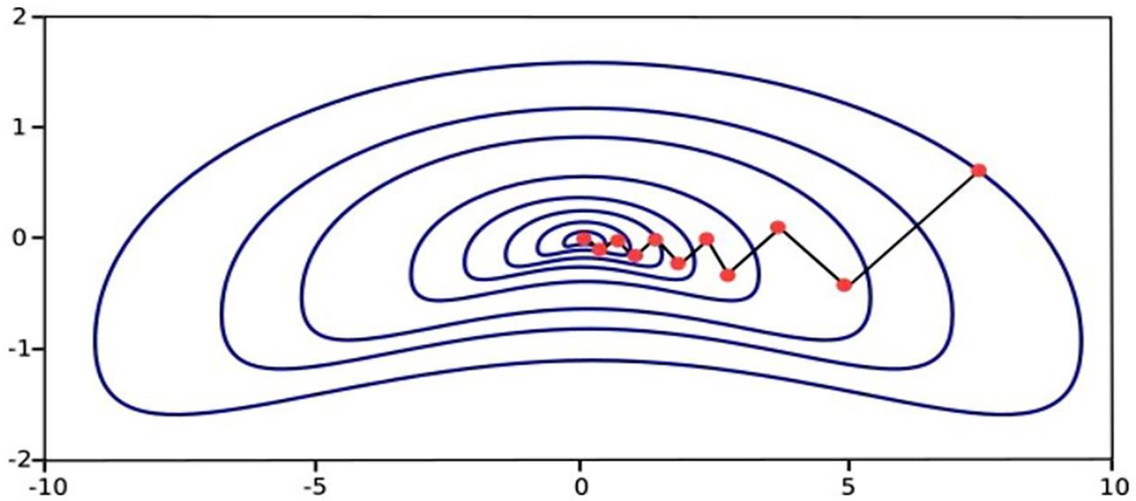


Figure 2.13. The search directions of Gradient descent (Du & Swamy, 2013)

When dealing with huge NNs with thousands of parameters the use of GD is recommended. The reason for this is because this approach saves just the gradient vector (size n), not the Hessian matrix (size n^2).

- **Newton's method**

Hessian matrices are used in Newton's second-order algorithm. The second derivatives of the loss function are used in this strategy to identify better training directions (Moré & Sorensen, 1982).

Let $f(\boldsymbol{w}^{(i)}) = f^{(i)}, \nabla f(\boldsymbol{w}^{(i)}) = g^{(i)}$ *and* $\mathbf{H}f(\boldsymbol{w}^{(i)}) = \mathbf{H}^{(i)}$. Consider the quadratic approximation of f at $\boldsymbol{w}^{(0)}$ by the use of the Taylor's series expansion.

$$\boldsymbol{f} = \boldsymbol{f}^{(\mathbf{0})} = \boldsymbol{g}^{(\mathbf{0})}.\left(\boldsymbol{w} - \boldsymbol{w}^{(\mathbf{0})}\right) + \mathbf{0.5}\,.\,(\boldsymbol{w} - \boldsymbol{w}^{(\mathbf{0})})^{\mathbf{2}}.\,\mathbf{H}^{(\mathbf{0})} \qquad \textbf{(2.28)}$$

$\mathbf{H}^{(0)}$ denotes the Hessian matrix of f at the location $\boldsymbol{w}^{(0)}$. By setting g to zero for the minimal value of $f(\boldsymbol{w})$, the following equation is obtained:

$$\boldsymbol{g} = \boldsymbol{g}^{(\mathbf{0})} + \mathbf{H}^{(\mathbf{0})}.\left(\boldsymbol{w} - \boldsymbol{w}^{(\mathbf{0})}\right) = \mathbf{0} \qquad \textbf{(2.29)}$$

As a result, Newton's approach iterates as follows:

$$\boldsymbol{w}^{(\boldsymbol{i}+\mathbf{1})} = \boldsymbol{w}^{(\boldsymbol{i})} - \mathbf{H}^{(\boldsymbol{i})-\mathbf{1}}.\,\boldsymbol{g}^{(\boldsymbol{i})} \qquad \textbf{(2.30)}$$

where i=0,1,2, …

Newton's step is the vector $H^{(i)-1}.\,g^{(i)}$. Considering the possibility that this parameter adjustment may trend toward a maximum rather than a minimum. If the Hessian matrix is not positive definite, something may happen to it. Because of this, Newton's technique does not always result in a decrease in the loss index. Newton's method equation is often reworked to avoid this:

$$\boldsymbol{w}^{(\boldsymbol{i}+\mathbf{1})} = \boldsymbol{w}^{(\boldsymbol{i})} - (\boldsymbol{H}^{(\boldsymbol{i})-\mathbf{1}}.\boldsymbol{g}^{(\boldsymbol{i})})\boldsymbol{\eta} \qquad \textbf{(2.31)}$$

where i=0,1,2, …

Line minimization or a fixed value may be used to determine the training rate η. As a result of this change, Newton's training direction is now known as $d^{(i)} = H^{(i)-1}.g^{(i)}$.

The state diagram for Newton's method training is shown in Figure 2.14. First the Newtonian training direction is determined and then an appropriate training rate is selected, and thus, in turn, could lead to optimizing the parameters further.

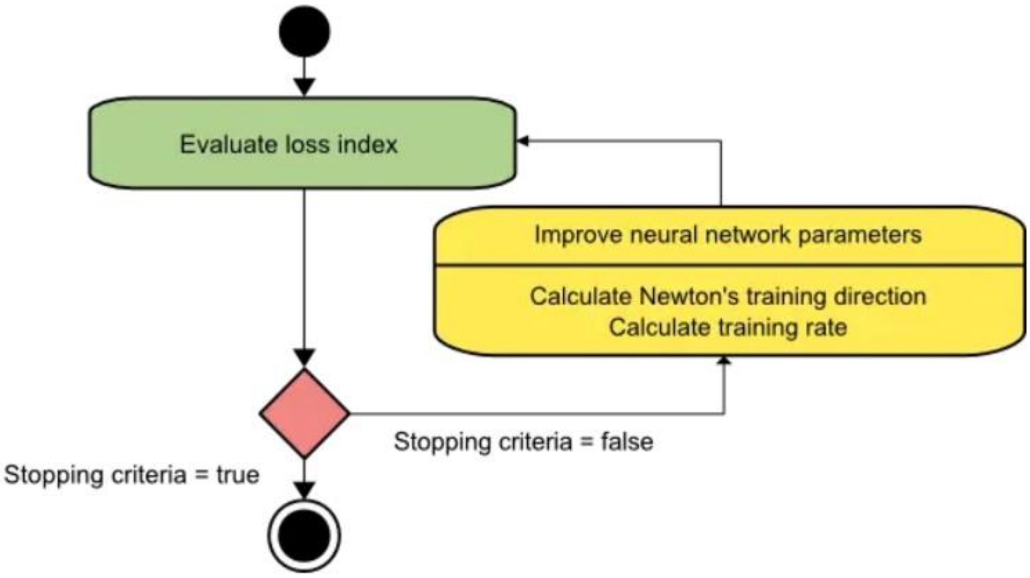

Evaluate loss index
Improve neural network parameters
Calculate Newton's training direction
Calculate training rate
Stopping criteria = false
Stopping criteria = true

Figure 2.14. The steps of Newton's method (Ruder, 2016)

The method's effectiveness is shown in Figure 2.15. Newton's approach takes fewer steps to obtain the least loss function value than GD.

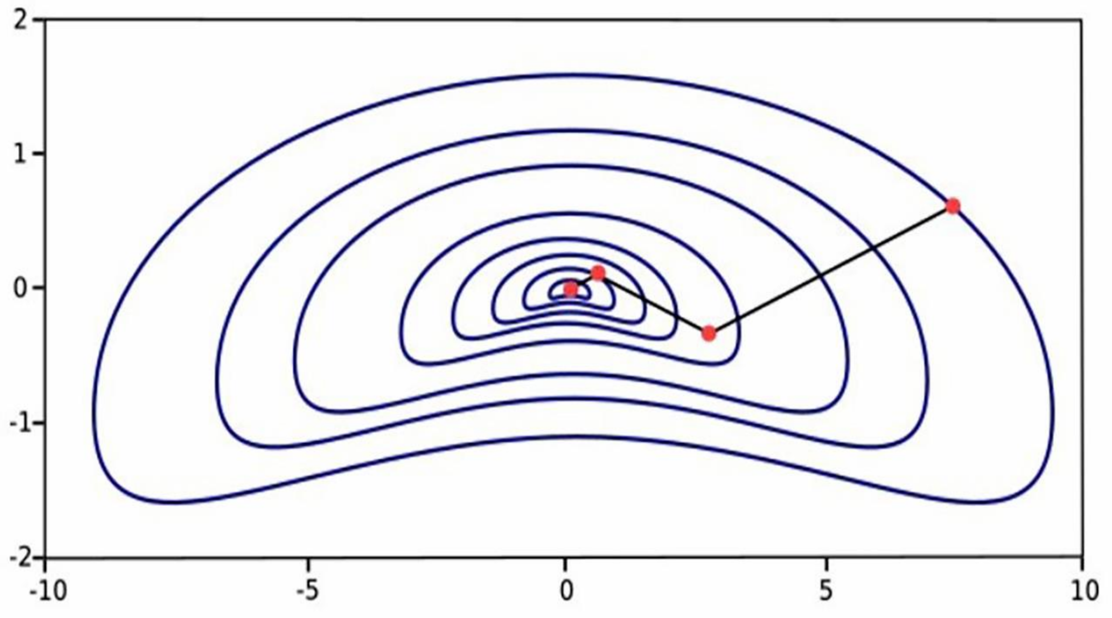


Figure 2.15. The search directions of Newton's method (Du & Swamy, 2013)

Newton's technique, on the other hand, suffers from the fact that the correct assessment of the Hessian and its inverse requires a high computational cost.

- **Conjugate Gradient (CG)**

The conjugate gradient technique is a hybrid of GD and Newton's method. It is driven by the need to hasten GD's generally sluggish convergence. Additionally, this technique eliminates the information needs involved with storing, evaluating, and inverting the Hessian matrix, as Newton's method does (Perry, 1978).

The search is carried out in conjunction with conjugate directions in the conjugate gradient training procedure. They often achieve convergence more quickly than GD directions. Conjugate these training directions using the Hessian matrix.

Let be the training direction vector denoted by **d**. The conjugate gradient technique then creates a series of training directions as follows, beginning with an initial parameter vector $\boldsymbol{w}^{(0)}$ and an initial training direction vector $d^{(0)} = -g^{(0)}$.

$$\boldsymbol{d}^{(\boldsymbol{i}+\mathbf{1})} = \boldsymbol{g}^{(\boldsymbol{i}+\mathbf{1})} + \boldsymbol{d}^{(\boldsymbol{i})}.\boldsymbol{\gamma}^{(\boldsymbol{i})} \qquad \textbf{(2.32)}$$

where i=0,1,2, …

The conjugate parameter is referred to γ, and it may be calculated in a variety of methods. Fletcher and Reeves and Polak and Ribiere are two of the most often cited (Perry, 1978). The training direction is regularly reset to the negative of the gradient in all conjugate gradient methods.

The following expression is used to enhance the parameters.

$$\boldsymbol{w}^{(\boldsymbol{i}+\mathbf{1})} = \boldsymbol{w}^{(\boldsymbol{i})} + \boldsymbol{d}^{(\boldsymbol{i})} . \boldsymbol{\eta}^{(\boldsymbol{i})} \qquad \textbf{(2.33)}$$

where i=0,1,2, …

Figure 2.16 illustrates the activity diagram for the conjugate gradient training process. The parameters are improved in two phases in this case. Initially, the method determines the direction of conjugate gradient training, then it determines a training rate that is appropriate in that direction.

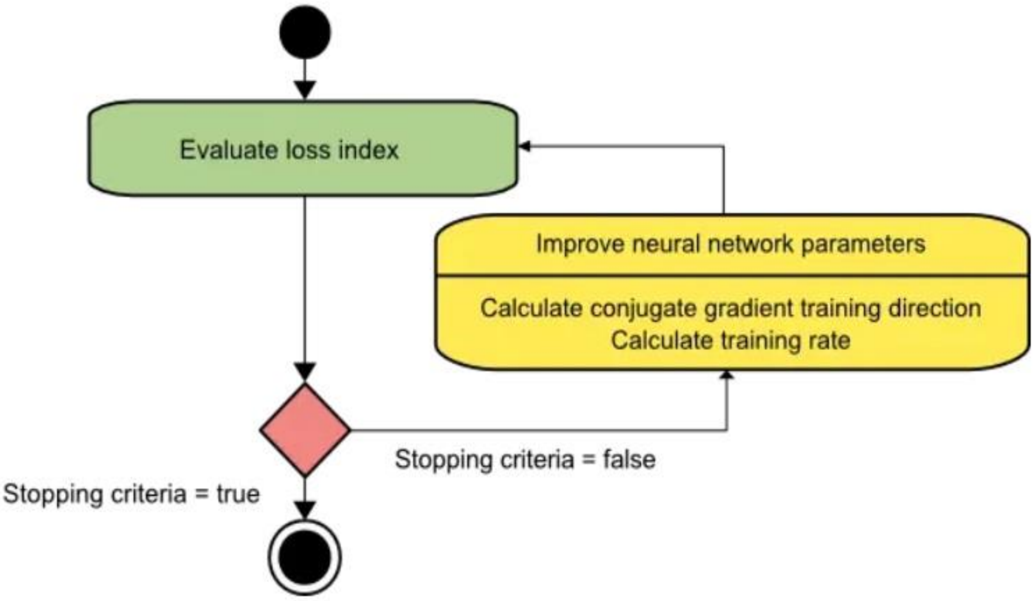

Evaluate loss index
Improve neural network parameters
Calculate conjugate gradient training direction
Calculate training rate
Stopping criteria = false
Stopping criteria = true

Figure 2.16. The steps ofconjugate gradient (Ruder, 2016)

When it comes to neural network training, this approach has shown to be superior than GD. The conjugate gradient works effectively with large NNs since it does not need the Hessian matrix (Du & Swamy, 2013).

- **Quasi-Newton method**

Newton's approach is computationally costly to apply. Indeed, evaluating the Hessian matrix and computing its inverse need a large number of operations. An alternative technique, Quasi-Newton, is developed to overcome this shortcoming. This approach does not begin by computing the Hessian and then evaluating its inverse. It rather constructs an approximation to the inverse Hessian. This approximation is generated using just information about the loss function's first derivatives (Dennis John E & Moré, 1977).

By using the second partial derivatives of the loss function, the Hessian matrix is constructed. Using just the initial partial derivatives of the loss function, the QN approach attempts to approximate the inverse Hessian by using a new matrix B. The QN equation has been formulated as in Eq. 2.34.

$$\boldsymbol{w}^{(\boldsymbol{i}+\mathbf{1})} = \boldsymbol{w}^{(\boldsymbol{i})} - (\boldsymbol{B}^{(\boldsymbol{i})}.\,\boldsymbol{g}^{(\boldsymbol{i})})\boldsymbol{\eta}^{(\boldsymbol{i})} \qquad \textbf{(2.34)}$$

where i=0,1,2, …

The learning rate **η** may be preset or determined using line minimization. The inverse Hessian Approximation B is available in a variety of flavors. The Davidon–Fletcher–Powell (DFP) (Gans & Vacca, 1974) and Broyden–Fletcher–Goldfarb–Shanno (BFGS) formulae are two of the most often utilized (Du & Swamy, 2013).

The QN training process is shown in Figure 2.17. The parameters are improved in two stages. The algorithms begin by determining the QN training direction, then it determines an acceptable training rate.

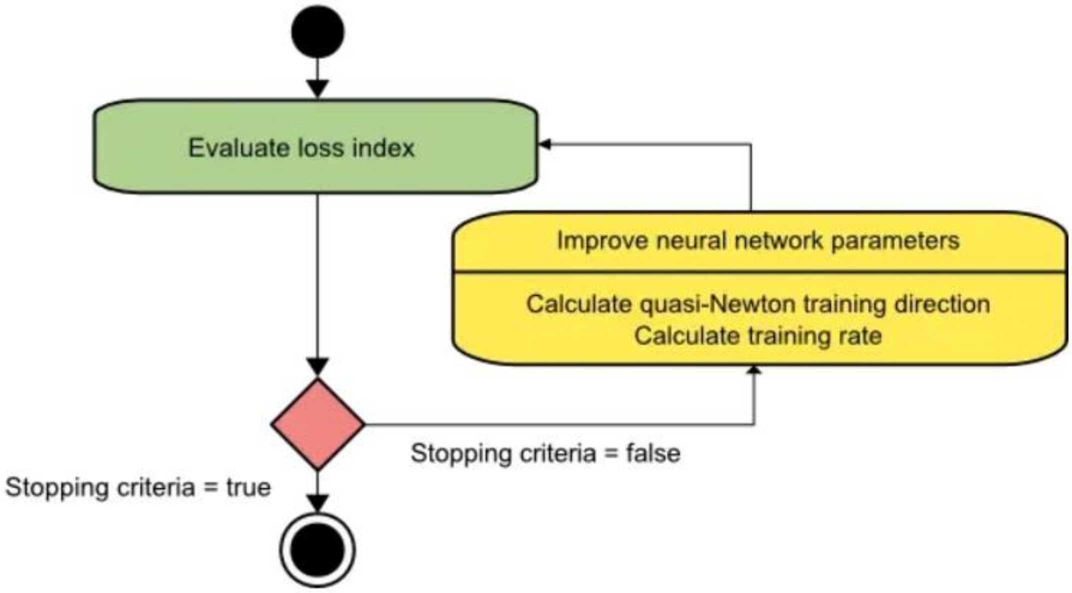

Evaluate loss index
Improve neural network parameters
Calculate quasi-Newton training direction
Calculate training rate
Stopping criteria = false
Stopping criteria = true

Figure 2.17. The Quasi-Newton method (Ruder, 2016)

In most circumstances, this is the best option: Because it does not need computing or inverting the actual Hessian, and also, it is far quicker than GD and CG method.

- **Generalized Reduced Gradient (GRG)**

The GRG algorithm is a deterministic optimization approach initially devised by Abadie and Carpentier in 1969 to address nonlinear constraints problems. It is based mostly on the reduced gradient method. This gradient-based technique constantly seeks the nearest to the initial point, whether global or local (Yeniay, 2005). GRG avoids the use of penalty parameters by searching along curves that are near the potential set. It removes a subset of variables using non-linear programming syntax and equality constraints, reducing the original issue to a space-constrained problem for the remaining variables. Essentially, this strategy employs slack variables to convert inequality constraints into equality constraints (Yeniay, 2005).

Considering a general issue in nonlinear programming with constraints (P), it may be identified as in Eq. 2.35 (Yeniay, 2005).

Minimize $f(x)$ **,** $x \in F \subseteq S$ **(2.35)**

Subject to:

$$H_i(x) = 0,\ i=1,\ldots,p$$

$$G_j(x) \leq 0,\ j=p+1,\ldots,q$$

$$a_k \leq x_k \leq b_k,\ k=1,\ldots,n$$

where $x = [x_1, \ldots, x_n]$ represents an n-variable vector, *f(x)* represents the objective function, $Hi\ (x)\ = 0, i = 1, \ldots, p$ is the i^{th} equality constraint, and $Gj\ (x)\ \leq\ 0\ , j =\ p + 1, \ldots, q$ is the j^{th} inequality constraint. *S* represents the whole search space, whereas *F* represents the feasible search space. The variables a_k andb_k denote the lower and upper boundaries of the variable x.

As a result, GRG assumes that all constraints on (P) are equal and may be expressed as follows:

$$H_i(x) = 0,\ i=1,\ldots,q \qquad (2.36)$$

Where *x* contains all of the variables (original and slacks) and it is then separated into two classes: dependent X_D variables and independent X_I variables.

$$X = \begin{bmatrix} X_D \\ \ldots. \\ X_I \end{bmatrix}$$

In the same way, the boundaries, the objective function gradient, and the Jacobite matrix can be divided as follows:

$$a = \begin{bmatrix} a_D \\ \cdots \\ a_I \end{bmatrix},\quad b = \begin{bmatrix} b_D \\ \cdots \\ b_I \end{bmatrix},\quad \nabla f(x) = \begin{bmatrix} \nabla_D f(x) \\ \cdots \\ \nabla_I f(x) \end{bmatrix},$$

$$J(x) = \begin{bmatrix} \nabla_D H_1(x) & \vdots & \nabla_I H_1(x) \\ \nabla_D H_1(x) & \vdots & \nabla_I H_1(x) \\ & \vdots & \\ \nabla_D H_q(x) & \vdots & \nabla_I H_q(x) \end{bmatrix}$$

Let x^0 be a feasible initial solution, which satisfies bound and equality constraints. The variables should be selected for this reason J_D (x^0) is non-singular. The vector of the reduced gradient is defined as Eq. 37.

$$\boldsymbol{g_I = \nabla_I\, f(\,x^0) - \nabla_D\, f(\,x^0)(J_D\,(x^0)^{\,-1}(J_I\,(x^0)} \qquad \textbf{(2.37)}$$

The search directions for the dependent and the independent variables are given by:

$$\boldsymbol{d_D = -\left(J_D(x^0)\right)^{-1} J_I(x^0) d_I} \qquad ,$$

$$\boldsymbol{d_I = \begin{Bmatrix} 0, & if\ x_i^0 = a_i, g_i > 0 \\ 0, & if\ x_i^0 = b_i, g_i < 0 \\ -g_i, & otherwise \end{Bmatrix}}$$

Calculate the step length α using a line search as a solution to the following problem:

Minimize → $\boldsymbol{f(x^0 + \alpha d)}$***,***

Subject to: $\boldsymbol{0 \le \alpha \le \alpha_{max}}$

Where:

$$\boldsymbol{\alpha_{max} = \sup\left\{\frac{\alpha}{a} \le x^0 \le x^0 + \alpha d \le b\right\}}$$

The optimal solution $\alpha^*$to the problem gives the next solution:

$$\boldsymbol{x^1 = x^0 + \alpha^* d} \qquad \textbf{(2.38)}$$

2.9 Evaluation Measures

This section discusses the various ways to check the performance of machine learning or deep learning models.

2.9.1 Performance Metrics

Mean absolute error (MAE) and Root Mean Squared Error (RMSE) measures have been used to evaluate the prediction error of regression models. The equations of measures mentioned above could be described as follows (Willmott et al., 1985):

- MAE is used to quantity how nearby the prediction outcome from the actual outcomes. MAE suitable to describe uniformly distributed errors.

$$\frac{1}{n}\sum_{k=0}^{n}|A_{k-}P_{k}| \qquad \textbf{(2.39)}$$

Where n: number of data rows, A_k = actual value of record k and P_k = predicted value of record k.

- RMSE is used regularly to measure the differences between the actual and expected values obtained by the model. RMSE provide a complete picture of the error distribution.

$$RMSE = \sqrt{\frac{1}{n}\sum_{k=0}^{n}(P_{k-}A_{k})^{2}} \qquad \textbf{(2.40)}$$

On the other hand, Precision, Recall, F1 score, Specificity, and Accuracy measures are used to evaluate the classification predictor models. Confusion Matrix (CM) is one of the most important method because all these measures fully depend on CM (Han et al., 2011).

Table 2.1. Confusion Matrix

		Predicted class	
		+	–
Actual Class	+	***TP***	**FN**
	–	***FP***	**TN**

CM is a conception of the performance of a supervised learning method. A CM can be seen in Table 2.1, where *TP* (True Positive) is the number of records from class zero predicted correctly as class zero. *TN* (True Negative) is the number of records from class one predicted correctly as class one. *FP* (False Positive) is the number of records from class one predicted incorrectly as class zero. *FN* (False Negative) is the number of records from class zero predicted incorrectly as class one.

- The Precision criterion indicates the proportion of the increasing behavior that can be foreseen accurately using the (FP) parameter. This criterion is determined using the Eq. 2.41. However, since the Precision criteria only considers rising behavior, it cannot detect increasing behavior that has been anticipated as declining (Han et al., 2011). As a result, the following criteria is examined.

$$\boldsymbol{Precision = \frac{TP}{TP+FP}} \qquad \textbf{(2.41)}$$

- The Recall (True Positive Rate) criterion is used to determine the number of true positive predictions associated with the (FN) parameter. Once again, Recall criterion is insufficient because it cannot distinguish the decreasing behavior that has been forecasted mistakenly as increasing. This criterion is determined using the Eq. 2.42 (Han et al., 2011).

$$\boldsymbol{Recall = \frac{TP}{TP+FN}} \qquad \textbf{(2.42)}$$

- F1 Measure criterion: Considering the above assertions, it is possible to lower the recall value by increasing the Precision parameter and vice versa. Given the importance of both Recall and Precision in the algorithm training technique, a third criterion called F1-Measure is employed. This criterion is a mixture of the two previously described criteria. This criterion indicates the degree to which the algorithm was effective in anticipating the growing behavior in terms of Precision. This criterion is determined using the Eq. 2.43 (Han et al., 2011).

$$\boldsymbol{F1 - Measure = \frac{2 * TP}{2 * TP + FN + FP}} \qquad \textbf{(2.43)}$$

- The Specificity (True Negative Rate) criterion indicates the proportion of predicted behavior that really decreases. In other words, it quantifies the likelihood of correctly identifying a declining behavior. This criterion is determined using the Eq. 2.44 (Han et al., 2011).

$$\boldsymbol{Specificity = \frac{TN}{FP+TN}} \qquad \textbf{(2.44)}$$

Given that none of the aforementioned criteria were helpful in measuring the algorithm's quality, the research sought for another criterion, namely Accuracy, to assess the quality of predictions.

- Accuracy Criterion estimates the quality of model. If the model can fairly identify the decreasing behavior but fails to identify the increasing behavior and vice versa, the criteria of Precision and Recall becomes zero and vice versa. However, the accuracy may can solve this problem. This criterion is determined using the Eq. 2.45 (Han et al., 2011).

$$\textbf{Accuracy} = \boldsymbol{\frac{TP+TN}{TP+TN+FP+FN}} \qquad \textbf{(2.45)}$$

2.9.2 K-Cross Validation

Splitting the available dataset between testing and training may lead to an incorrect assessment of the model. The difficulty arises as a result of the fact that the sampled data may not be representative of the whole dataset. With cross-validation, the process of training and testing may be repeated several times using a variety of data sets. For each iteration of the K cross-validation tests, 1-K of the data would be picked for testing and the remaining data would be utilized for training the prediction model. In order to acquire an overall error rate, all iterations' errors are averaged together (Tan et al., 2016).

2.9.3 Portfolio Metric Evaluation

The Sharpe ratio, a risk-return evaluation metric, is utilized in this work to provide a unified measure based on a risk rate. In certain circumstances, it is also known as the reward-volatility measure since its composition is based on calculating the excess return compared to the total risk unit. Eq. 2.46 is used to determine the Sharpe ratio (Sharpe, 1994).

$$\boldsymbol{Sharpe\ Ratio} = \frac{\boldsymbol{Rp - Rf}}{\boldsymbol{\sigma p}} \qquad \textbf{(2.46)}$$

where Rp is the return on the portfolio under consideration and Rf is the risk-free rate of return. The standard deviation of the portfolio is denoted by σp.

Chapter Three

The Proposed System

Design of the Proposed BI System

3.1 Introduction

This dissertation attempts to build an investment system that can help investors and fund managers alike find the best solutions in times of uncertainty. Since the research can contribute to the expansion of local capital markets through increasing investment in portfolios, the research can help improve portfolio efficiency and cash flow of the stock market.

In the field of financial economics, to address the problem of portfolio optimization, a variety of statistical and mathematical approaches have been used. Although there is still a lack of effective trading strategies that correctly integrate the investor's risk appetite with market behavior, there is a dearth of powerful trading. Accordingly, the proposed system uses several deep learning models to optimize portfolios and distribute a lucrative mix of assets. This dissertation illustrates how, in comparison to more conventional approaches, deep learning may contribute in discovering and choosing the best assets to construct a portfolio. The first-DNN is called DLR model which is the primary engine for examining the expected returns of assets, while the second-DNN (whole new network called DPO model) which helps create a portfolio of stocks with a low level of risk and a high expected return. finally, the third-DNN is called DLC model which is used to examine the decision to buy, sell or wait for stocks.

Figure 3.1 shows an overview of the possible functions that the proposed system fulfills. This can be summarized in three main aspects. The first is related to data processing, in particular Iraq's SM dataset, while the second aspect focuses on predicting prices for the general index of the market, companies, or expected daily returns. It is also possible to provide a decision to sell, wait or buy stocks within the stock exchange. Finally, the third aspect considers for how to build effective portfolios that outperform the general market index. It is also worked on improving portfolio by allocating weights to the most important stocks, which in turn meet the investor's demands by achieving the best profits. The selection starts from the dataset and according to the investor's desire, the path is taken by selecting the arrow for the aspect to be decided.

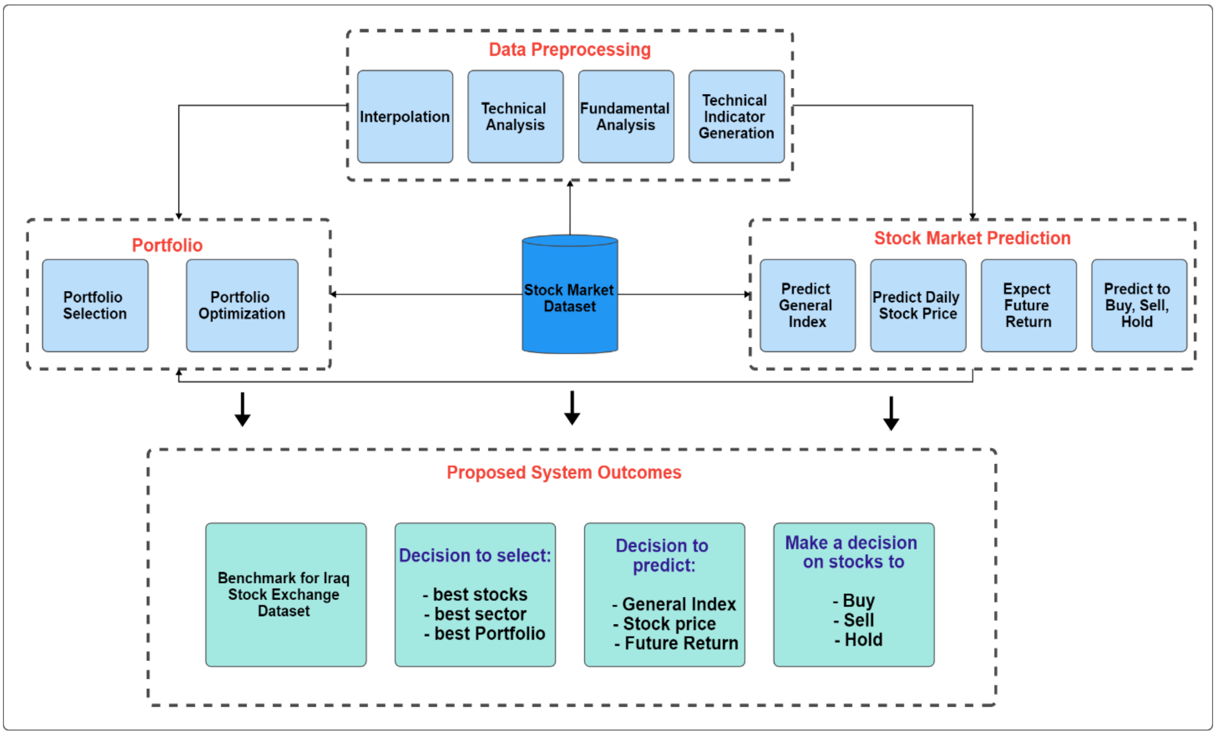


Figure 3.1. An overview of the proposed system

3.2 The Proposed System Architecture

Figure 3.2 depicts the structure of the proposed three models. Unlike previous literature, the proposed system can perform two functionalities which are predicting stock prices and allocating the optimal investment weights to construct portfolio. The proposed system consists of two phases. The first phase aims to predict stock prices. It comprises three layers. The first and second are data preprocessing, whereas the third is for stock price prediction. The second phase is allocated assists' weights for portfolio construction and optimization. The proposed models are based on a newly developed architecture of DNN to automating the trading system. The output of the first phase is either providing the prediction of the stock price as a decision or sharing this prediction with the second phase to construct and optimize the portfolio.

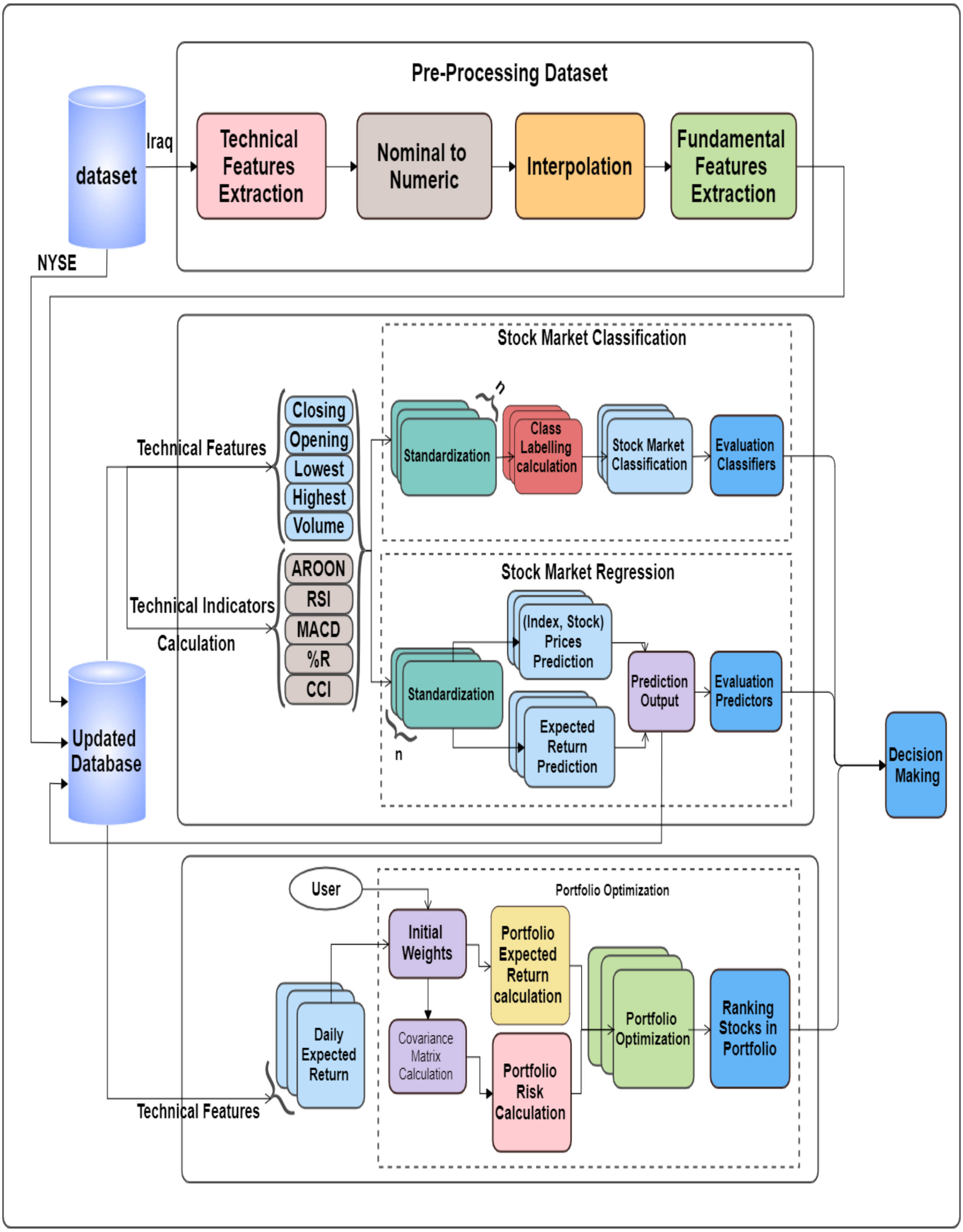


Figure 3.2. The architecture of the proposed system

The proposed system includes four main stages: the first stage is preprocessing dataset that includes: technical features extraction, transform nominal to numeric, interpolation, and fundamental features extraction. The second stage involves building prediction model, which is implemented by one from two models: SM classification model, or time series forecasting model. Both models are built on three main steps (generating technical indicators, standardization, and forecasting) with the exception of the first model, which is interspersed with the class labeling stage to give a decision to sell, hold or buy. The third stage is the evaluation of the result based on the test data using the concept of K-cross validation. Finally, the fourth stage is portfolio optimization, which consists of four steps (calculating portfolio return, calculating portfolio risk, applying the optimization model represented by the GRG method, allocating weights to stocks).

Figure 3.3 shows the two phases and their layers of the proposed DNN topology. It depicts the key structure of the proposed topology, the design of the layers' nodes, and the connection weights between them. The first phase includes three layers namely, TIs generation, standardization, and price prediction, whereas the PO layer represents the second phase of the proposed topology.

The input of the proposed topology is a portfolio of n stocks. Each stock is represented by $\boldsymbol{S_i}$. The information sequence for all stocks is combined into a single entry. For example, the input features of a single stock may be close C_j, open O_j, lowest L_j, highest H_j, and volume V_j with dimensions (5, *r*) where ***r*** denotes the revision window. By pooling the features across all stocks, the input layer's dimension will be increased (5, *m, n*) where n is the number of stocks and m is the number of historical data. This input is subsequently sent to the network to extract the features of TIs.

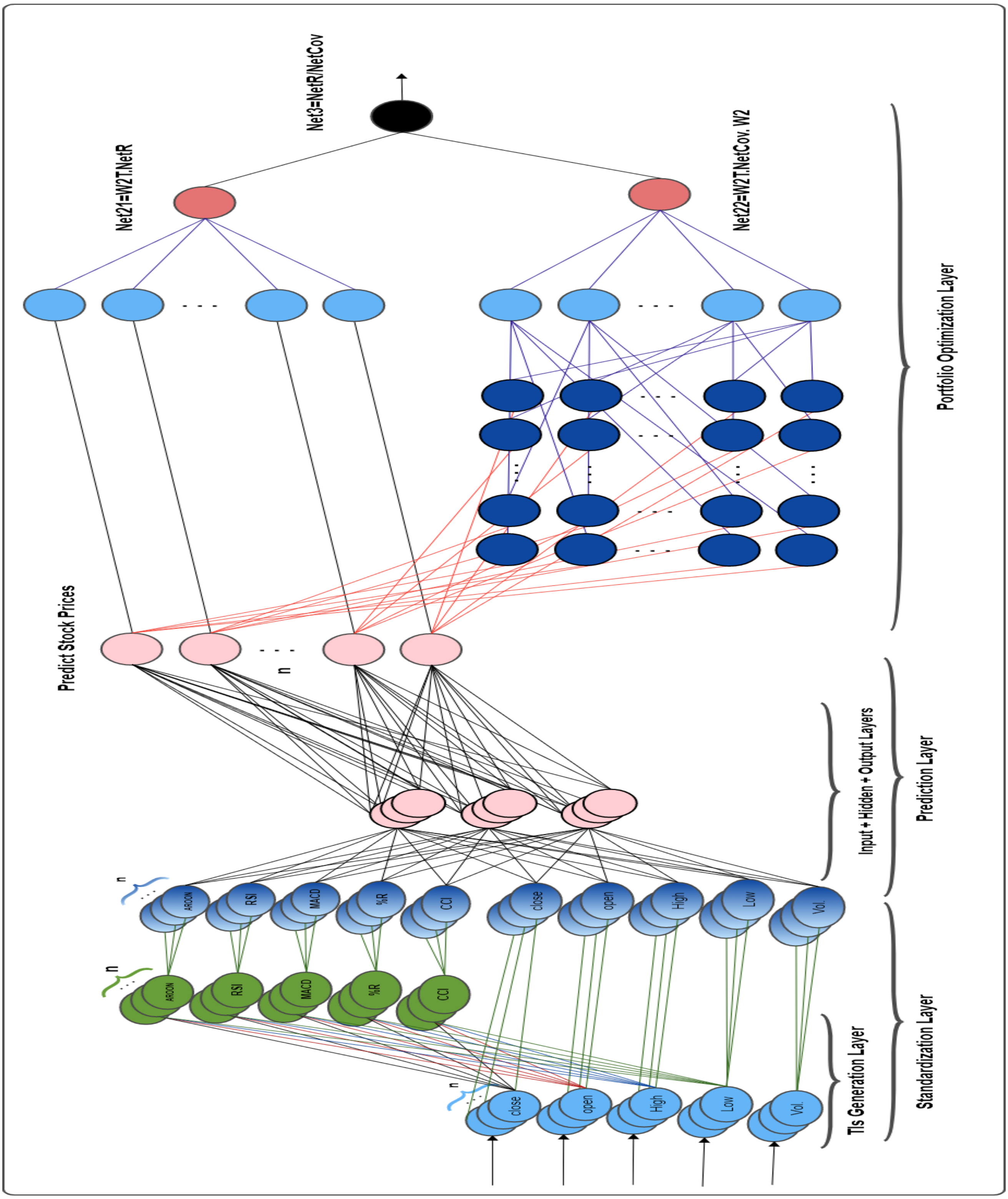


Figure 3.3. The structure of the proposed deep neural network

The detailed description of the proposed system is presented in the next sub-sections.

3.2.1 The Preprocessing Stage

Pre-processing means preparing data for mining in a proper form. The goal of this stage is to build a database for ISX that is organized, structured and evaluated as a benchmark. The data is processed and prepared into prediction (classification or regression) or to PO stage because actual dataset like Iraq SM might have some unsuitable structure. This stage consists of three main steps, including features extraction, transforming nominal to numeric, and interpolation. Algorithm 3.1 illustrates the preprocessing stage.

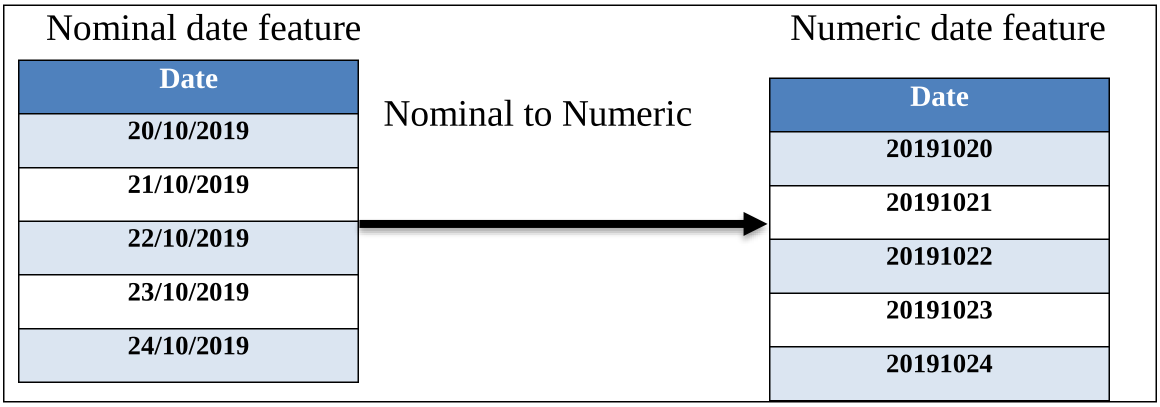
```
Algorithm 3.1: Preprocessing stock market dataset
- Input: An array RD $(n \times m \times r)$,represents raw stock market dataset where n
  is the no. of stocks, m is the no. of Standard technical features, and r is
  historical data size.
- Output: Array PD$(n \times |F| \times r)$: Processed stock market dataset, where |F| is
  the no. of technical features
  1.  For each Stock s in RD do
  2.    Select F from 1…m in RD
        /* Transforming Date nominal feature in RD, to numeric feature by*/
  3.    Set count ← 0
  4.    For each Date_value in F do
  5.          P_date [count] ← value (concatenation (Date_value [count]))
  6.          count ← count+1
  7.    End For
        /* Applying Linear Regression to find missing value */
  8.    Set k ←5, the most recent days
  9.    For each Feature f in F do
  10.     For each data row r in RD do
  11.       If($RD_r^f$ is missing) then
  12.          $RD_r^f \leftarrow$ procedure Interpolation $(P\_date, f, r,\ k)$
  13.       End if
  14.     End For
  15.   End For
  16.   Set PD ← RD after replacing Date_value by P_Date feature
  17.End for
  End.
```

According to Algorithm 3.1, the first step is to extract the traditional technical features (date, open, high, low, close, and trading volumes) and this is done in (step no. 2) by deleting any other features that are not necessary for the prediction stage, such as the company symbol, current rate, previous rate, change (%), and shares traded. The goal of this step is to make the ISX dataset match its features with the standard NYSE dataset. Table (3.1) shows the type of standard technical features.

Table (3.1) Standard technical features of SM

Feature's name	Type
Date	nominal
Open	numeric
Low	numeric
High	numeric
Close	numeric
Volume	numeric

After extracting the technical features, the next step is transforming nominal features of SM dataset into numerical features. Steps 3-7 from Algorithm 3.1 are used to implement this step. The Linear Regression and DNN Models of the proposed system do not deal with nominal features. For this reason, this step is applied to make the proposed system capable to deal with all SM features. Figure 3.4 describes the idea of transformation of SM data.



Date
20/10/2019
21/10/2019
22/10/2019
23/10/2019
24/10/2019

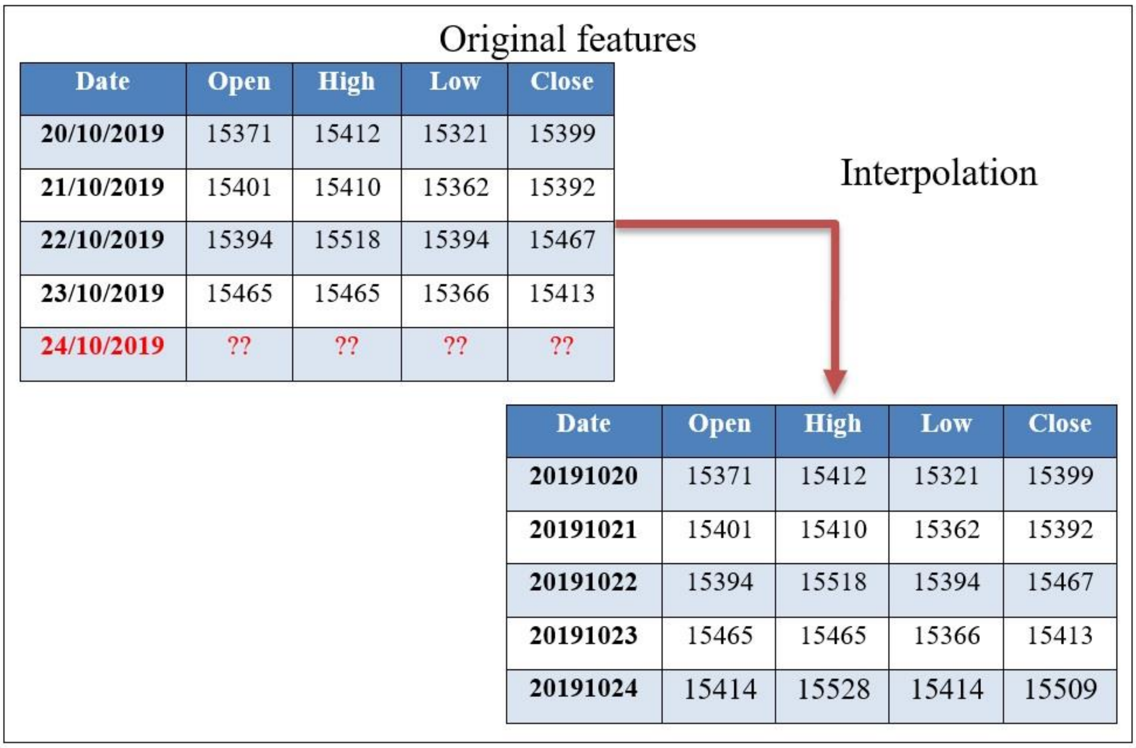
Date
20191020
20191021
20191022
20191023
20191024

Figure 3.4. Nominal to Numeric

Steps 8-15 of Algorithm 3.1 represent the interpolation step. Interpolation is carried out when information about prices and volume for a trading day is not available. For some stocks, several points are missing in the data. Overall, the missing data constitutes less than 0.1% out of all data points. In this step, sliding-window Linear Regression model is employed for replacing missing value with the existing adjacent prices according to Procedure 3.1 Interpolation.

```
Procedure 3.1: Interpolation (X, Y, r, period)
1.  Let sp: represent the start slide window and ep:  represent the end slide
    window
2.  If (There is no missing value before and after Y(r) within the period) then
3.          $sp \leftarrow r - period,\ ep \leftarrow r + period$
4.  Else if (There is a missing value after the Y(r) within the period) then
5.          $sp \leftarrow r - period,\ ep \leftarrow r$
6.  End if
7.  $X \leftarrow \sum_{i=sp}^{ep} X_i$
8.  $Y \leftarrow \sum_{i=sp}^{ep} Y_i$
9.  $XY \leftarrow \sum_{i=sp}^{ep} (X_i \times Y_i)$
10. $X^2 \leftarrow \sum_{i=sp}^{ep} (X_i \times X_i)$
11. $Y^2 \leftarrow \sum_{i=sp}^{ep} (Y_i \times Y_i)$
12. $N \leftarrow (ep - sp)$            /* Find the length of the period */
13. $\boldsymbol{a} \leftarrow \frac{(Y)(X^2) - (X)(XY)}{N(X^2) - (X)^2}$            /* Find Slope */
14. $\boldsymbol{b} \leftarrow \frac{N(XY) - (X)(Y)}{N(X^2) - (X)^2}$            /* Find Intercept */
15. $\boldsymbol{\hat{Y}_r} \leftarrow \boldsymbol{a} + \boldsymbol{b}.\boldsymbol{X_r}$            /* Apply Linear Regression */
16. Return ($\boldsymbol{\hat{Y}_r}$)
End.
```

The main idea of this procedure is to scan the dataset from the beginning until a trading day (i) has a missing value. On that day (i), we check k-period of days before and after that day, so if all the data is available, we determine the size of the window by range [i-k to i+k], else if there is any missing value after day (i) within the k-period, we determine the size of the window by the range [i-k to i]. After that, steps 7- 16 in Procedure 3.1 are used to handling missing value based on Linear Regression. Figure 3.5 describes the idea of interpolation of SM data.



Original features

Date	Open	High	Low	Close
20/10/2019	15371	15412	15321	15399
21/10/2019	15401	15410	15362	15392
22/10/2019	15394	15518	15394	15467
23/10/2019	15465	15465	15366	15413
24/10/2019	??	??	??	??

Interpolation

Date	Open	High	Low	Close
20191020	15371	15412	15321	15399
20191021	15401	15410	15362	15392
20191022	15394	15518	15394	15467
20191023	15465	15465	15366	15413
20191024	15414	15528	15414	15509

Figure 3.5. Interpolation

The analysis of financial markets is done through technical or fundamental analysis, so in some cases the researchers may need to include the fundamental features with the technical features to give a better decision. Therefore, in addition to the above three steps for building a benchmark for Iraq dataset, the dissertation worked on extracting most of the fundamental features, as shown in Table (3.2).

Table (3.2) Fundamental features of Iraq SM dataset

Feature's name	Type	Feature's name	Type
Listing date	nominal	Current capital	numeric
Trading started	nominal	Capital at establishment	numeric
Last trading date	nominal	Rate of change	numeric
No. of trading days	numeric	Initial share price	numeric
Average number of deals	numeric	Current share price	numeric
Turnover rate	numeric		

3.2.2 The Prediction Stage

The SM is highly volatile and complex in nature. Technical analysts often apply technical analysis on historical price data, which is an exhaustive task and might produce incorrect predictions. The machine learning coupled with fundamental and / or technical analysis which can yield satisfactory results for SM prediction. In this work, an effort is made to predict the price and price trend of stocks by building a new prediction model. It is implemented by one from two models: DLR model to predict the future stock price, general index price, and daily return or DLC model to make a decision of buy, sell, or hold stocks. The approach proposed in this work is capable to identify hidden relationships and underlying dynamics in the historical price data through an optimal deep learning approach.

- **Deep Learning Regression model**

The main objective of the Deep Learning Regression (DLR) model is to predict the future price of stocks, which can successfully trade SM for a profit. This is achieved by building a certified and a perfect regression model for the SM inspired by the idea of deep learning in NN. This model is based on solving the problem completely from the input until predicting the price. It consists of three main layers (TIs generation layer, standardization layer, and prediction layer) sequentially with specific objective to each layer. The topology of NN is determined according to the problem given as a first step before designing any NN. The topology of the suggested DNN is shown in Figure 3.6.

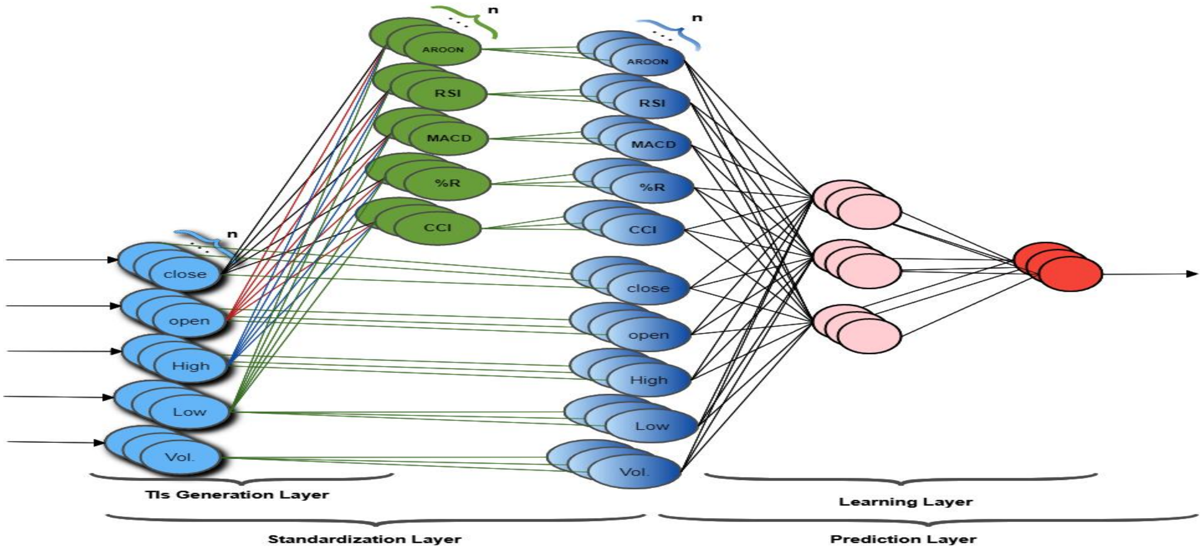


Figure 3.6. The topology DLR model

Layer1: (TIs Generation layer)

The active stock traders and technical analysts commonly use TIs to analyze short-term and long-term price movements and to identify entry and exit points. Technical indicators can be useful while predicting the future prices of assets so they can be integrated into automated trading systems. As a result, this layer aims to enhance the accuracy of prediction by taking the standard features (five features) of the SM as an inputs for the network. The outputs from this layer are the five TIs features.

The major criteria of this layer are:

I. Each output neuron (indicator) connect only with input neurons were contributed to find this indicator.

II. The values of weights are constant values and not updated.

III. All output neurons are calculated in parallel by using the multi-thread principle.

IV. Linear activation function is used to find the output neurons.

Other details of the TIs generating layer are presented in Algorithm 3.2. This algorithm is constructed using Eq. 2.12 to 2.20. By using the concept of transformation, the mathematical formulations for these equations are translated into formulas suited for NNs (statistical with NNs). This implies that each equation has been adjusted to the form like (net), and then the linear activation function F() is used to determine the layer's output.

Algorithm 3.2: The TIs Generation Layer

```
Input: An array S (n × m × r), represents stock market dataset where n is no. of stocks, m is no. of standard technical features, and r is historical data size.
Output: Array O1(n × j × r), j is the  no. of technical Indicator features
Begin
  Let k represents period sliding window
  Initialize k ←14
  /*the initial weights matrix (W)*/
  $W \leftarrow \{\{\frac{100}{k},0,0,0,0\},\{100,0,0,0,0\},\{\frac{1}{0.015},\frac{1}{0.015},\frac{1}{0.015}, ,0,0\},\{\frac{1}{k},\frac{1}{k},\frac{1}{k},0,0\},\{-100,-100,-100,0,0\}\}$
  For each Stock s in S do
     Set $Close \leftarrow s_{1r}$, $High \leftarrow s_{2r}$, $Low \leftarrow s_{3r}$
     For i ←0 to m do
           $AroonNet(i) \leftarrow$ procedure AROON $(s, Close, i, k)$
           $O_1^1(i) \leftarrow F(AroonNet(i) \bullet W_{1i})$
           $RSINet(i) \leftarrow$ procedure RSI $(s, Close, i, k)$
           $O_1^2(i) \leftarrow F(RSINet(i) \bullet W_{2i})$
           $CCINet(i) \leftarrow$ procedure CCI $(s, Close, High, Low, i, k)$
           $O_1^3(i) \leftarrow F(CCINet(i) \bullet W_{3i})$
           $MACDNet(i) \leftarrow$ procedure MACD $(s, Close, i, k)$
           $O_1^4(i) \leftarrow F(MACDNet(i) \bullet W_{4i})$
           $RNet(i) \leftarrow$ procedure %R $(s, Close, High, Low, i, k)$
           $O_1^5(i) \leftarrow F(RNet(i) \bullet W_{5i})$
       End For
     End For
  Input2 [n×|S+O1|×r] =Merge Array S [n×m×r] with Array O1[n × j × r]
End.
```

The input to Algorithm 3.2 is ***n*** arrays with size (m× r) , which represents SM dataset where m is the no. of standard technical features, and r is historical data size. This algorithm works in parallel on all the stocks, and for each stock it implements five procedures (AROON, RSI, CCI, MACD, and %R), these are also executed in parallel to find a TIs features.

Procedure 3.2 AROON: In essence, this indicator measures the time between highs and the time between lows over a time period. The idea is that strong uptrends will regularly see new highs, and strong downtrends will regularly see new lows. The indicator consists of the "AroonUp" line, which measures the strength of the uptrend, and the "AroonDown" line, which measures the strength of the downtrend. Procedure 3.2 employs only “AroonUp" line.

Procedure 3.2: ***AROON (s, Close, i, k)***

```
Let b represents the length of the period from i to maximum close in period k
Set a ← 0
Set b ← 0
Set max ← Close[i]
For j ← i down to (i-k)
    If(close[i]<max) then
        max ← close[i]
        a ←b;
    end if
    b ← b+1
End for
b ←14-a
Return (b)
End.
```

AroonUp measures how long it has been since prices have recorded a new high within the specified period. If the current bar’s high is the highest within the user defined number of periods before it, then the AroonUp value is 100. In other words, it is a new high for the period. Otherwise, it returns a percent value indicating the time since a new high occurred for the specified period. Procedure 3.2 calculates AroonUp by finding how many periods have been since the last 14-period high, after that, subtract this from 14.

The RSI indicator shown in Procedure 3.3 is a momentum indicator with a range of 0 to 100. Traditional interpretation and usage of the RSI is that values of 70 or above indicate that a security is becoming overbought or overvalued and may be primed for a trend reversal or corrective pullback in price. An RSI reading of 30 or below indicates an oversold or undervalued condition.

```
Procedure 3.3: RSI (s, Close, i, k)
Set Boolean flag ← false
If (flag =false)
  // Find Previous gain average
    For t← i down to i-k do
        Set gain ← Close[t]-Close[t-1]
        If(gain>0) then
            AvgGain ←AvgGain+Abs(gain)
        else
            AvgLoss ←AvgLoss+ Abs(gain)
        End If
    End for.
    flag ← true
    AvgGain ← AvgGain/k
    AvgLoss ← AvgLoss/k
Else
  if (Close[i]>Close[i-1]) then
      CurrentGain ← Abs(Close[i]-Close[i-1])
  Else
      CurrentLoss ← Abs(Close[i]-Close[i-1])
  End if
  AvgGain ← [AvgGain*(k-1) +CurrentGain]/k
  AvgLoss ← [AvgLoss*(k-1) +CurrentLoss]/k
End if
RS ←AvgGain/AvgLoss
Return (1-[1/(1+RS)])
End.
```

Procedure 3.3 calculates the RSI indicator using the average changes in closing prices during the specified period, according to the following steps:

1. Calculating the average gains and average losses during the specified period (typically 14 days). To do so, by taking the difference in price between subsequent days, (pricet – pricet-1), and if we see the result is positive or negative. If) pricet – pricet-1 > 0), we store the absolute difference in the AvgGain counter. Otherwise if (pricet – pricet-1 < 0), we store the absolute difference in the AvgLoss counter.

2. Calculating RS stands for relative strength by dividing average gains on average losses during the specified period.
3. Calculate and retrieve the value of the RSI through step no. 25

The CCI indicator measures the difference between the current price and the historical average price. When the CCI is above zero, it indicates the price is above the historical average. Conversely, when the CCI is below zero, price is below the historical average.

```
Procedure 3.4: CCI (s, Close, High, Low, i, k)
Set ma ← 0
Set md ← 0
Define array TP with size k
For j ← i down to (i-k) do
    TP[j] ← (close[j]+High[j]+Low[j])/3
    ma ← ma+TP[j]
End for
ma ← ma/k
For j ← i down to (i-k) do
    md ← md+Abs(TP[j]-ma)
End for
md ← md/k
Return ((TP[i]-ma)/md)
End.
```

The following steps explain how the Procedure 3.4 works:

1. Determining how many periods your CCI will analyze. Fourteen is commonly used. Fewer periods result in a more volatile indicator, while more periods will make it smoother. In this calculation, we assume 14 periods.

2. In a dataset, tracking the high, low, and close for 20 periods and computing the Typical Price (TP).

3. After 14 periods, computing the Moving Average (MA) of the TP by summing the last 14 TP and dividing by 14.

4. Calculating the Mean Deviation (MD) by subtracting the MA from TP for the last 14 periods. Sum the absolute values (ignore minus signs) of MD and then divide by 14.

5. Inserting the most recent TP, the MA, and the MD into the formula to compute the current CCI reading.

6. Repeat the process as each new period ends.

Moving average convergence divergence (MACD) is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price. The MACD is calculated by subtracting the 26-period EMA from the 12-period EMA.

```
Procedure 3.5: MACD (s, Close, i, k)
Set K1 ← 2/(1+12)
Set K2 ← 2/(1+26)
Set sum ← 0
For t ← 26 to k do
    Calculate E26(t) ← EMA(Close(i), 26)
    A ← [K1 * Close(t) – E26(t-1)] +E26(t-1)
    Calculate E12(t) ← EMA(Close(i), 12)
    B ← [K2 * Close(t) – E12(t-1)] +E12(t-1)
    $MACD_t \leftarrow A-B$
End for
$I \leftarrow \sum_{t=i-k}^{i} MACD_t$
Return (I)
End.
```

Procedure 3.5 calculates MACD by subtracting the long-term EMA (26 periods) from the short-term EMA (12 periods). An EMA is a type of MA that places a greater weight and significance on the most recent data points.

Williams %R is a type of momentum indicator that moves between 0 and -100 and measures overbought and oversold levels.

Procedure 3.6: ***%R (s, Close, High, Low, i, k)***

```
Set max ← High[i]
Set min ← Low[i]
For j ← i down to (i-k)
    If (High[j]>max) then
        max ← High[j]
    End if
    If (Low[j]<min) then
        Min ← Low[j]
    End if
End for
Return ((max-close[i])/(max-min))
End.
```

Procedure 3.6 calculates %R based on price, typically over the last 14 periods, accordingly these steps:

1. Recording the high and low for each period over 14 periods.
2. On the 14th period, note the current price, the highest price, and lowest price. It is now possible to fill in all the formula variables for Williams %R.

- **Layer2: (Standardization Regression Layer)**

Standardization scales each input variable separately by subtracting the mean (called centering) and dividing by the standard deviation to shift the distribution to have a mean of zero and a standard deviation of one. Standardization assumes that your observations fit a Gaussian distribution (bell curve) with a well-behaved mean and standard deviation. This layer's primary purpose is to prevent a huge variable from influencing the computation results. A theoretical illustration of the standardization layer can be found in Algorithm 3.3.

Algorithm 3.3: Standardization Layer

***Input**: Array $Input_2$ $[n\times|S+O_1|\times r]$, input and output of TIs layer where **n** is the no. of stocks, S is input array of TIs layer, O_1 is output array of TIs layer, and **r** is the historical data size.*

***Output:** Array (**O_2**$[n \times |S + O1| \times r]$)*

```
Begin
For s ← 0 to n do
    For f ← 0 to |S+O_1| do
        /* Compute the Mean of each input neuron f */
        $\mu_f \leftarrow \sum_{k=0}^{r} \mathrm{Input}_f^s(k)$
        $\mu_f \leftarrow \mu_f / r$
        /* Compute the standard deviation of each input neuron f */
        $\mathrm{SD}_f \leftarrow \sum_{k=0}^{r} \mathrm{Input}_f^s(k) - \mu_f$
        $\mathrm{SD}_f \leftarrow \mathrm{SD}_f / r$
        /* Calculate the weight of each neuron */
        $W_f \leftarrow 1/SD_f$
        For k ← 0 to r do
            StandNet ← $\mathrm{Input}_f^s(k) - \mu_f$
            $O_2^f(k) \leftarrow W_f$* StandNet.
        End for
    End for
End for
Set $Input_3$ $[n\times| O_2|\times r] \leftarrow O_2[n \times |S + O1| \times r]$
End.
```

The input to Algorithm 3.3 is ***n*** arrays with size $(\mathrm{m}\times r)$, which represents SM dataset where m is the no. of inputs and outputs of TI's Generation layer, and r is historical data size. . Ten standardize features are produced as a result from this layer. This algorithm works in parallel on all the stocks. For each stock, the algorithm standardizes the ten features simultaneously. Algorithm 3.3 calculates the statistical mean and standard deviation of the attribute values, subtract the mean from each value, and divide the result by the standard deviation. This process is called standardizing a statistical variable and results in a set of values with range [-1,1], in which their mean is zero and standard deviation is one.

Layer3: (Prediction Layer)

The basic idea of this layer is developing a prediction model through optimizing Multi-Layer Perceptron (MLP) neural network, which trains using Back-Propagation algorithm with one hidden layer by minimizing the given loss function (Mean Squared Error) using Quasi-Newton method based on projected gradients instead of using standard Gradient Descent.

MLP composes three layers. Input is the first layer and it is compatible with input variables of a problem with a node for each input variable (10 neurons). The hidden layer is the second layer, which is used to capture nonlinear relationships between variables. Two neurons are determined in hidden layer in addition to bias. The third layer represents the output layer, which is used to provide the expected values. One neuron is selected in output layer, which represents the desired output (close price in continuous form). Equation 3.1 describes the relationship between the input x_t and the output y_t:

$$\boldsymbol{Y_t = F(w_O \ + \textstyle\sum_{a=1}^{h} w_a \bullet F\big[w_{0,a} + \sum_{b=1}^{m} w_{b,a} \bullet x_t\ \big])} \qquad \textbf{(3.1)}$$

Where $\boldsymbol{w_{b,a}}\ (a = 1,2,\dots,h;\ b = 1,2,\dots,m)$, $\boldsymbol{w_a}\ (a = 0,1,2,\dots,h)$, $\boldsymbol{w_{0,a}}$, and $\boldsymbol{w_O}$ are the weights of the network, *m*: input neurons, h: hidden neurons. $\boldsymbol{w_{b,a}}$ represents the weights between input and hidden layer, $\boldsymbol{w_a}$ represents the weights between hidden and output layer, $\boldsymbol{w_{0,a}}$ represents the weight of input bias neuron, $\boldsymbol{w_O}$ represents the weight of hidden bias neuron, and finally **F()** represents a non-linear activation function.

A sigmoid (bipolar) represented in Eq. 2.24, sigmoid (unipolar) represented in Eq. 2.25 and hyperbolic (tanh) represented in Eq. 2.26 activation functions are used in the hidden and/or output layer. Additionally, the proposed approximation activation function can also be used in these layers. The idea is to approximate sigmoid (logistic) and the hyperbolic (tanh) activation function by exploit Taylor's or Maclaurin's formula to find approximate value to ($\boldsymbol{e^x}$). The approximation procedure is shown in Equations 3.2 to 3.6.

$$\mathbf{e^x = 1 + \sum_{n=0}^{\infty} \frac{x^n}{n!}} \quad \textbf{(3.2)}$$

For example, if we need to find the value of any node in hidden layer. These approximation functions are given by:

$$\boldsymbol{net_a = \sum_{b=1}^{m} w_{b,a} \bullet input_b} \quad \textbf{(3.3)}$$

$y1 = 1.0 + (-net_a) / 4096.0$

$y2 = 1.0 + (net_a) / 4096.0$

$x = y1 * y1$

$x=\prod_{t=1}^{11} x$

$z=y2*y2$

$z=\prod_{t=1}^{11} z$

The approximation function of bipolar sigmoid function is formed as:

$$\boldsymbol{F\ (net_a) = 2/\ (1+x)-1} \quad \textbf{(3.4)}$$

The approximation function of unipolar sigmoid function is formed as:

$$\boldsymbol{F\ (net_a) = 1\ /\ (1 + x)} \quad \textbf{(3.5)}$$

The approximation function of tanh hyperbolic function is formed as:

$$\boldsymbol{F\ (net_a) = (z - x)/\ (z + x)} \quad \textbf{(3.6)}$$

After determining the best topology for the network, the next stage is the learning. The procedure used to carry out the learning process is called the training (or learning) strategy. The training strategy is applied in NN to obtain the possible minimum loss function. This is done by searching for parameters (weights) that fit the neural network to the dataset.

The loss function plays a vital role in the use of neural networks. It defines the task the neural network is required to do and provides a measure of the quality of the representation required to learn. The mean squared error is used as a loss function in this topology, which it calculates the average squared error between the outputs from the neural network and the targets in the dataset.

The learning problem for neural networks can then be stated as finding a neural network function for which the loss function takes on a minimum value. That is, to find the set of parameters (weights) that minimize the above function. The learning problem for neural networks consists of searching for a set of parameters at which the loss function takes a minimum value. The necessary condition states that the gradient is zero when the neural network is at a minimum of the loss index.

The loss index is, in general, a non-linear function of the parameters. Consequently, it is not possible to find closed optimization algorithms for the minima. Instead, we consider a search through the parameter space consisting of a succession of steps or epochs. At each epoch, the loss will decrease by adjusting the neural network parameters. The change of parameters between two epochs is called the parameter increment.

In this way, to train a neural network, we start with some parameters vector (often chosen at random). We generate a sequence of parameter vectors so that the loss index is reduced at each algorithm iteration.

The optimization algorithm determines how the adjustment of the parameters in the neural network takes place. The optimization algorithm stops when a specified condition is satisfied. This topology stops when the loss has been minimized to a goal value (tolerance is equal to 1.0E-6).

Finally, the prediction layer uses the Quasi-Newton method which is based on Newton's method but does not require the calculation of second derivatives. Instead, the Quasi-Newton method computes an approximation of the inverse Hessian at each iteration of the algorithm by only using gradient information. Algorithm 3.4 shows optimization algorithm steps to train the MLP.

Algorithm 3.4: Optimization DLR model using BFGS method

```
  Input: Array Input_3 ($n\times|O_2|\times r$), output of Standardization layer where n is the no. of stocks, $|O_2|$ is the size of Standardization layer array output, and r is the historical data size.
  Output: Array W, An array of final weights.
Begin
Let W be the vector of all weight in NNs
Let CF be the Converges Factor
  $k \leftarrow 0$
  Tolerance $\leftarrow$ 1.0E-6
  Ridge $\leftarrow$ 0.01
  Get initial variable values $W_k$ randomly in rang (0,1)
  Initialize a symmetric positive definite matrix $B_k$
      $d_k \leftarrow -B_k^{-1}g_k$
  while CF $\neq$ Tolerance do
      Calculate the Jacobian (gradient) vector $G_k$ using $W_k$, and $input_3$
      Search for $\alpha$ using a line search and set $W_{k+1} = W_k + \alpha d_k$
      $G_{k+1} \leftarrow G_i + Ridge * W_{K+1}$
      $\Delta W_k \leftarrow W_{k+1} - W_k$
      $\Delta G_k \leftarrow G_{k+1} - G_k$
      $B_{k+1} \leftarrow B_k + \frac{\Delta G_k \Delta G_k^T}{\Delta G_k^T \Delta W_k} - \frac{B_k \Delta W_k \Delta W_k^T B_k}{\Delta W_k^T B_k \Delta W_k}$
      $k \leftarrow k+1$
  End while
End
```

- **Deep Learning Classification model**

The basic idea of the Deep Learning Classification (DLC) model is to predict the decision for (buy, sell, or hold) stocks, which can successfully trade SM for a profit. This achieves by building a certified and a perfect classification model for the SM inspired by the idea of deep learning in NN, where this model is based on solving the problem completely from the input until predicting the price. This model consists of three main layers (TIs generation layer, standardization layer, and prediction layer) sequentially with specific objective to each layer. The topology of NN determines according to the problem given as a first step before design any NN. The topology of the suggested DNN is shown in Figure 3.7.

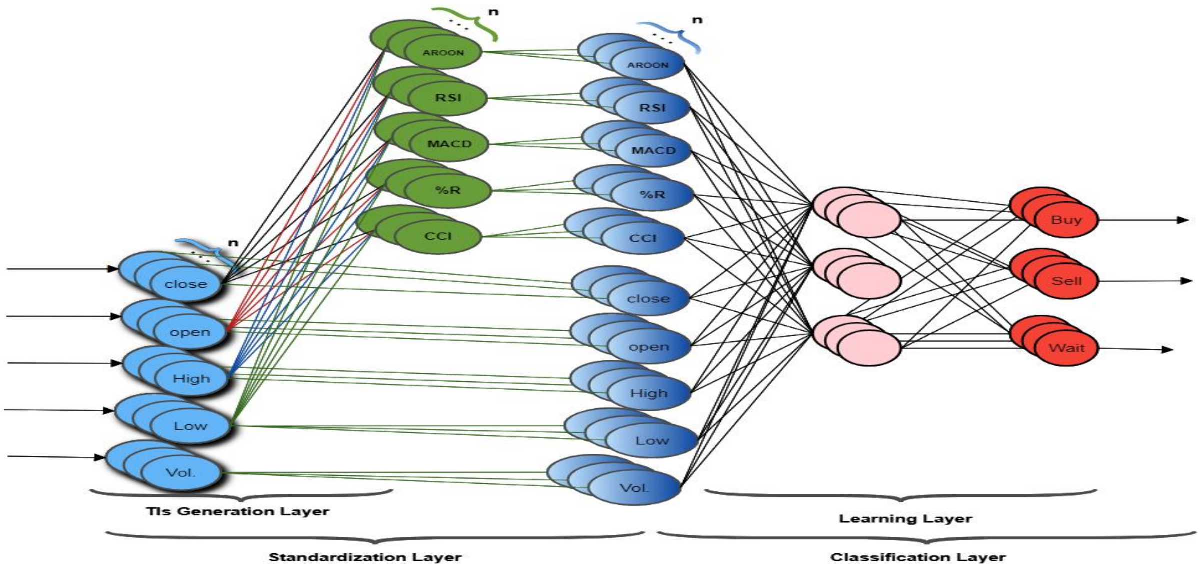


Figure 3.7. The topology DLC Model

The proposed classification network is similar to the DLR model with some minor changes in the third layer. Where the third layer starts with class labeling step, which is necessary for the classification stage.

The class attribute is calculated using two different equations. First, Eq. (3.7) current data point of close price minus previous data point of close price, if positive is denoted by a ‘sell’, negative is denoted by a ‘buy’ otherwise is denoted by a ‘hold’. Second, Eq. (3.8) a traditional percentage gain is required before classifying a point as ‘buy’. Where if at least 0.5 % gain is wanted with once a day from the previous day’s close then will be classified as ‘sell’, otherwise it is classified ‘buy’.

$$\mathrm{C_i} = \Delta\, close\ price \qquad \textbf{(3.7)}$$

$$\mathrm{C_i} = \begin{cases} X & if\ close_i > close_{i-1} \\ 0 & if\ close_i = close_{i-1} \\ -1 & if\ close_i < close_{i-1} \end{cases} \qquad \textbf{(3.8)}$$

$$\mathrm{X} = \begin{cases} 1 & if\ gain_i \geq 0.5 \\ -1 & otherwize \end{cases}$$

$$gain_i = \frac{close_i - close_{i-1}}{close_i} * 100$$

Optimization algorithm consists of many steps to optimize the DLC as shown in Algorithm 3.5

Algorithm 3.5: Optimization DLC model using BFGS method

```
   Input: Array Input$_3$ ($n\times|O_2|\times r$), output of Standardization layer where n is the
   no. of stocks, $|O_2|$ is the size of Standardization layer array output, and r is
   historical data size.
   Output: Array W, An array of final weights.
Begin
Let W be the vector of all weight in NNs
Let CF be the Converges Factor
For i ← 1 to r do
    $gain_i \leftarrow \frac{close_i - close_{i-1}}{close_i} * 100$
    If gaini >= 0.5 then
      X ← 1
    Else
      X ← -1
    End if
    If $close_i > close_{i-1}$ then
      $C_i \leftarrow X$
    else if $close_i = close_{i-1}$ then
      $C_i \leftarrow 0$
    else
      $C_i \leftarrow -1$
    End if
End for
k ← 0
Ridge ← 0.01
Get initial variable values $W_k$ randomly in rang (0,1)
Initialize a symmetric positive definite matrix $B_k$
    $d_k \leftarrow -B_k^{-1} g_k$
while CF does not converge to 0 do
    Calculate the Jacobian (gradient) vector $G_k$ using $W_k$, and input$_3$
    Search for α using a line search and set $W_{k+1} \leftarrow W_k + \alpha d_k$
    $G_{k+1} \leftarrow G_i + Ridge * W_{K+1}$
    $\Delta W_k \leftarrow W_{k+1} - W_k$
    $\Delta G_k \leftarrow G_{k+1} - G_k$
    $B_{k+1} \leftarrow B_k + \frac{\Delta G_k \Delta G_k^T}{\Delta G_k^T \Delta W_k} - \frac{B_k \Delta W_k \Delta W_k^T B_k}{\Delta W_k^T B_k \Delta W_k}$
    $k \leftarrow k+1$
End while
End
```

3.2.3 The Evaluation Stage

In this stage, the proposed system is evaluated based on testing dataset. RMSE and MAE measures are used to evaluate DLR model. On the other hand, Accuracy, Recall, F1-Measure, Specificity and Precision are used to evaluate DLC model. The concept of Cross-Validation is employed to evaluate the model. The procedure of training and testing is repeated ten times (i.e. 10 folds) as shown in Figure 3.8.

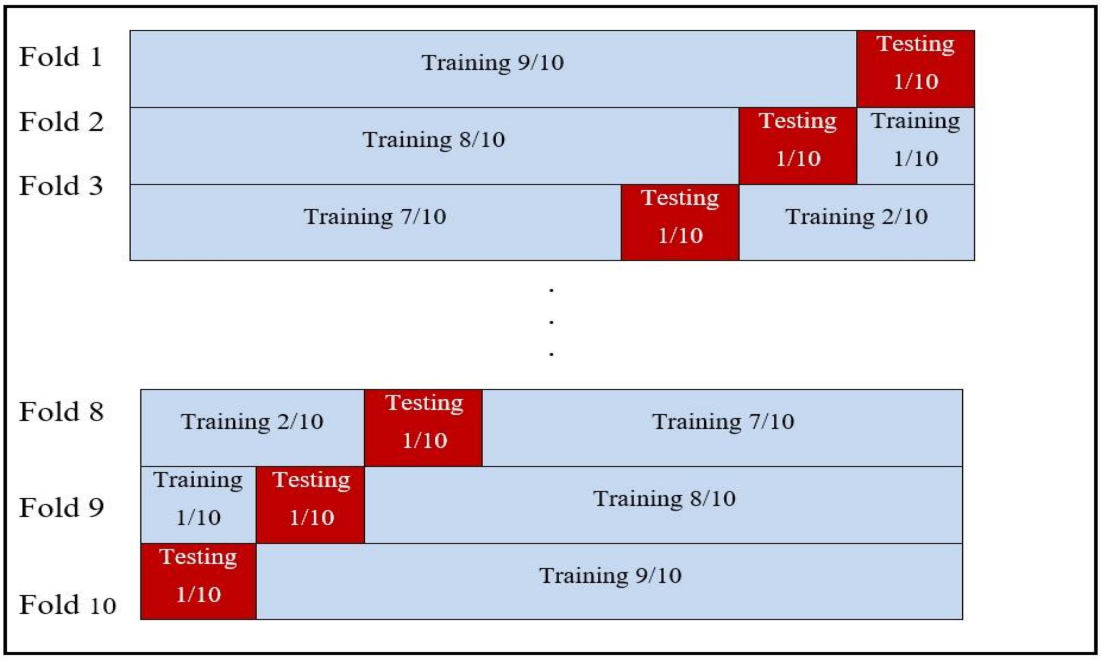


Figure 3.8. Cross-Validation with 10-Folds for SM Dataset

3.2.4 The Portfolio Construction and Optimization Stage

The suggested methodology of this stage has two major steps. The first step is selecting stocks to form the portfolio and allocating weights to each stock in the portfolio to optimize earnings. The stocks are chosen form the portfolio in a certain way that may be random, according to popularity, applying an optimization model or other. While the second step, is based on Markowitz's work on portfolio theory in 1952 and its developments that have been carried out to improve the original technique of portfolio optimization. The current dissertation relies on the topic focuses on building deep learning models that design the whole portfolio in only one step. Even though deep learning has helped to solve a wide range of problems in different areas, it still some has limitations in some branches of finance such as portfolio design. In this stage, we deal with the problem of portfolio optimization by designing a completely new deep learning model called Deep Portfolio Optimization (DPO) model.

- **Deep Portfolio Optimization Model**

The DPO model consists of four steps: calculating portfolio return, calculating portfolio risk, applying the optimization model represented by the General Reduce Gradient method, and allocating weights to stocks. The topology of the proposed DPO is shown in Figure 3.9.

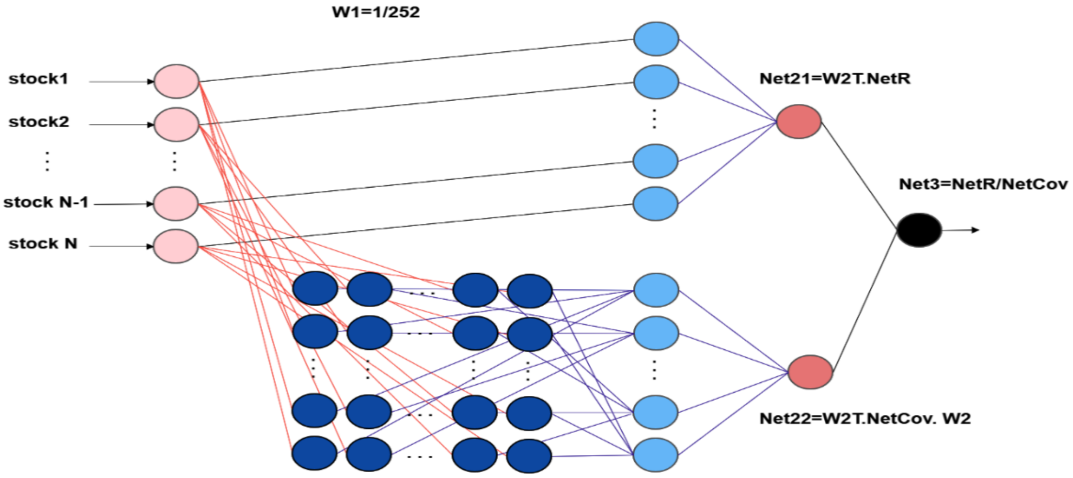


Figure 3.9. The topology DPO layer

DPO consists of four layers. Input is the first layer and it is compatible with input variables of a problem with a node for each input variable (n stocks). The second layer is a statistical layer, which is employed to calculate the return and risks of the portfolio, whereas the third layer is learning weights layer, which uses the GRG approach to train the network weights. Finally, the fourth layer is output, which is used to find and check the best Sharp Ratio.

The DPO model is described as a hybrid. This model combines the principle of diversification from the Markowitz model, which is based on calculating the correlation coefficient to find the relationship between stocks, the principle of local search (exploitation) from the GRG method, and Deep learning principle. The key actions of DPO model are outlined in Algorithm 3.6.

Algorithm 3.6: Deep Portfolio Optimization Layer

Input: *An array Input$_4$ ($n \times r$), represents stock market dataset where* **n** *is the no. of stocks, and* **r** *is historical data size.*

Output: *Array X(n), An array of final weights and Sharp ratio output neurons (O4).*

```
Begin
Let CF be the Converges Factor
Let X: vector of all weights in NNs with size n and set each cell to 1/n
Set W1 ← 1/252.
/*Find Expected Return to each stock*/
For i ← 1 to n do
    For j ← 1 to r do
        $R_{ij} \leftarrow Ln\left(\frac{Input4_{ij}}{Input4_{ij-1}}\right)$
    End for
    $ER_i \leftarrow R_i . W1_i$
End for
/*Calculate covariance matrix NetCov between all stocks*/
For i ← 1 to n do
    For j ← i+1 to n do
        $NetCov_{ij} \leftarrow \left(\frac{\sum(R_i - ER_i)(R_j - ER_j)}{n-1}\right)$
    End for
End for
While Evaluate Objective function if not satisfy do
    /*Find Expected Return to portfolio*/
    Set Net21 ← ER . X
    /*Find Risk to portfolio*/
    Net22 ← $X^T$. NetCov . X
    /*Calculate objective function according to eq. 20*/
    Net3 ← (Net21-riskFree)/ Net22
    /*update weights according to GRG method */
    X ← Procedure GRG(X)
End while
End.
```

The problem of portfolio optimization based on the classical model involves solving the quadratic programming problem for return and risk tradeoffs in order to achieve the efficient frontier, and hence the problem belongs to the NP-Complete class of problems. For this purpose, many heuristics and exact solutions have been proposed to solve the PO problem. However, these kinds of approaches require extensive CPU time to find a solution. To deal with this issue GRG approach has been employed to train the network weights. The idea lies under exploiting those updated weights and assigning them to stocks within the portfolio after the training phase.

Although the proposed DPO model dealt with the issue of CPU time, it is only suitable for long-term trading because it depends on calculating the annual rate of return using the arithmetic average. To make the DPO model able to be suitable for short-term trading, an Integrated Deep Learning (IDL) system is proposed, which combines DLR model to expect future return with DPO model to allocate optimal weights to stocks within portfolio. The advantages of the IDL system in term of portfolio optimization are summarized in Figure 3.10.

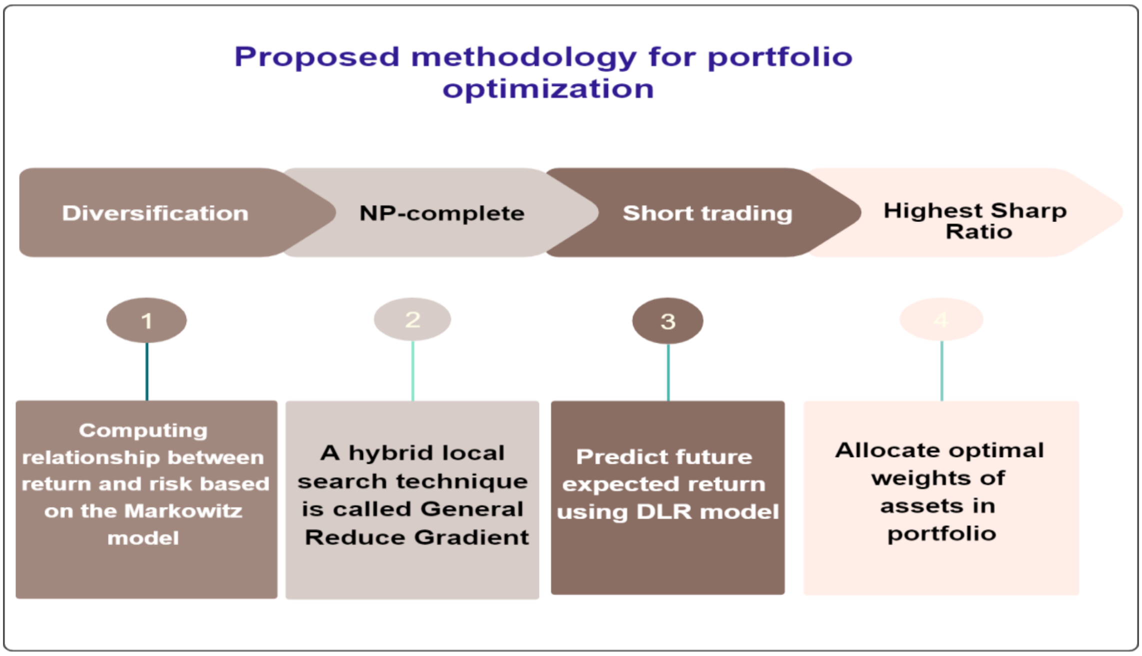

Proposed methodology for portfolio optimization
Diversification
NP-complete
Short trading
Highest Sharp Ratio
1
2
3
4
Computing relationship between return and risk based on the Markowitz model
A hybrid local search technique is called General Reduce Gradient
Predict future expected return using DLR model
Allocate optimal weights of assets in portfolio

Figure 3.10. The characteristics of portfolio optimization in IDL system

As for the accuracy of prediction, the prediction models (DLR and DLC) within the proposed IDL system are summarized by the characteristics shown in Figure 3.11. Finally, the advantages of the IDL system in terms of reducing the implementation time are summarized in Figure 3.12.

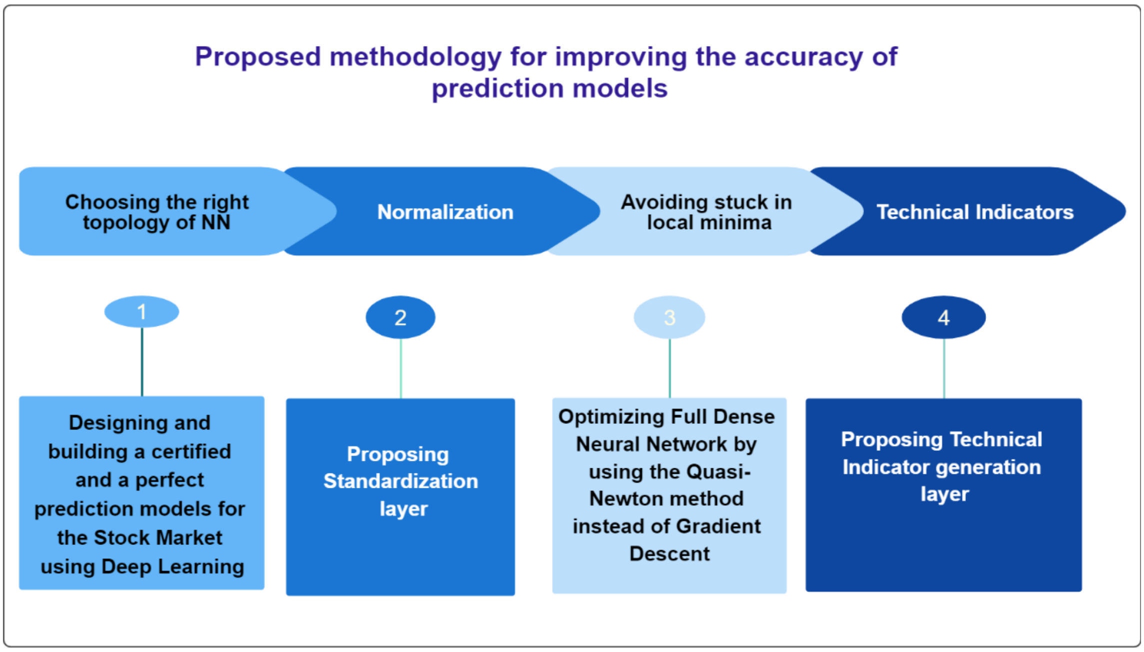

Proposed methodology for improving the accuracy of prediction models
Choosing the right topology of NN
Normalization
Avoiding stuck in local minima
Technical Indicators
1
2
3
4
Designing and building a certified and a perfect prediction models for the Stock Market using Deep Learning
Proposing Standardization layer
Optimizing Full Dense Neural Network by using the Quasi-Newton method instead of Gradient Descent
Proposing Technical Indicator generation layer

Figure 3.11. The characteristics of enhancing accuracy in IDL system

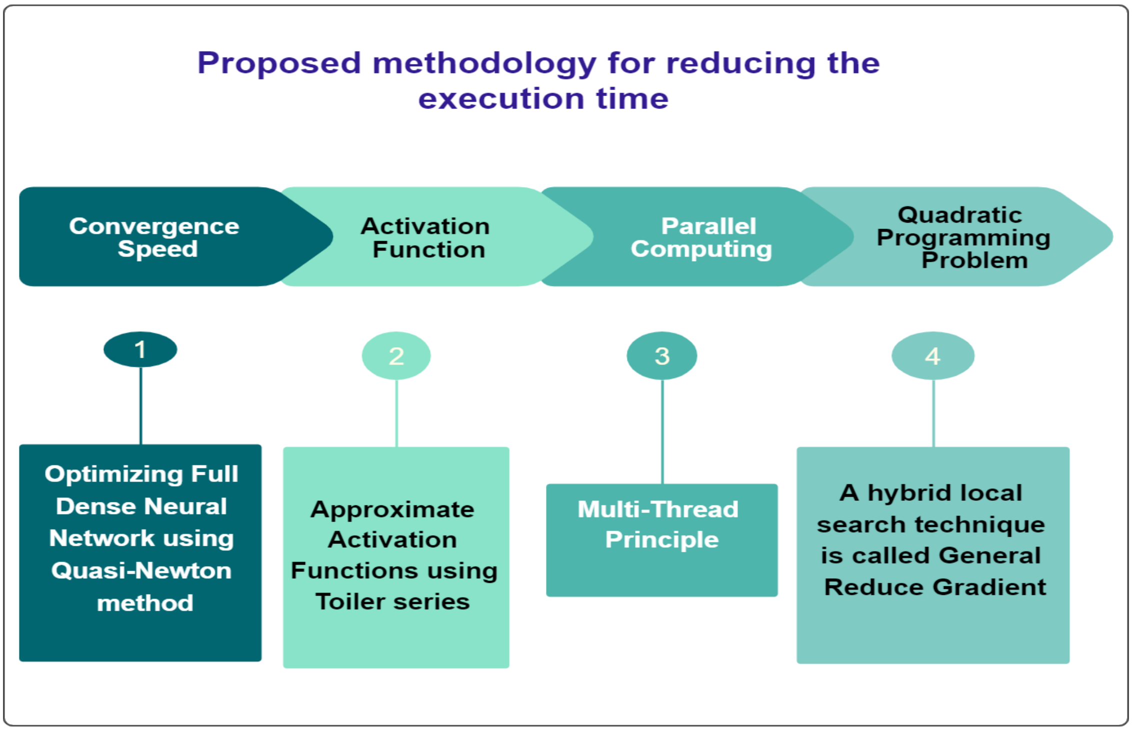

Proposed methodology for reducing the execution time
Convergence Speed
Activation Function
Parallel Computing
Quadratic Programming Problem
1
2
3
4
Optimizing Full Dense Neural Network using Quasi-Newton method
Approximate Activation Functions using Toiler series
Multi-Thread Principle
A hybrid local search technique is called General Reduce Gradient

Figure 3.12. The characteristics of enhancing execution time in IDL system

Chapter Four

Results and Discussion

Results and Discussion

4.1 Introduction

The effectiveness of the proposed system illustrated in the previous chapter has been tested with different parameters values, and the results are presented and discussed in this chapter. A real global and local dataset has been applied as an employment case study to determine the behavior of the proposed system. Furthermore, the experimental results stages of the proposed system are described and shown in this chapter. Before starting to analyze the results of the proposed system, it is important to present the general characteristics of the proposed system:

1- It deals with real, numeric, complex, and huge dataset of SM.

2- Parallelism.

3- It deals with decisions as discrete and continuous (classification, regression).

4- Optimization

  a- Optimal activation function.

  b- Optimal convergence.

  c- Optimal Portfolio

5- It depends on deep learning principals.

6- It builds rules for making decision.

7- It generates TIs to increase the accuracy of the prediction.

8- It constructs active portfolio.

9- It deals with multi-objective function (Trade-off between conflicting goals [max Return and min Risk]).

4.2 System Requirement

Hardware: Processor Intel i7, Ram 16 GB, Storage 1 TB SSD, Freq. 2.9 GHz.

Operating System: Windows 10 64 bit.

Programming Language: JAVA

IDE: Programming is done in the Eclipse IDE environment.

WEKA 3.8.5: WekaDeeplearning4j is a deep learning package for the Weka workbench. It is developed to incorporate the modern techniques of deep learning into Weka. The backend is provided by the Deeplearning4j Java library.

4.3 Description of the Stock Market Dataset

This section details the data that was extracted from the public data sources, and the final dataset that was prepared. The dissertation is based on experimental tests with historical dataset from global SM such as NYSE and local SM such as ISX.

4.3.1 The New York Stock Exchange Dataset

The New York Stock Exchange (NYSE) dataset is used in this dissertation. An updated version of the dataset can be found at (https://www.kaggle.com/dgawlik/nyse). More than 450 stocks from 1/1/2010 to 31/12/2016 are included in the dataset. Twenty-five well-known stocks were chosen in this dissertation for testing and evaluating the proposed system. The rationale behind the selection of these twenty-five stocks is that the performance of the proposed models are compared with earlier literature such as (NekoeiQachkanloo et al., 2019; Yuan Song, 2018), which used the same stocks as well. However, the proposed model can be applied to any SM.

The Twenty-five selected stocks are shown in Table 4.1. The dataset includes five technical features namely, date (the trading day), open (stock’s opening price for a specific day), low (the lowest value of the stock price during the day), high (the highest value of the stock price during the day), close (a closing stock's price on any given day), and finally, volume (volume of stock trades (buy/sell)). Figure 4.1 shows the historical closing values of the twenty-five stocks during the study period.

Table 4.1 The New York Stock Exchange dataset.

No.	Stock name	Symbol	No.	Stock name	Symbol
1	American International Group	AIG	14	IBM	IBM
2	Apple	AAPL	15	Johnson & Johnson	JNJ
3	Amazon	AMZN	16	JPMorgan Chase & Co.	JPM
4	Boeing	BA	17	Coca-Cola	KO
5	Caterpillar	CAT	18	Microsoft	MSFT
6	Capital One	COF	19	Nike	NKE
7	EBAY	EBAY	20	Oracle	ORCL
8	Ford	F	21	PepsiCo	PEP
9	FedEx	FDX	22	AT&T	T
10	General Electric	GE	23	Walmart	WMT
11	General Motors	GM	24	ExxonMobil	XOM
12	Google Alphabet Inc.	GOOG	25	Xerox	XRX
13	The Home Depot	HD			

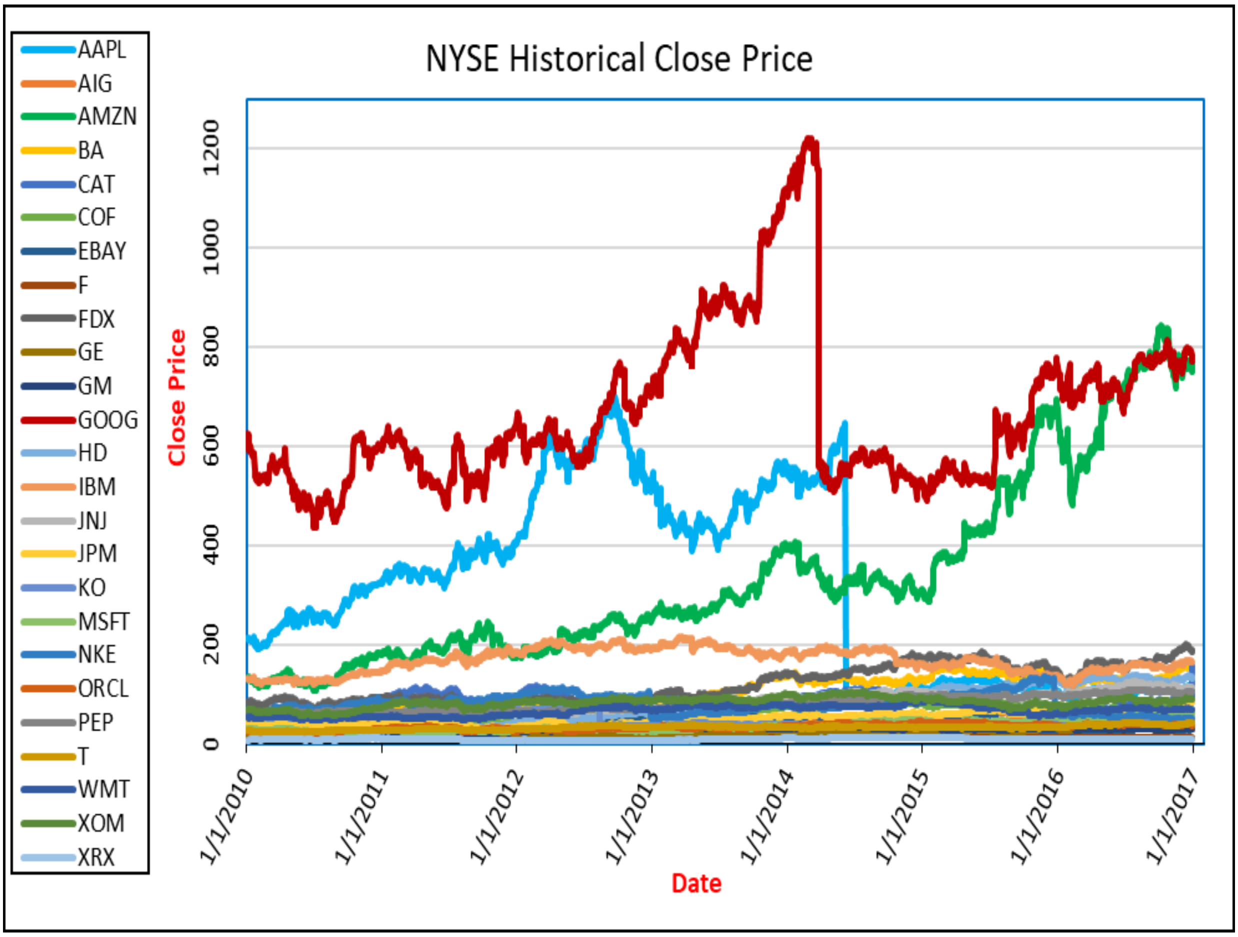


Figure 4.1. The NYSE historical close prices

4.3.2 The Iraq Stock Exchange Dataset

During the study of global stock markets, including the Iraq Stock Exchange (ISX) and their impact on the economy of the country. The dissertation observed that there is no organized dataset for SM in Iraq. However, just daily and weekly-scattered reports was founded, even though trading in the SM of Iraq began in 2004. The ISX includes 132 companies under nine sectors namely: banking, telecommunications, investment, insurance, services, agriculture, industry, hotels, and financial transfer. For this reason, a regular benchmark for ISX is constructed. Figure 4.2 shows the organizational picture of the ISX data.

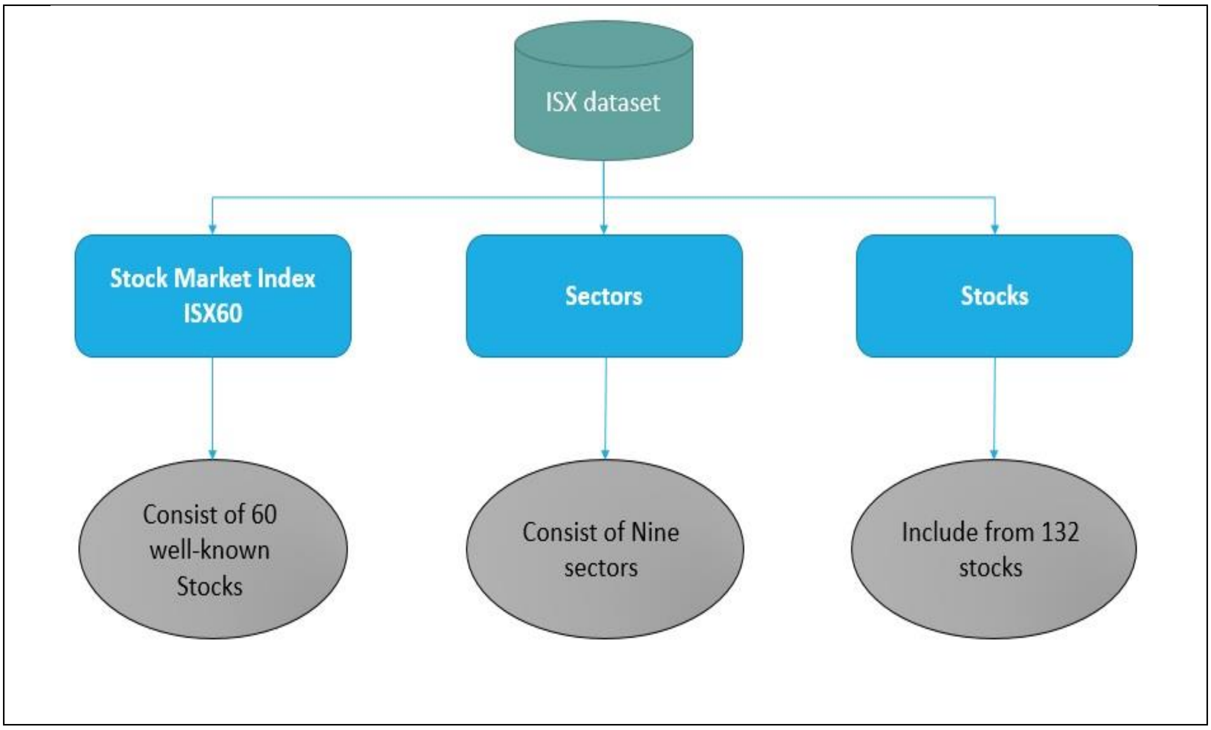
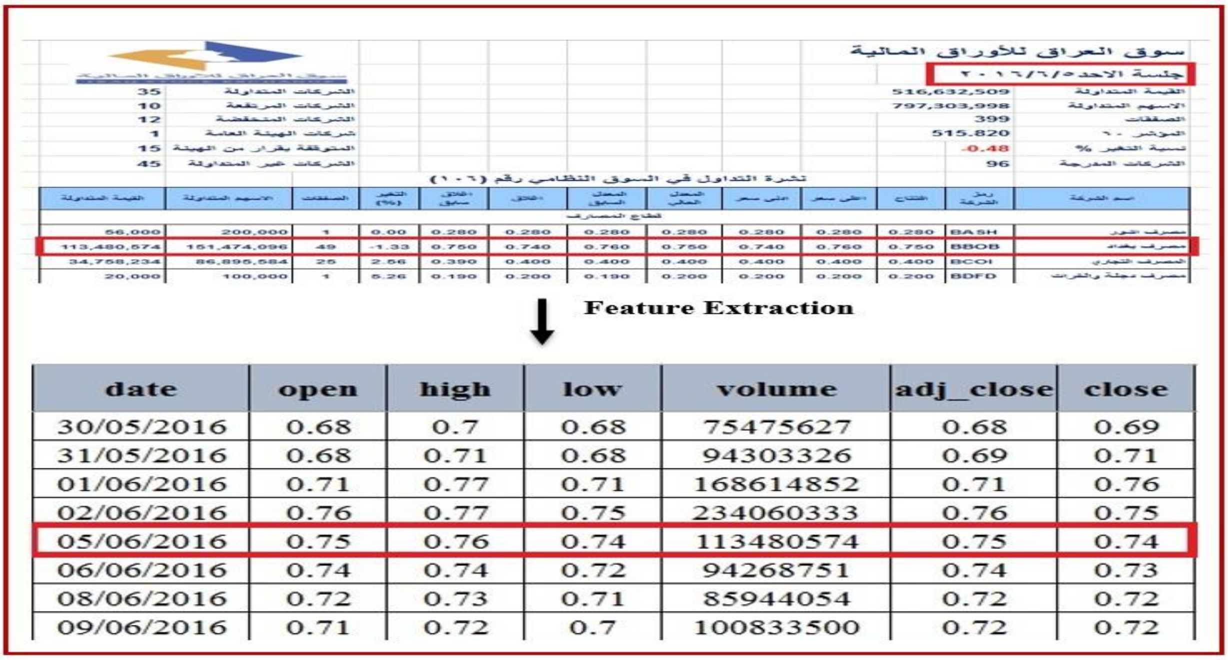


Figure 4.2. The organizational view of ISX

In the beginning, all the daily reports available to the listed companies are downloaded from the website (http://www.isx-iq.net). After collecting the daily reports in the form of Excel files, the standard features (open, high, low, close, trading volumes) are selected for all listed companies in the market as shown in Figure 4.3. This step is very tiring and required about 6 months because the work was manual.



سوق العراق للأوراق المالية

جلسة الاحد٥/٦/٢٠١٦

35	الشركات المتداولة	516,632,509	القيمة المتداولة
10	الشركات المرتفعة	797,303,998	الاسهم المتداولة
12	الشركات المنخفضة	399	الصفقات
1	شركات الهيئة العامة	515.820	المؤشر ٦٠
15	الموقوفة بقرار من الهيئة	-0.48	نسبة التغير %
45	الشركات غير المتداولة	96	الشركات المدرجة

نشرة التداول في السوق النظامي رقم (١٠٦)

القيمة المتداولة	الاسهم المتداولة	الصفقات	التغير (%)	اغلاق سابق	اغلاق	المعدل السابق	المعدل الحالي	ادنى سعر	اعلى سعر	افتتاح	رمز الشركة	اسم الشركة
قطاع المصارف												
56,000	200,000	1	0.00	0.280	0.280	0.280	0.280	0.280	0.280	0.280	BASH	مصرف اشور
113,480,574	151,474,096	49	-1.33	0.750	0.740	0.760	0.750	0.740	0.760	0.750	BBOB	مصرف بغداد
34,758,234	86,895,584	25	2.56	0.390	0.400	0.400	0.400	0.400	0.400	0.400	BCOI	المصرف التجاري
20,000	100,000	1	5.26	0.190	0.200	0.190	0.200	0.200	0.200	0.200	BDFD	مصرف دجلة والفرات

Feature Extraction

date	open	high	low	volume	adj_close	close
30/05/2016	0.68	0.7	0.68	75475627	0.68	0.69
31/05/2016	0.68	0.71	0.68	94303326	0.69	0.71
01/06/2016	0.71	0.77	0.71	168614852	0.71	0.76
02/06/2016	0.76	0.77	0.75	234060333	0.76	0.75
05/06/2016	0.75	0.76	0.74	113480574	0.75	0.74
06/06/2016	0.74	0.74	0.72	94268751	0.74	0.73
08/06/2016	0.72	0.73	0.71	85944054	0.72	0.72
09/06/2016	0.71	0.72	0.7	100833500	0.72	0.72

Figure 4.3. The feature extraction step

After creating a local database that includes all available stocks, each with five technical features as a benchmark for ISX. The step of selecting stocks as a case study for the dissertation is performed. The values of the daily prices for the stocks of the study sample companies, as well as the corresponding daily prices for the ISX general index, were chosen for the period from 9 Oct. 2013 to 21 May 2020. The beginning of the sample was chosen because the ISX was in a transitional phase towards electronic trading before 2013, and the end of the sample was the consequence of the market temporarily shutting down because of the Corona epidemic.

The dissertation population consists of all (132) equities on the ISX. The first step (stocks selection) begins with the identification of assets that fulfil the three parameters listed below:

1. Historical data for the stocks is available.
2. The stocks must be publicly traded and listed in ISX from the start of the research period until the end.
3. The stocks listed throughout the sample period should have at least 998 trading days (Average trading days for all businesses during the study period).

Following the application of the first and second conditions, the stocks are filtered to 89 stocks, and with approval of 998 trading as a minimum, the dissertation can accept 43 stocks as seen in Figure 4.4, which shows greater flexibility in building portfolios with different stocks and approaching the characteristics of the general market, as the general index of the market relied on 60 stocks. Therefore, the general market index is known as the ISX60, as seen in Figure 4.5. Table 4.2 lists the names of the companies, their symbol, sector, and the number of trading days.

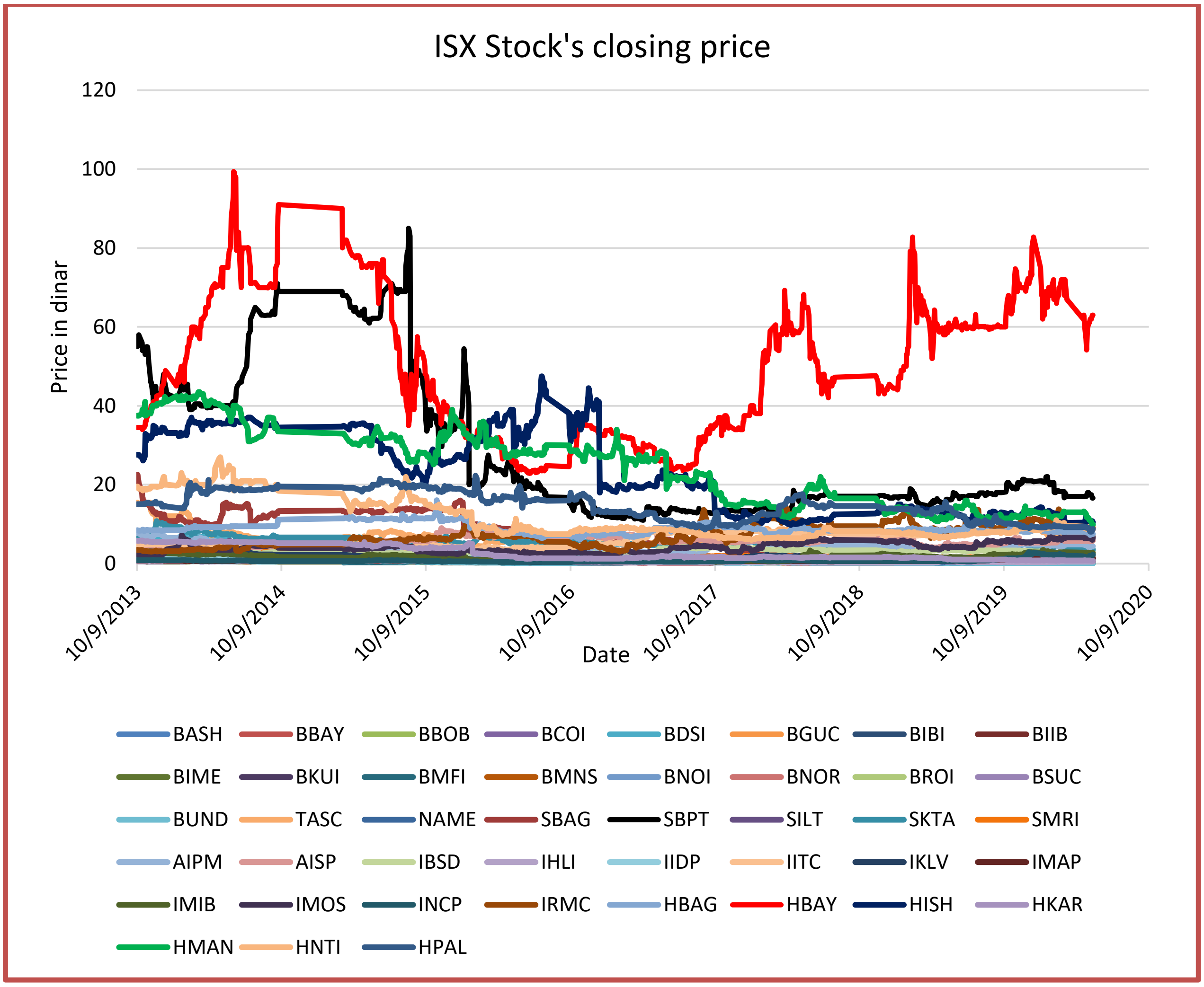


Figure 4.4. ISX stock's closing Price

Table 4.2. No. of trading Stock in ISX

#	Stock Name	Sector	Stock Symbol	Trading days	#	Stock Name	Sector	Stock Symbol	Trading days
1	Ashur International Bank	Banks	BASH	1335	23	Asia Cell	Telecom	TASC	1289
2	Babylon Bank	Banks	BBAY	1789	24	Al-Ameen Insurance	Insurance	NAME	1002
3	Bank Of Baghdad	Banks	BBOB	2221	25	Iraqi Agricultural Products Marketing Meat	Agriculture	AIPM	1674
4	Trade Bank Of Iraq	Banks	BCOI	2042	26	Iraqi for Seed Production	Agriculture	AISP	1303
5	Dar Al Salam Investment Bank	Banks	BDSI	1336	27	Baghdad Soft Drinks	Industry	IBSD	2035
6	Gulf Commercial Bank	Banks	BGUC	2005	28	Al-Hilal Industries	Industry	IHLI	1548
7	Iraqi Investment Bank	Banks	BIBI	2050	29	Iraqi Date Processing and Marketing	Industry	IIDP	1285
8	Iraqi Islamic Bank	Banks	BIIB	1473	30	Iraqi For Tufted Carpets	Industry	IITC	1451
9	Iraqi the Middle East Investment	Banks	BIME	1958	31	Al- Kindi of Veterinary Vaccines Drugs	Industry	IKLV	1683
10	Kurdistan International Bank	Banks	BKUI	1143	32	Al-Mansour Pharmaceuticals Industries	Industry	IMAP	1715
11	Mosul Bank for Development & Investment	Banks	BMFI	1631	33	Metallic & Bicycles Industries	Industry	IMIB	1396
12	Mansour bank for investment	Banks	BMNS	1796	34	Modern Sewing	Industry	IMOS	1407
13	National Bank of Iraq	Banks	BNOI	1361	35	National Chemical & Plastic Industries	Industry	INCP	1926
14	North Bank	Banks	BNOR	1226	36	Ready-Made Clothes	Industry	IRMC	1184
15	Credit Bank of Iraq	Banks	BROI	1816	37	Baghdad Hotel	Tourism & Hotels	HBAG	1566
16	Sumer Commercial Bank	Banks	BSUC	1179	38	Babylon Hotel	Tourism & Hotels	HBAY	1784
17	United Bank for Investment in Iraq	Banks	BUND	1925	39	Ishtar Hotels	Tourism & Hotels	HISH	1269
18	AL-Badia for General Trans	Services	SBAG	1143	40	Karbala Hotels	Tourism & Hotels	HKAR	1347
19	Iraq Baghdad for General Transport	Services	SBPT	1214	41	Al-Mansour Hotels	Tourism & Hotels	HMAN	1247
20	Iraqi Land Transport	Services	SILT	1577	42	National Company for Tourism Investment	Tourism & Hotels	HNTI	1758
21	Kharkh Tour Amusement City	Services	SKTA	1554	43	Palestine Hotel	Tourism & Hotels	HPAL	1258
22	Mamoura Real-estate Investment	Services	SMRI	2222					

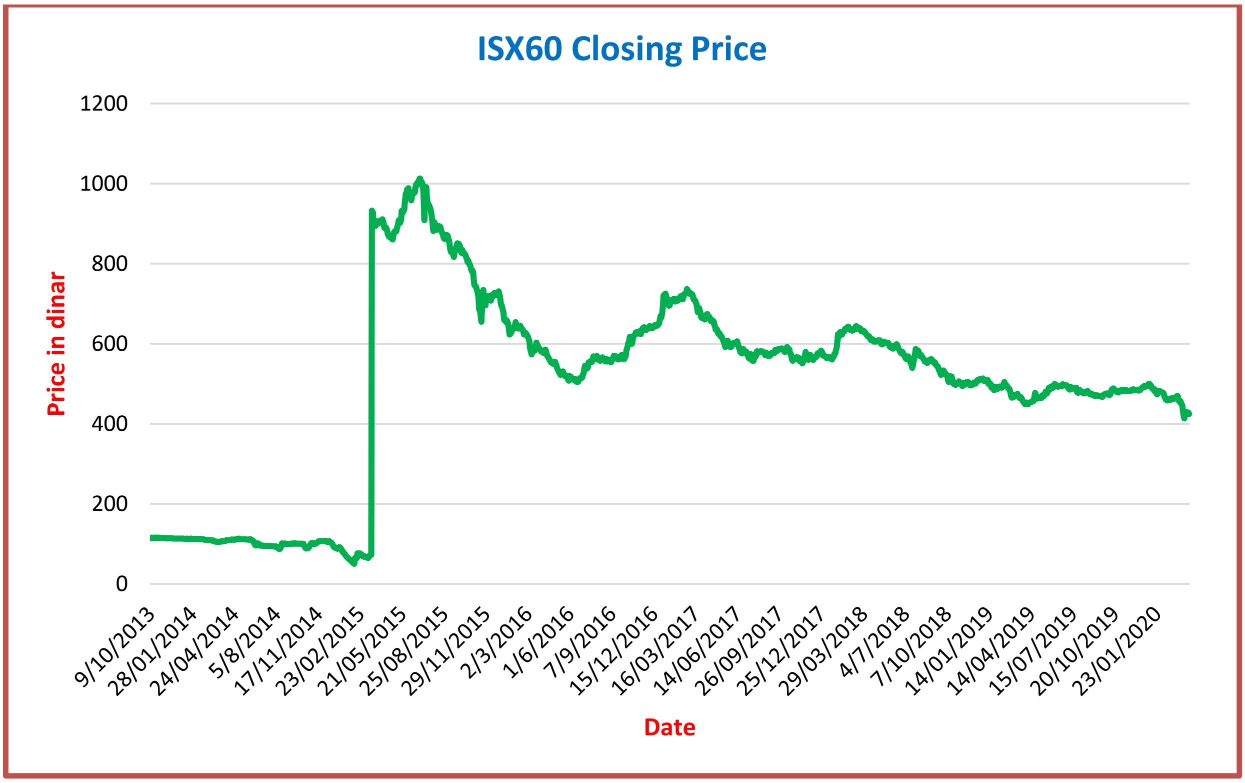


Figure 4.5. ISX60 closing Price

4.4 The Preprocessing Stage

The goal of this stage is to prepare the data for the prediction and the portfolio optimization stages. Steps of the preprocessing stage, which are explained in Section 3.2.1 are applied only to the Iraq SM dataset. After applying this stage, the database contains 43 companies for the Iraq Stock Exchange and 25 companies for the New York Stock Exchange. This data is stored in the form of a csv file to be used as an input for the next stages.

4.5 Evaluating the Classification Model

The specific product price or commodity in the SM is represented as a continuous value. Class labeling is used as a method to demonstrate close price in an easier form. The classification process is improved using this representation. Thus, before evaluating the classification models, class feature should be created with three values (buy, sell and hold) based on equations (3.7) and (3.8) for all stocks in database. The best value of gain is selected equal to (5%) because the change price in ISX was very small. In the local database, new 43 files for ISX and 25 files for NYSE are created after insert class feature to old files in the database.

For data training, the 10-fold Cross-Validation testing method is adopted. The dataset is divided into K (10) equal subsets where each record or subset uses the K−1 (9) of times for training and exactly once for testing. The total error is calculated by aggregating the errors found through all K runs divided by K as the average across runs. The aim of using this approach is to determine the best set of parameters (weights and biases) that can fit the DNN.

4.5.1 The Results of ISX

Initially, the DLC model (only with Standardization and Prediction layers) is implemented using Quasi Newton (GN) training model and unipolar activation functions on ISX60 general index to find the best results in terms of Accuracy, Precision, Recall, F1-measure, Specificity, and execution time. After that, the MLP using DeepLearning4J training by Gradient Decent (GD) method is also implemented on the same dataset for comparison. Table 4.3 shows the results of the evaluation measures after implementing DLC and MLP using DeepLearning4J models on ISX60 dataset.

Table 4.3. Evaluation criteria results of ISX60

Classifier Model	Accuracy	Precision	Recall	F1-measure	Specificity	Exe. time in seconds
MLP using DeepLearning4J	54.6729	54.7	54.7	70.7	54.8	18.79
The proposed DLC	**98.36**	**98.30**	**98.62**	**98.46**	**98.93**	**0.489**

According to Table 4.3, two important issues can be argued. The first relates to the results of the two models in terms of accuracy and execution time depending on the training method, whereas the second relates to evaluation criteria. Regarding the first issue, the prediction accuracy was improved by a large percentage (79.9%) when replacing the training method (GN) instead of (GD), as well as in terms of execution time there is also an improvement, as the improvement rate was 38 times lower than the previous execution time. The reason for this is due to the speed of convergence in the GN method and its avoidance of falling into the local minima.

As for the second issue, the four evaluation criteria (Precision, Recall, F1-measure, Specificity) described in section (2.8.1) have advantages and disadvantages that have been explained in the said section, and it was shown that the Accuracy measure has outweighed most of the defects of the four criteria. Therefore, the Accuracy measure will be adopted only to evaluate the results of classification models.

After that, the DLC model is implemented using GN training method and unipolar sigmoid activation functions on 43 stocks in the ISX dataset to find the best results in terms of Accuracy and execution time. After that the MLP using DeepLearning4J training by Gradient Decent (GD) method is also implemented on the same dataset for comparison. Table 4.4 shows the results of evaluation measures after implementing DLC and MLP using DeepLearning4J models on 43 stocks in the ISX dataset.

Table 4.4. Evaluation criteria results for ISX dataset

Stocks	Evaluation criteria	DLC	MLP using DeepLearning4J	Stocks	Evaluation criteria	DLC	MLP using DeepLearning4J
BASH	**Accuracy**	99.276	46.50	**TASC**	**Accuracy**	92.761	43.46
	Exe. time	0.644	6.43		**Exe. time**	0.533	6.59
BBAY	**Accuracy**	98.55	62.31	**NAME**	**Accuracy**	97.308	54.57
	Exe. time	0.388	6.38		**Exe. time**	0.578	6.36
BBOB	**Accuracy**	98.55	34.0426	**AIPM**	**Accuracy**	99.276	42.55
	Exe. time	0.544	6.51		**Exe. time**	0.453	7.08
BCOI	**Accuracy**	96.48	48.02	**AISP**	**Accuracy**	99.37	46.50
	Exe. time	0.396	6.65		**Exe. time**	0.425	6.47
BDSI	**Accuracy**	91.31	44.98	**IBSD**	**Accuracy**	99.069	50.45
	Exe. time	0.486	6.72		**Exe. time**	0.458	6.64
BGUC	**Accuracy**	95.243	59.574	**IHLI**	**Accuracy**	99.069	55.01
	Exe. time	0.481	6.76		**Exe. time**	0.431	6.59
BIBI	**Accuracy**	92.24	45.28	**IIDP**	**Accuracy**	98.655	40.12
	Exe. time	0.542	6.5		**Exe. time**	0.426	7.35
BIIB	**Accuracy**	87.90	55.62	**IITC**	**Accuracy**	95.553	50.75
	Exe. time	0.257	8.2		**Exe. time**	0.403	6.36
BIME	**Accuracy**	98.96	63.52	**IKLV**	**Accuracy**	98.759	39.51
	Exe. time	0.481	6.48		**Exe. time**	0.574	6.37
BKUI	**Accuracy**	95.243	57.75	**IMAP**	**Accuracy**	99.482	53.19
	Exe. time	0.446	6.31		**Exe. time**	0.435	6.76
BMFI	**Accuracy**	99.482	51.97	**IMIB**	**Accuracy**	97.724	37.99
	Exe. time	0.417	6.23		**Exe. time**	0.455	6.5
BMNS	**Accuracy**	96.38	54.10	**IMOS**	**Accuracy**	99.1726	41.03
	Exe. time	0.438	6.42		**Exe. time**	0.507	6.42
BNOI	**Accuracy**	94.51	46.20	**INCP**	**Accuracy**	96.38	55.92
	Exe. time	0.46	6.47		**Exe. time**	0.478	6.53
BNOR	**Accuracy**	98.96	35.56	**IRMC**	**Accuracy**	96.07	49.84
	Exe. time	0.45	6.87		**Exe. time**	0.428	6.59
BROI	**Accuracy**	98.55	59.87	**HBAG**	**Accuracy**	96.173	41.64
	Exe. time	0.602	6.66		**Exe. time**	0.491	6.39
BSUC	**Accuracy**	91.31	82.67	**HBAY**	**Accuracy**	96.587	36.77
	Exe. time	0.247	6.62		**Exe. time**	0.488	6.34
BUND	**Accuracy**	98.44	59.57	**HISH**	**Accuracy**	94.6225	41.94
	Exe. time	0.543	6.39		**Exe. time**	0.459	6.3
SBAG	**Accuracy**	93.89	42.55	**HKAR**	**Accuracy**	74.203	41.56
	Exe. time	0.397	6.62		**Exe. time**	0.469	5.95
SBPT	**Accuracy**	99.482	53.19	**HMAN**	**Accuracy**	92.967	40.42
	Exe. time	0.422	6.54		**Exe. time**	0.324	6.31
SILT	**Accuracy**	95.24	44.68	**HNTI**	**Accuracy**	88.6246	40.42
	Exe. time	0.46	6.62		**Exe. time**	0.472	6.36
SKTA	**Accuracy**	96.173	43.46	**HPAL**	**Accuracy**	92.761	43.46
	Exe. time	0.442	7.5		**Exe. time**	0.514	6.64
SMRI	**Accuracy**	98.86	47.41				
	Exe. time	0.669	6.34				

Table 4.4 shows the terrible superiority of the Accuracy results and the execution time of the DLC model. To analyze these results, Table 4.5 presents a summary of them. According to Table 4.5, the Accuracy average obtained for 43 stocks is (95.805) while the highest accuracy is (99.482) for stocks (BMFI, SBPT and IMAP) and the lowest accuracy is (74.203) for a stock (HKAR). As for the execution time, the average execution time is (0.465) second and the most time taken is (0.669) second for the stock (SMRI) second and the least was for the stock (BSUC).

Table 4.5. Analysis the results of Table 4.4

	Accuracy			**Execution time in second**		
	DLC model	**MLP using DeepLearning4J model**	**Accuracy Improvement rate**	**DCL model**	**MLP using DeepLearning4J model**	**improvement rate times**
Average	**95.805**	48.50	104%	0.465	6.584	14.75
Max	**99.482**	82.67	189%	0.669	8.2	31.90
Min	**74.203**	34.04	10%	0.247	5.95	9.476

The percentage of the total Accuracy improvement rate is (104%) for 43 companies, while the highest improvement for the stock (BBOB) with a percentage (189%) and the lowest improvement for the stock (BSUC) with (10%). However, in terms of execution time, the overall improvement rate is (14.75) times better, while the highest improvement is (31.90) times for a stock (BIIB), and the least improvement is (9.476) times for a stock (SMRI).

Despite the satisfactory results of the DLC model, some companies had relatively poor predictive accuracy. For this reason, the TIs generation layer is added to improve the accuracy of these companies, as shown in Table 4.6.

Table 4.6. The DLC results with and without TIs generation layer for ISX

Stocks	Evaluation criteria	DLC without TIs layer	DLC with TIs layer	Stocks	Evaluation criteria	DLC without TIs layer	DLC with TIs layer
BASH	Accuracy	99.27	99.27	TASC	Accuracy	92.761	99.37
	Exe. time	0.644	0.642		Exe. time	0.533	0.688
BBAY	Accuracy	98.55	98.34	NAME	Accuracy	97.308	97.204
	Exe. time	0.388	0.635		Exe. time	0.578	0.657
BBOB	Accuracy	98.55	98.65	AIPM	Accuracy	99.276	99.172
	Exe. time	0.544	0.615		Exe. time	0.453	0.645
BCOI	Accuracy	96.48	99.27	AISP	Accuracy	99.37	99.06
	Exe. time	0.396	0.618		Exe. time	0.425	0.472
BDSI	Accuracy	91.31	97.51	IBSD	Accuracy	99.069	99.069
	Exe. time	0.486	0.575		Exe. time	0.458	0.598
BGUC	Accuracy	95.243	99.06	IHLI	Accuracy	99.069	99.069
	Exe. time	0.481	0.516		Exe. time	0.431	0.692
BIBI	Accuracy	92.24	98.96	IIDP	Accuracy	98.655	98.655
	Exe. time	0.542	0.699		Exe. time	0.426	0.633
BIIB	Accuracy	87.90	98.96	IITC	Accuracy	95.553	98.448
	Exe. time	0.257	0.573		Exe. time	0.403	0.627
BIME	Accuracy	98.96	98.96	IKLV	Accuracy	98.759	98.759
	Exe. time	0.481	0.514		Exe. time	0.574	0.51
BKUI	Accuracy	95.243	98.241	IMAP	Accuracy	99.482	99.482
	Exe. time	0.446	0.641		Exe. time	0.422	0.505
BMFI	Accuracy	99.482	99.379	IMIB	Accuracy	97.724	97.621
	Exe. time	0.417	0.63		Exe. time	0.455	0.493
BMNS	Accuracy	96.38	99.06	IMOS	Accuracy	99.1726	98.86
	Exe. time	0.438	0.479		Exe. time	0.507	0.601
BNOI	Accuracy	94.51	97.20	INCP	Accuracy	96.38	99.172
	Exe. time	0.46	0.514		Exe. time	0.478	0.652
BNOR	Accuracy	98.96	98.96	IRMC	Accuracy	96.07	95.966
	Exe. time	0.45	0.665		Exe. time	0.428	0.511
BROI	Accuracy	98.55	98.55	HBAG	Accuracy	96.173	96.380
	Exe. time	0.602	0.637		Exe. time	0.491	0.496
BSUC	Accuracy	91.31	96.38	HBAY	Accuracy	96.587	96.380
	Exe. time	0.247	0.632		Exe. time	0.488	0.466
BUND	Accuracy	98.44	98.44	HISH	Accuracy	94.6225	94.725
	Exe. time	0.543	0.612		Exe. time	0.459	0.62
SBAG	Accuracy	93.89	93.07	HKAR	Accuracy	74.203	96.072
	Exe. time	0.397	0.507		Exe. time	0.469	0.545
SBPT	Accuracy	99.48	99.48	HMAN	Accuracy	92.967	95.553
	Exe. time	0.422	0.602		Exe. time	0.324	0.648
SILT	Accuracy	95.24	98.86	HNTI	Accuracy	88.6246	95.1396
	Exe. time	0.46	0.569		Exe. time	0.472	0.596
SKTA	Accuracy	96.173	99.37	HPAL	Accuracy	92.761	92.657
	Exe. time	0.442	0.614		Exe. time	0.514	0.489
SMRI	Accuracy	98.86	98.75				
	Exe. time	0.6	0.631				

According to Table 4.6, the accuracy results are improved by adding a TIs generation layer, especially for the stocks that are suffering from a deterioration in accuracy. The number of improved stocks is 20 and is marked in green, and the number of stocks whose accuracy has decreased is 12 and is marked in blue while the number of stocks that have kept their results is 11. The results of Table 4.6 are analyzed and shown in Table 4.7 to give a summary of them.

Table 4.7. Analysis the results of Table 4.6

	Accuracy			Execution time in second		
	DLC Without TIs layer	DLC With TIs layer	Accuracy Improvement	DLC Without TIs layer	DLC With TIs layer	improvement rate times
Average	95.804	97.942	2.1378	0.463	0.587	-0.12402
Max	99.482	99.482	21.869	0.644	0.699	0.064
Min	74.203	92.657	-0.82	0.247	0.466	-0.385

According to Table 4.7 in term of accuracy, the accuracy average for all 43 stocks was (95.804) and became (97.942) after adding the TIs generation layer to the DLC model with an improvement rate of (2.1378). The DLC model also maintained the best accuracy of (99.482), while the lowest accuracy improved, which was (74,203) for a stock (HKAR) and became (92,657) with an improvement rate of (21.869). Notwithstanding the good improvements that have been seen, but some stocks have decreased in accuracy, but that deterioration is not significant, as the rate of deterioration has reached (0.215) and the most significant deterioration was for stock (SBAG) by (-0.82). On the other hand, for the execution time, the time increased by adding the TIs generation layer by a small percentage amounting to (0.12402) to the total stocks, as shown in Table 4.7.

After presenting and analyzing the results in Table 4.7, the DLC model with the addition of the TIs generation layer proved its efficiency in terms of prediction accuracy but at the expense of execution time. As a result, the activation functions have been approximated to reduce the execution time with an accuracy very close to the original sigmoid activation function. Table 4.8 illustrates the accuracy and execution time results for the DLC model after replacing unipolar sigmoid activation functions with a proposed approximation unipolar sigmoid activation functions based on Eq. (3.5).

Table 4.8. The DLC results with approximation activation function for ISX

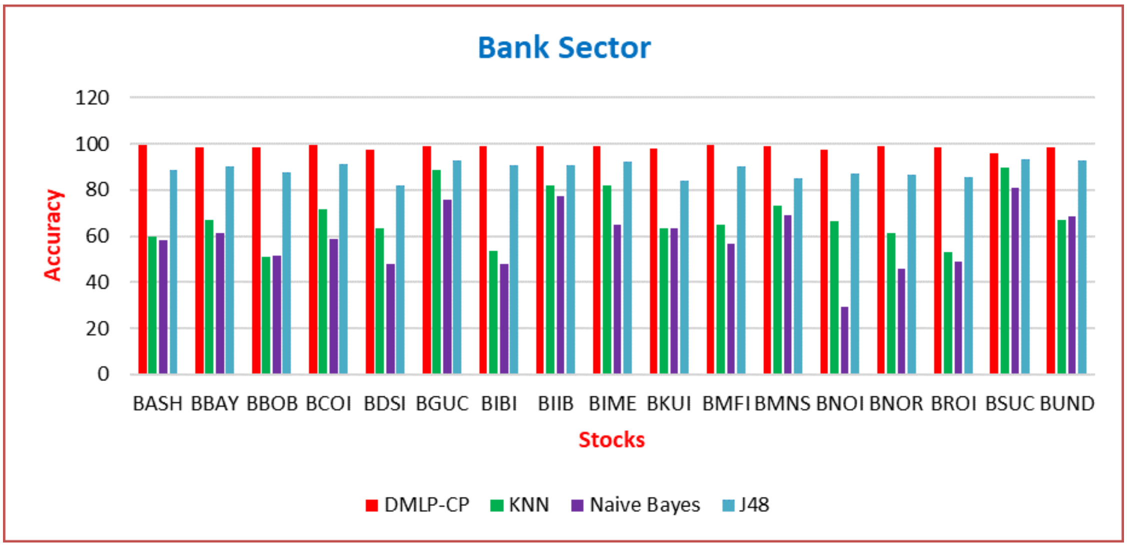
Stocks	Evaluation criteria	Unipolar activation function	Appx. unipolar activation function	Stocks	Evaluation criteria	Unipolar activation function	Appx. unipolar activation function
BASH	Accuracy	99.27	99.27	TASC	Accuracy	99.37	99.37
	Exe. time	0.642	**0.251**		Exe. time	0.688	**0.225**
BBAY	Accuracy	98.34	98.34	NAME	Accuracy	97.204	97.204
	Exe. time	0.635	**0.249**		Exe. time	0.657	**0.252**
BBOB	Accuracy	98.65	98.55	AIPM	Accuracy	99.172	99.069
	Exe. time	0.615	**0.263**		Exe. time	0.645	**0.274**
BCOI	Accuracy	99.27	99.27	AISP	Accuracy	99.06	99.06
	Exe. time	0.618	**0.332**		Exe. time	0.472	**0.179**
BDSI	Accuracy	97.51	97.51	IBSD	Accuracy	99.069	99.172
	Exe. time	0.575	**0.226**		Exe. time	0.598	**0.231**
BGUC	Accuracy	99.06	99.06	IHLI	Accuracy	99.069	99.069
	Exe. time	0.516	**0.179**		Exe. time	0.692	**0.267**
BIBI	Accuracy	98.96	98.96	IIDP	Accuracy	98.655	98.655
	Exe. time	0.699	**0.34**		Exe. time	0.633	**0.255**
BIIB	Accuracy	98.96	99.069	IITC	Accuracy	98.448	98.448
	Exe. time	0.573	**0.243**		Exe. time	0.627	**0.25**
BIME	Accuracy	98.96	98.862	IKLV	Accuracy	98.759	98.759
	Exe. time	0.514	**0.247**		Exe. time	0.51	**0.179**
BKUI	Accuracy	98.241	98.241	IMAP	Accuracy	99.482	99.482
	Exe. time	0.641	**0.28**		Exe. time	0.505	**0.174**
BMFI	Accuracy	99.379	99.379	IMIB	Accuracy	97.621	97.621
	Exe. time	0.63	**0.231**		Exe. time	0.493	**0.174**
BMNS	Accuracy	99.06	99.06	IMOS	Accuracy	98.86	98.86
	Exe. time	0.479	**0.152**		Exe. time	0.601	**0.18**
BNOI	Accuracy	97.20	97.104	INCP	Accuracy	99.172	99.069
	Exe. time	0.514	**0.2**		Exe. time	0.652	**0.241**
BNOR	Accuracy	98.96	98.96	IRMC	Accuracy	95.966	95.966
	Exe. time	0.665	**0.266**		Exe. time	0.511	**0.239**
BROI	Accuracy	98.55	98.55	HBAG	Accuracy	96.38	96.38
	Exe. time	0.637	**0.259**		Exe. time	0.496	**0.168**
BSUC	Accuracy	96.38	97.10	HBAY	Accuracy	96.380	96.587
	Exe. time	0.632	**0.246**		Exe. time	0.466	**0.172**
BUND	Accuracy	98.44	98.44	HISH	Accuracy	94.725	94.725
	Exe. time	0.612	**0.263**		Exe. time	0.62	**0.242**
SBAG	Accuracy	93.07	93.07	HKAR	Accuracy	96.072	95.859
	Exe. time	0.507	**0.181**		Exe. time	0.545	**0.179**
SBPT	Accuracy	99.48	99.48	HMAN	Accuracy	95.553	95.553
	Exe. time	0.602	**0.171**		Exe. time	0.648	**0.262**
SILT	Accuracy	98.86	98.86	HNTI	Accuracy	95.1396	95.1396
	Exe. time	0.569	**0.18**		Exe. time	0.596	**0.261**
SKTA	Accuracy	99.37	99.37	HPAL	Accuracy	92.657	92.554
	Exe. time	0.614	**0.183**		Exe. time	0.489	**0.16**
SMRI	Accuracy	98.75	98.75				
	Exe. time	0.631	**0.262**				

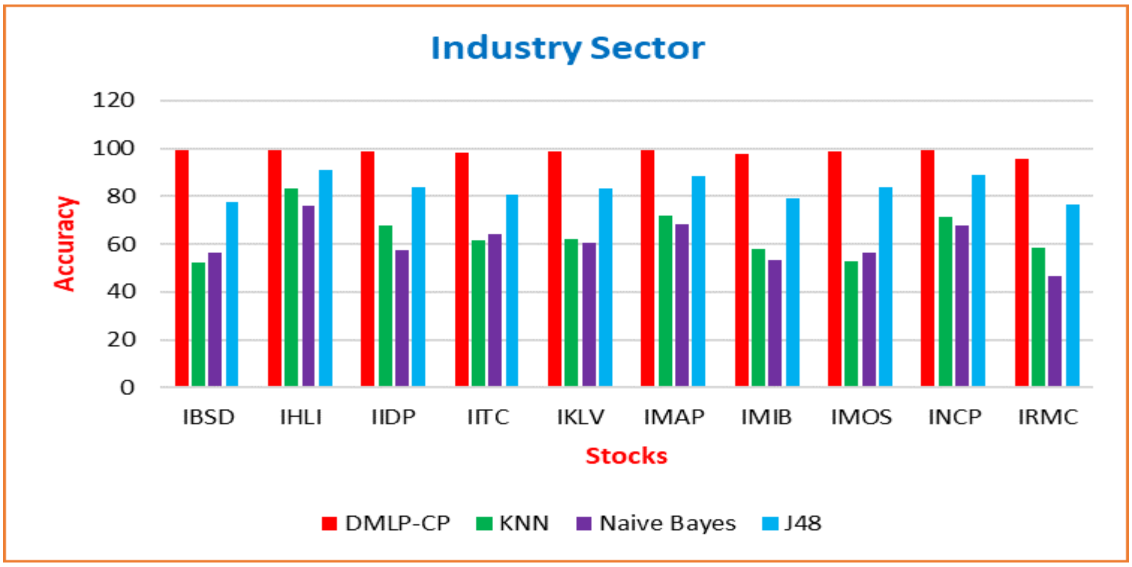
According to Table 4.8, the execution time is significantly improved when using approximated activation function, while maintaining an accuracy very similar to that obtained from the original sigmoid activation function.

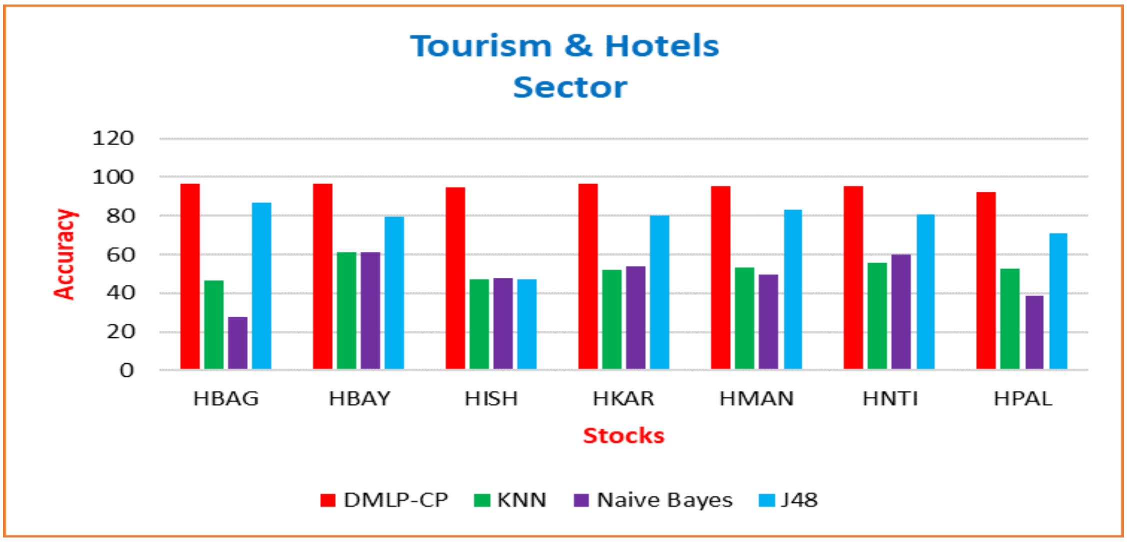
After displaying and analyzing the results in tables from 4.4 to 4.8 above, the DLC model with the addition of a TIs generation layer and approximation activation function proved its efficiency in terms of accuracy and execution time. To further confirm the performance of the DLC model, it is compared with three standard classifiers: KNN, J48 DT and Naïve Bays. The accuracy results of these models are summarized in Table 4.9. For the more Figure 4.6 visuals the accuracy bar chart for all seven sectors in ISX after applying four classification models.

Table 4.9. The accuracy comparison results for ISX

Stocks	DLC	KNN	Naive Bayes	J48	Stocks	DLC	KNN	Naive Bayes	J48
BASH	**99.27**	59.8759	58.4281	88.831	**TASC**	**99.37**	60.393	56.0496	78.594
BBAY	**98.34**	66.8693	61.3982	90.274	**NAME**	**97.204**	65.2174	63.9752	82.919
BBOB	**98.65**	51.0638	51.3678	87.842	**AIPM**	**99.172**	60.2896	58.0145	82.937
BCOI	**99.27**	71.5615	58.7384	91.107	**AISP**	**99.06**	50.3619	46.8459	76.008
BDSI	**97.51**	63.2219	48.0243	82.067	**IBSD**	**99.069**	52.5836	56.535	77.812
BGUC	**99.06**	88.7538	75.9878	93.009	**IHLI**	**99.069**	83.5574	75.9878	90.9
BIBI	**98.96**	53.4954	48.0243	90.882	**IIDP**	**98.655**	68.0455	57.4974	84.075
BIIB	**98.96**	82.0062	77.456	90.693	**IITC**	**98.448**	61.5305	64.0124	80.972
BIME	**98.96**	82.0669	64.7416	92.097	**IKLV**	**98.759**	62.3578	60.4964	83.351
BKUI	**98.241**	63.5258	63.2219	83.891	**IMAP**	**99.482**	72.182	68.5626	88.521
BMFI	**99.379**	64.7416	56.8389	89.97	**IMIB**	**97.621**	58.0145	53.3609	79.318
BMNS	**99.06**	72.9483	68.997	85.106	**IMOS**	**98.86**	52.7404	56.6701	83.661
BNOI	**97.2**	66.3909	29.576	86.867	**INCP**	**99.172**	71.4286	67.7812	88.754
BNOR	**98.96**	61.0942	45.8967	86.626	**IRMC**	**95.966**	58.3587	46.5046	76.596
BROI	**98.55**	53.1915	48.9362	85.714	**HBAG**	**96.38**	46.8085	27.9635	86.93
BSUC	**96.38**	89.6657	80.8511	93.313	**HBAY**	**96.38**	61.3982	61.3982	79.331
BUND	**98.44**	66.8693	68.3891	92.705	**HISH**	**94.725**	47.4164	47.7204	47.416
SBAG	**93.07**	51.7063	44.8811	73.113	**HKAR**	**96.072**	52.3659	53.6278	79.811
SBPT	**99.48**	47.4164	49.848	79.939	**HMAN**	**95.553**	53.1915	49.5441	83.283
SILT	**98.86**	56.6701	39.5036	80.145	**HNTI**	**95.1396**	55.6231	59.8784	80.851
SKTA	**99.37**	55.3191	54.7112	76.292	**HPAL**	**92.657**	52.5836	38.9058	70.821
SMRI	**98.75**	66.1841	60.1861	81.489					


Bank Sector
Accuracy
120
100
80
60
40
20
0
BASH BBAY BBOB BCOI BDSI BGUC BIBI BIIB BIME BKUI BMFI BMNS BNOI BNOR BROI BSUC BUND
Stocks
DMLP-CP
KNN
Naive Bayes
J48


Industry Sector
Accuracy
120
100
80
60
40
20
0
IBSD IHLI IIDP IITC IKLV IMAP IMIB IMOS INCP IRMC
Stocks
DMLP-CP
KNN
Naive Bayes
J48


Tourism & Hotels Sector
Accuracy
120
100
80
60
40
20
0
HBAG HBAY HISH HKAR HMAN HNTI HPAL
Stocks
DMLP-CP
KNN
Naive Bayes
J48

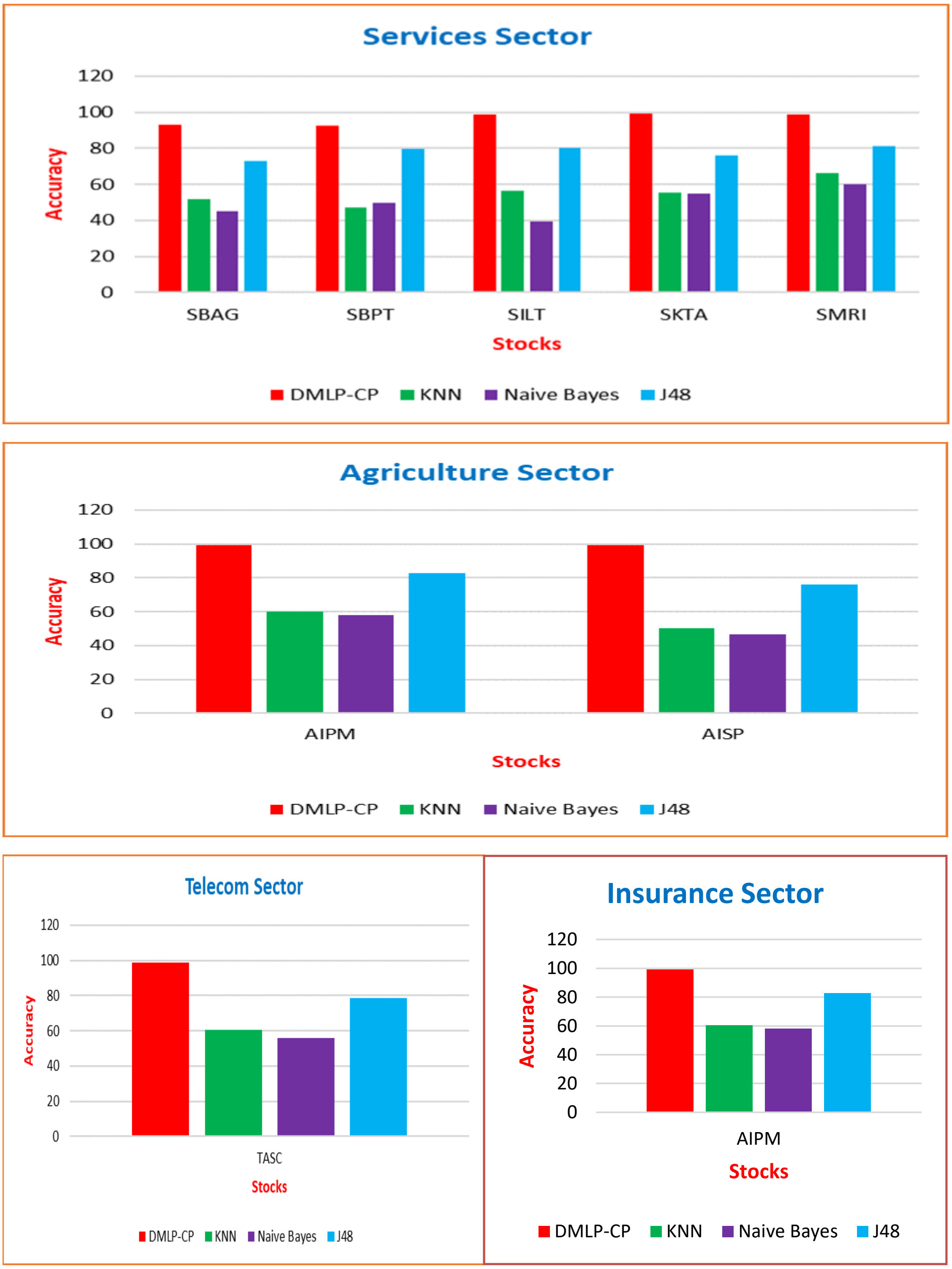


Figure 4.6. The bar chart accuracy of 43 stocks in ISX

4.5.2 The Results of NYSE

Initially, the DLC model is implemented using GN training method and unipolar sigmoid activation functions on 25 stocks in the NYSE dataset to find the best results in terms of Accuracy and execution time. After that the MLP using DeepLearning4J training by Gradient Decent (GD) method is also implemented on the same dataset for comparison. Table 4.10 shows the results of evaluation measures after implementing DLC and MLP using DeepLearning4J models on 25 stocks in the NYSE dataset.

Table 4.10. Evaluation criteria results for NYSE dataset

Stocks	Evaluation criteria	DLC	MLP using DeepLearning4J	Stocks	Evaluation criteria	DLC	MLP using DeepLearning4J
AAPL	**Accuracy**	89.473	53.401	**IBM**	**Accuracy**	99.132	50.85
	Exe. time	0.213	14.96		**Exe. time**	0.186	14.45
AIG	**Accuracy**	99.074	48.809	**JNJ**	**Accuracy**	84.268	50
	Exe. time	0.176	19.37		**Exe. time**	0.236	13.2
AMZN	**Accuracy**	89.647	51.19	**JPM**	**Accuracy**	99.074	53.57
	Exe. time	0.172	13.55		**Exe. time**	0.183	12.98
BA	**Accuracy**	98.901	52.04	**KO**	**Accuracy**	93.464	51.19
	Exe. time	0.184	14.53		**Exe. time**	0.197	14.78
CAT	**Accuracy**	94.33	50.51	**MSFT**	**Accuracy**	98.611	50.51
	Exe. time	0.157	14.12		**Exe. time**	0.179	12.8
COF	**Accuracy**	99.074	48.63	**NKE**	**Accuracy**	99.421	47.44
	Exe. time	0.144	13.73		**Exe. time**	0.158	13.68
EBAY	**Accuracy**	89.705	51.19	**ORCL**	**Accuracy**	98.380	50.17
	Exe. time	0.152	13.09		**Exe. time**	0.152	13.17
F	**Accuracy**	97.628	47.44	**PEP**	**Accuracy**	94.042	51.19
	Exe. time	0.19	13.45		**Exe. time**	0.138	14.5
FDX	**Accuracy**	99.248	49.48	**T**	**Accuracy**	98.496	50.68
	Exe. time	0.187	13.68		**Exe. time**	0.145	13.13
GE	**Accuracy**	97.860	52.21	**WMT**	**Accuracy**	98.669	50
	Exe. time		12.77		**Exe. time**	0.151	13.39
GM	**Accuracy**	97.628	47.44	**XOM**	**Accuracy**	98.958	51.87
	Exe. time	0.195	13.31		**Exe. time**	0.158	13.23
GOOG	**Accuracy**	98.95	50.85	**XRX**	**Accuracy**	97.108	50.17
	Exe. time	0.156	12.69		**Exe. time**	0.245	13.04
HD	**Accuracy**	84.904	51.87				
	Exe. time	0.159	14.18				

Table 4.10 shows the terrible superiority of the Accuracy results and the execution time of the DLC model. To analyze these results, Table 4.11 shows that the Accuracy average obtained for 25 stocks is (95.8418), while the highest accuracy is (99.421) for stock (FDX) and the lowest accuracy is (84.268) for stock (JNJ). For the execution time, the average execution time is (0.1783) second and the most time taken is (0.246) second for the stock (GE) and the least was for the stock (PEP).

Table 4.11. Analysis the results of Table 4.10

	Accuracy			**Execution time in second**		
	DLC model	**MLP using DeepLearning4J model**	**Accuracy Improvement rate**	**DCL model**	**MLP using DeepLearning4J model**	**improvement rate times**
Average	**95.8418**	50.508	90%	0.178	13.83	79.585
Max	**99.421**	53.57	110%	0.246	19.37	110.056
Min	**84.268**	47.44	64%	0.138	12.69	51.91

The percentage of the total Accuracy improvement rate is (90%) for 25 companies, while the highest improvement for the stock (NKE) with a percentage (110%) and the lowest improvement for the stock (HD) with (64%). However, in terms of execution time, the overall improvement rate is (79.585) times better, while the highest improvement is (110.056) times for a stock (AIG), and the least improvement is (51.91) times for a stock (GE).

Despite the satisfactory results of the DLC model, some companies had relatively poor predictive accuracy. For this reason, the TIs generation layer is added to improve the accuracy of these companies, as shown in Table 4.12.

Table 4.12. The DLC results with and without TIs generation layer for NYSE

Stocks	Evaluation criteria	DLC without TIs layer	DLC with TIs layer	Stocks	Evaluation criteria	DLC without TIs layer	DLC with TIs layer
AAPL	Accuracy	89.473	97.975	IBM	Accuracy	99.132	99.074
	Exe. time	0.213	0.631		Exe. time	0.186	0.53
AIG	Accuracy	99.074	99.016	JNJ	Accuracy	84.268	93.406
	Exe. time	0.176	0.443		Exe. time	0.236	0.629
AMZN	Accuracy	89.647	98.380	JPM	Accuracy	99.074	98.554
	Exe. time	0.172	0.576		Exe. time	0.183	0.425
BA	Accuracy	98.901	98.496	KO	Accuracy	93.464	97.686
	Exe. time	0.184	0.429		Exe. time	0.197	0.6
CAT	Accuracy	94.33	98.90	MSFT	Accuracy	98.611	98.264
	Exe. time	0.157	0.56		Exe. time	0.179	0.61
COF	Accuracy	99.074	98.496	NKE	Accuracy	99.421	98.958
	Exe. time	0.144	0.503		Exe. time	0.158	0.499
EBAY	Accuracy	89.705	98.38	ORCL	Accuracy	98.380	98.207
	Exe. time	0.152	0.574		Exe. time	0.152	0.569
F	Accuracy	97.628	97.686	PEP	Accuracy	94.042	98.149
	Exe. time	0.19	0.451		Exe. time	0.138	0.534
FDX	Accuracy	99.248	98.843	T	Accuracy	98.496	98.438
	Exe. time	0.187	0.401		Exe. time	0.145	0.625
GE	Accuracy	97.860	97.802	WMT	Accuracy	98.669	98.554
	Exe. time	0.246	0.547		Exe. time	0.151	0.558
GM	Accuracy	97.628	97.686	XOM	Accuracy	98.958	98.611
	Exe. time	0.195	0.45		Exe. time	0.158	0.465
GOOG	Accuracy	98.95	98.207	XRX	Accuracy	97.108	97.108
	Exe. time	0.156	0.472		Exe. time	0.245	0.536
HD	Accuracy	84.904	98.033				
	Exe. time	0.159	0.444				

According to Table 4.12, the accuracy results are improved by adding a TIs generation layer, especially for the stocks that are suffering from a deterioration in accuracy. The number of improved stocks is 10 and is marked in green, and the number of stocks in which their accuracy has decreased is 14 and is marked in blue while the number of stocks that have maintained their results is 1. The results of Table 4.12 are analyzed and shown in Table 4.13.

Table 4.13. Analysis the results of Table 4.12

	Accuracy			**Execution time in second**		
	DLC Without TIs layer	**DLC With TIs layer**	**Accuracy Improvement**	**DLC Without TIs layer**	**DLC With TIs layer**	**improvement rate times**
Average	95.841	**98.116**	2.274	0.178	0.522	-0.34408
Max	99.421	**99.074**	13.129	0.246	0.631	-0.48
Min	84.268	**93.406**	-0.743	0.138	0.401	-0.214

Table 4.13 illustrates that the accuracy average for all 25 stocks was (95.841) and became (98.116) after adding the TIs generation layer to the DLC model with an improvement rate of (2.274). As well the DLC model obtained the best accuracy of (99.074), while the lowest accuracy improved, which was (84.268) for a stock (JNJ) and became (93.406) with an improvement rate of (6.1192). Notwithstanding the good improvements that have been seen, but some stocks have decreased in accuracy, but that deterioration is not significant, as the rate of deterioration has reached (0.309) and the most significant deterioration was for stock (GOOG) by (-0.743). On the other hand, for the execution time, the time increased by adding the TIs generation layer by a small percentage amounting to (0. 34408) second to the total stocks, as shown in Table 4.13.

After presenting and analyzing the results in Table 4.13, the DLC model with the TIs generation layer proved its efficiency in terms of prediction accuracy but at the expense of execution time. As a result, the activation functions have been approximated to reduce the execution time with an accuracy very close to the original sigmoid activation function. Table 4.14 illustrates the accuracy and execution time results for the DLC model after replace unipolar sigmoid activation functions with a proposed approximation unipolar sigmoid activation functions based on Eq. 3.5.

Table 4.14. The DLC results with approximation activation function for NYSE

Stocks	Evaluation criteria	Unipolar activation function	Appx. unipolar activation function	Stocks	Evaluation criteria	Unipolar activation function	Appx. unipolar activation function
AAPL	Accuracy	97.975	97.802	IBM	Accuracy	99.074	99.132
	Exe. time	0.631	**0.218**		Exe. time	0.53	**0.183**
AIG	Accuracy	99.016	98.958	JNJ	Accuracy	93.406	93.58
	Exe. time	0.443	**0.15**		Exe. time	0.629	**0.25**
AMZN	Accuracy	98.380	98.554	JPM	Accuracy	98.554	98.785
	Exe. time	0.576	**0.25**		Exe. time	0.425	**0.151**
BA	Accuracy	98.496	98.322	KO	Accuracy	97.686	97.744
	Exe. time	0.429	**0.148**		Exe. time	0.6	**0.237**
CAT	Accuracy	98.90	98.785	MSFT	Accuracy	98.264	98.438
	Exe. time	0.56	**0.193**		Exe. time	0.61	**0.249**
COF	Accuracy	98.496	98.322	NKE	Accuracy	98.958	99.0167
	Exe. time	0.503	**0.146**		Exe. time	0.499	**0.202**
EBAY	Accuracy	98.38	98.207	ORCL	Accuracy	98.207	98.091
	Exe. time	0.574	**0.226**		Exe. time	0.569	**0.139**
F	Accuracy	97.686	97.686	PEP	Accuracy	98.149	98.554
	Exe. time	0.451	**0.161**		Exe. time	0.534	**0.203**
FDX	Accuracy	98.843	98.727	T	Accuracy	98.438	98.496
	Exe. time	0.401	**0.125**		Exe. time	0.625	**0.237**
GE	Accuracy	97.802	97.802	WMT	Accuracy	98.554	98.496
	Exe. time	0.547	**0.252**		Exe. time	0.558	**0.154**
GM	Accuracy	97.686	97.686	XOM	Accuracy	98.611	98.785
	Exe. time	0.45	**0.147**		Exe. time	0.465	**0.138**
GOOG	Accuracy	98.207	98.380	XRX	Accuracy	97.108	97.108
	Exe. time	0.472	**0.216**		Exe. time	0.536	**0.211**
HD	Accuracy	98.033	97.802				
	Exe. time	0.444	**0.206**				

According to Table 4.14, the execution time is significantly improved when using approximated activation function, while maintaining an accuracy very similar to that obtained from the original sigmoid activation function. Even though the satisfactory results, there is a JNJ stock whose accuracy result is (93.58) which is relatively less than its peers, so other activation functions such as (sigmoid bipolar, proposed approximation sigmoid bipolar, hyperbolic Tanh, and proposed approximation hyperbolic Tanh) are applied to improve the accuracy as shown in Table 4.15.

Table 4.15. The DLC results with six activation function for JNJ stock

JNJ	Unipolar	Approx. Unipolar	Bipolar	Approx. Bipolar	Tanh	Approx. Tanh
Accuracy	93.406	93.58	95.835	96.1249	97.397	**97.397**
Exe. time	0.629	0.25	0.62	0.485	1.294	**0.346**

According to the Table 4.15, the best accuracy (97.397) obtained when applying the function (approximation hyperbolic Tanh) in addition to the execution time also decreased by (0.283) second.

After displaying and analyzing the results in tables 4.10 to 4.15, the DLC model with the addition of a TIs generation layer and approximation activation function proved its efficiency in terms of accuracy and execution time. To further confirm the performance of the DLC model, it is compared with research study conducted by Yuan Song (2018) and NekoeiQachkanloo et al. (2019). The accuracy results of these models are summarized in Table 4.16. For the more Figure 4.7 visualize the accuracy bar chart for all twenty-five stocks in NYSE after applying three classification models.

Table 4.16. The DLC accuracy of the NYSE

No.	Stock	DLC	Song 2018	Nekoei 2019	No.	Stock	DLC	Song 2018	Nekoei 2019
1	AIG	**99.016**	58.79	91.59	14	IBM	**99.074**	-	92.46
2	AAPL	**97.975**	57.58	92.17	15	JNJ	**97.397**	56.52	93.04
3	AMZN	**98.38**	56.21	93.33	16	JPM	**98.554**	61.67	94.49
4	BA	**98.496**	60.15	85.50	17	KO	**97.686**	59.24	90.14
5	CAT	**98.9**	-	88.69	18	MSFT	**98.264**	59.09	89.27
6	COF	**98.496**	57.12	95.65	19	NKE	**98.958**	60.61	90.14
7	EBAY	**98.38**	59.70	89.56	20	ORCL	**98.207**	58.94	93.91
8	F	**97.686**	-	86.08	21	PEP	**98.149**	59.39	88.98
9	FDX	**98.843**	59.55	93.04	22	T	**98.438**	58.64	93.91
10	GE	**97.802**	61.21	92.17	23	WMT	**98.554**	60.91	94.49
11	GM	**97.686**	-	83.76	24	XOM	**98.611**	60.45	93.62
12	GOOG	**98.207**	58.33	89.70	25	XRX	**97.108**	-	85.50
13	HD	**98.033**	59.70	92.75					

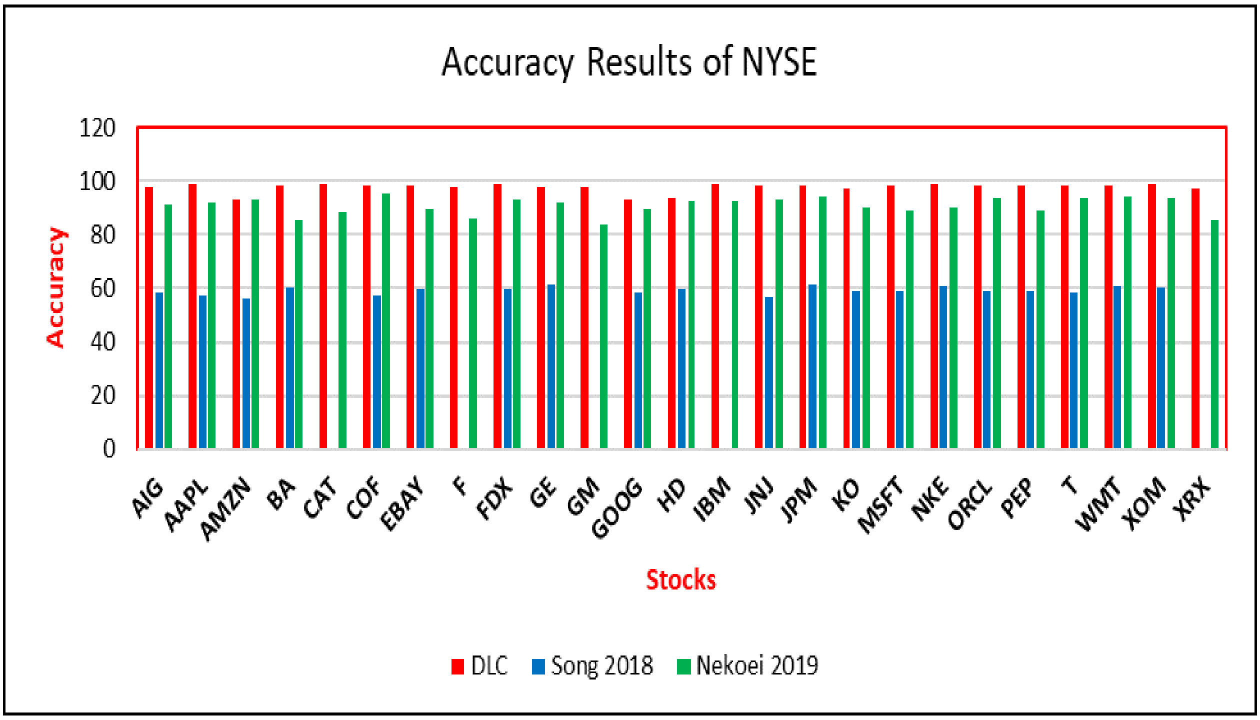


Figure 4.7. The bar chart accuracy of 25 stocks on the NYSE

4.6 Evaluating the Regression Model

Reliable evaluation of data mining techniques is performed on test data that had not been seen before during the training phase. In Cross-Validation (CV), each fold of the SM dataset is split into two parts in which the test data is 1/K from the original data and the rest is for training. In this dissertation, the value of K is equal to 10. Two measurements are used to evaluate the proposed DLR model which are RMSE and MAE.

4.6.1 The Results of ISX

Initially, the proposed DLR model is implemented (without and with) using the TIs generation layer on 43 stocks in the ISX dataset to find the best results in terms of MAE and RMSE. The evaluation results of the DLR model are shown in Table 4.17.

Table 4.17. The evaluation results of DLR model with and without TIs generation layer on ISX

Stocks	DLR without TIs layer			Stocks	DLR with TIs layer		
	RMSE	MAE	Exe. time		RMSE	MAE	Exe. time
BASH	0.0195	0.00950	0.0483	BASH	0.016997	0.0093	0.064
BBAY	0.01372	0.00725	0.051	BBAY	0.0099	0.00675	0.055
BBOB	0.04193	0.01442	0.062	BBOB	0.03949	0.01481	0.143
BCOI	0.01508	0.0065	0.0515	BCOI	0.01113	0.0061	0.1255
BDSI	0.2153	0.054	0.0705	BDSI	0.11640	0.0456	0.1
BGUC	0.01547	0.0058	0.073	BGUC	0.01473	0.0054	0.106
BIBI	0.01728	0.00763	0.041	BIBI	0.01379	0.0072	0.211
BIIB	0.01618	0.00714	0.0735	BIIB	0.01493	0.0067	0.0566
BIME	0.0135	0.0058	0.069	BIME	0.0110	0.00532	0.1045
BKUI	0.1013	0.0354	0.0875	BKUI	0.1017	0.0394	0.141
BMFI	0.01941	0.0075	0.0563	BMFI	0.0225	0.0075	0.138
BMNS	0.01279	0.0076	0.107	BMNS	0.0113	0.0074	0.0615
BNOI	0.0422	0.0137	0.0705	BNOI	0.0447	0.0155	0.219
BNOR	0.08057	0.0223	0.075	BNOR	0.0787	0.0270	0.124
BROI	0.0263	0.0110	0.0875	BROI	0.0279	0.01092	0.08
BSUC	0.01405	0.0047	0.103	BSUC	0.01532	0.0056	0.115
BUND	0.019	0.0066	0.082	BUND	0.01475	0.0067	0.1085
SBAG	0.3950	0.1372	0.04166	SBAG	0.33875	0.13857	0.07
SBPT	0.01767	0.00994	0.0685	SBPT	0.01507	0.00923	0.105
SILT	0.21415	0.0388	0.0575	SILT	0.2082	0.0398	0.089
SKTA	0.0768	0.04591	0.066	SKTA	0.0649	0.04225	0.149
SMRI	0.04954	0.02849	0.0715	SMRI	0.04482	0.02729	0.0605
TASC	0.32954	0.18427	0.0775	TASC	0.3404	0.1853	0.105
NAME	0.0788	0.0320	0.0935	NAME	0.07391	0.03062	0.16
AIPM	0.13459	0.0758	0.072	AIPM	0.1231	0.0713	0.125
AISP	0.28017	0.0886	0.0615	AISP	0.2261	0.0960	0.0855
IBSD	0.0599	0.0283	0.055	IBSD	0.0573	0.0264	0.065
IHLI	0.0289	0.00912	0.095	IHLI	0.0283	0.0092	0.1145
IIDP	0.03442	0.0196	0.053	IIDP	0.0273	0.0180	0.143
IITC	0.2112	0.07947	0.046	IITC	0.2053	0.07912	0.1445
IKLV	0.0293	0.0196	0.053	IKLV	0.0268	0.01849	0.11
IMAP	0.0176	0.0099	0.07	IMAP	0.01507	0.00923	0.1065
IMIB	0.04792	0.031	0.057	IMIB	0.0425	0.0289	0.0625
IMOS	0.12649	0.0820	0.062	IMOS	0.12230	0.07384	0.0975
INCP	0.0249	0.0133	0.0443	INCP	0.0228	0.0125	0.0635
IRMC	0.44308	0.2212	0.0656	IRMC	0.4009	0.198	0.1025
HBAG	0.2693	0.101	0.068	HBAG	0.19346	0.09197	0.099
HBAY	1.5950	0.86954	0.0705	HBAY	1.4733	0.8597	0.102
HISH	1.48811	0.4538	0.085	HISH	1.5487	0.5139	0.071
HKAR	0.1055	0.040912	0.0516	HKAR	0.1345	0.0437	0.097
HMAN	1.1813	0.4019	0.0725	HMAN	1.11615	0.441	0.053
HNTI	0.5831	0.18914	0.089	HNTI	0.3274	0.16025	0.121
HPAL	0.3295	0.18427	0.08	HPAL	0.3404	0.1853	0.1

Table 4.17 shows that the error rate based on (RMSE and MAE) of the proposed DLR model is very satisfactory. After analysing the results RMSE shows that 34 stocks have been improved out of 43 stocks by adding the TIs generation layer, For MAE, 27 stocks have been improved. Table 4.18 shows the statistical analysis of these results.

Table 4.18. Analysis the results of Table 4.17

	RMSE			**MAE**			**Execution time in second**		
	Without TIs layer	**With TIs layer**	**Change rate**	**Without TIs layer**	**With TIs layer**	**Change rate**	**Without TIs layer**	**With TIs layer**	**Change rate**
Average	0.20547	**0.1879**	0.01749	0.084	0.084	0	0.0682	0.1059	0.03763
Worst	1.595	**1.5487**	0.06059	0.869	0.859	0.0601	0.107	0.219	0.17
Best	0.01279	**0.0099**	-0.2557	0.004	0.005	-0.02889	0.041	0.053	-0.0455

From Table 4.18, it was noted that the percentage of improvement is relatively small with an increase in the execution time. The DLR model had a relatively small error rate before adding the layer and does not need an additional improvement. The logical reason for that is the closing prices for stocks in the ISX have very low values ranging between 0.3 and 30 Iraqi dinars and the percentage of change was few after analyzing all historical data. As a result, the proposed DLR model will be satisfied without adding the TIs generation layer when predicting stock prices for the ISX.

4.6.2 The Results of NYSE

Initially, the proposed DLR model is implemented on 25 stocks in the NYSE dataset to find the best results in terms of MAE and RMSE. Table 4.19 summarizes the outcomes of MAE and RMSE for the proposed DLR model. It illustrates the outcomes of estimating stocks prices for stocks before and after the use of the TIs generation layer. The findings demonstrate the essential importance of this layer in enhancing the model's forecasting performance.

Table 4.19. The evaluation results of DLR model with and without TIs generation layer on NYSE

Stocks	DLR without TIs layer			Stocks	DLR with TIs layer		
	RMSE	MAE	Exe. time		RMSE	MAE	Exe. time
AAPL	3.0287	1.9795	0.083	**AAPL**	**2.7222**	**1.6730**	0.186
AIG	0.338	0.2516	0.111	**AIG**	**0.2047**	**0.1588**	0.379
AMZN	2.9533	2.0542	0.1335	**AMZN**	**2.5653**	**1.659**	0.317
BA	0.6515	0.4872	0.0555	**BA**	**0.3805**	**0.2926**	0.168
CAT	0.9936	0.4314	0.0785	**CAT**	**0.4624**	**0.1423**	0.0825
COF	0.4158	0.3161	0.179	**COF**	**0.2215**	**0.17069**	0.1335
EBAY	0.3085	0.2248	0.079	**EBAY**	**0.18775**	**0.14043**	0.078
F	0.095	0.0711	0.0516	**F**	**0.03757**	**0.02814**	0.1845
FDX	0.8364	0.57708	0.079	**FDX**	**0.4265**	**0.289**	0.156
GE	0.1469	0.1014	0.067	**GE**	**0.0699**	**0.0526**	0.128
GM	0.0952	0.0712	0.0503	**GM**	**0.0398**	**0.0283**	0.151
GOOG	3.9398	2.942	0.17	**GOOG**	**2.3669**	**1.4419**	0.139
HD	0.7122	0.4939	0.24	**HD**	**0.3866**	**0.2921**	0.423
IBM	0.7825	0.595	0.1925	**IBM**	**0.3806**	**0.2906**	0.233
JNJ	0.3981	0.2905	0.111	**JNJ**	**0.2483**	**0.1863**	0.352
JPM	0.3363	0.2367	0.133	**JPM**	**0.1608**	**0.1048**	0.325
KO	0.2542	0.17917	0.0975	**KO**	**0.2124**	**0.1513**	0.141
MSFT	0.2261	0.1665	0.322	**MSFT**	**0.11426**	**0.0788**	0.0965
NKE	0.5295	0.3816	0.206	**NKE**	**0.3265**	**0.234**	0.262
ORCL	0.1937	0.1481	0.0815	**ORCL**	**0.0886**	**0.06591**	0.404
PEP	0.4392	0.2701	0.214	**PEP**	**0.2307**	**0.1759**	0.123
T	0.14808	0.1124	0.285	**T**	**0.1386**	**0.1064**	0.3
WMT	0.3106	0.229	0.219	**WMT**	**0.1634**	**0.1192**	0.18
XOM	0.4128	0.3078	0.167	**XOM**	**0.1683**	**0.11723**	0.287
XRX	0.0717	0.0524	0.214	**XRX**	**0.0328**	**0.0244**	0.186

According to Table 4.19, the prediction results of the two criteria (RMSE, MAE) are improved by adding TIs generation layer for all 25 stocks. The results of Table 4.19 are analyzed and shown in Table 4.20 to give a summary of them.

Table 4.20. Analysis the results of Table 4.19

	RMSE			MAE			Execution time in second		
	Without TIs layer	With TIs layer	Change rate	Without TIs layer	With TIs layer	Change rate	Without TIs layer	With TIs layer	Change rate
Average	0.7447	0.4934	-0.2512	0.51883	0.3209	-0.1978	0.1447	0.2166	0.0718
Worst	3.9398	2.7222	-0.0094	2.942	1.673	-1.5001	0.322	0.423	0.3225
Best	0.0717	0.0328	-1.5729	0.0524	0.0244	-0.006	0.0503	0.078	-0.2255

Table 4.20 illustrates that the average results of RMSE for all 25 stocks was (0.7447) and became (0.4934) after adding the TIs generation layer to the DLR model with an improvement rate of (0.2512). The DLR model obtained the best RMSE of (0.0328), while the worst RMSE is (2.7222). However, in term of MAE, the average results for all 25 stocks was (0.51883) and became (0.3209) after adding the TIs generation layer to the DLR model with an improvement rate of (0.1978). The DLR model obtained the best MAE of (0.0244), while the worst RMSE is (1.673). On the other hand, for the execution time, the time increased by adding the TIs generation layer by a small percentage (0.0718) second to the total stocks.

After presenting and analyzing the results in Table 4.20, the DLR model with the addition of the TIs generation layer proved its efficiency in terms of prediction error rate but at the expense of execution time. As a result, the activation functions have been approximated to reduce the execution time with an error rate very close to the original sigmoid activation function as explained in the previous section. Even though the acceptable results, there are three stocks (APPL, AMZN and GOOG) which have a relatively high error rate, thus other activation functions such as sigmoid bipolar, proposed approximation sigmoid bipolar, hyperbolic Tanh, and proposed approximation hyperbolic Tanh are applied to improve the accuracy as shown in Table 4.21.

Table 4.21. The DLR results with six activation function for three stocks in NYSE

	APPL			**AMZN**			**GOOG**		
	RMSE	**MAE**	**Time**	**RMSE**	**MAE**	**Time**	**RMSE**	**MAE**	**Time**
Unipolar	2.722	1.673	0.113	2.565	1.659	0.151	2.366	1.441	0.112
App. Unipolar	2.342	1.62	0.143	2.336	1.612	0.239	2.509	1.561	0.211
Bipolar	2.433	1.160	0.115	2.006	1.385	0.084	**1.3**	**0.767**	0.25
App. Bipolar	1.541	1.102	0.087	2.074	1.539	0.144	1.712	0.991	0.141
Tanh	**1.504**	**1.076**	0.124	**1.9476**	**1.232**	0.0845	1.477	0.855	0.145
App. Tanh	1.837	1.280	0.082	2.0468	1.361	0.064	1.47	0.874	0.106

In Table 4.21, it was noticed that the activation functions affected the error rate, as the error rate of Apple and Amazon stocks decreased by applying the hyperbolic Tanh and it achieved the best results, while the Google stock achieved the best results by applying the bipolar activation function.

To confirm the performance of the DLR model, it is compared with a research study conducted by NekoeiQachkanloo et al. (2019). The RMSE and MAE results of these models are summarized in Table 4.22. Figures 4.8 and 4.9 visualize the MAE and RMSE bar chart respectively for all twenty-five stocks in NYSE after applying the prediction models.

Table 4.22. The Comparison results of RMSE and MAE for NYSE

Stocks	NekoeiQachkanloo 2019		Proposed DLR		Stocks	NekoeiQachkanloo 2019		Proposed DLR	
	MAE	RMSE	MAE	RMSE		MAE	RMSE	MAE	RMSE
AAPL	**1.058**	**1.483**	1.6730	2.7222	**IBM**	1.178	1.681	**0.2906**	**0.3806**
AIG	0.41	0.71	**0.1588**	**0.2047**	**JNJ**	0.613	0.889	**0.1863**	**0.2483**
AMZN	7.973	11.085	**1.659**	**2.5653**	**JPM**	0.598	0.879	**0.1048**	**0.1608**
BA	1.31	1.748	**0.2926**	**0.3805**	**KO**	0.236	0.333	**0.1513**	**0.2124**
CAT	0.848	1.15	**0.1423**	**0.4624**	**MSFT**	0.461	0.708	**0.0788**	**0.11426**
COF	0.757	1.018	**0.17069**	**0.2215**	**NKE**	0.598	0.834	**0.234**	**0.3265**
EBAY	0.291	0.474	**0.14043**	**0.18775**	**ORCL**	0.286	0.408	**0.06591**	**0.0886**
F	0.131	0.19	**0.02814**	**0.03757**	**PEP**	0.579	0.802	**0.1759**	**0.2307**
FDX	1.517	2.227	**0.289**	**0.4265**	**T**	0.229	0.313	**0.1064**	**0.1386**
GE	0.228	0.32	**0.0526**	**0.0699**	**WMT**	0.502	0.799	**0.1192**	**0.1634**
GM	0.332	0.455	**0.0283**	**0.0398**	**XOM**	0.669	0.885	**0.11723**	**0.1683**
GOOG	6.676	9.707	**1.4419**	**2.3669**	**XRX**	0.11	0.16	**0.0244**	**0.0328**
HD	1.086	1.478	**0.2921**	**0.3866**					

The proposed DLR model outperformed the prediction accuracy of earlier research based on the MAE and RMSE measures in Twenty-four stocks. The only exception was for AAPL stock. Based on Table 4.22, the error rate of DLR model amounted to (0.32094 and 0.49347), while it are equal to (1.14704 and 1.62944) for the model of (NekoeiQachkanloo et al., 2019) according to the two measures MAE and RMSE, respectively. Finally, the research concludes that (XRX) company has the best predictive error rate, as its values were (0.0244) and (0. 0328) for the evaluation measures MAE and RMSE, respectively for the proposed DLR model. While the results of the previous model (NekoeiQachkanloo et al., 2019) are (0.110) and (0.160) for the same evaluation measures. As for APPL company, it had the worst forecast error rate, as its values are (1.673) and (2.7222) for the evaluation measures MAE and RMSE, respectively for the proposed DLR model. Moreover, the worst expected error rate for AMZN results obtained from the previous model (NekoeiQachkanloo et al., 2019) is (7.973) and (11.085) for the same evaluation measures.

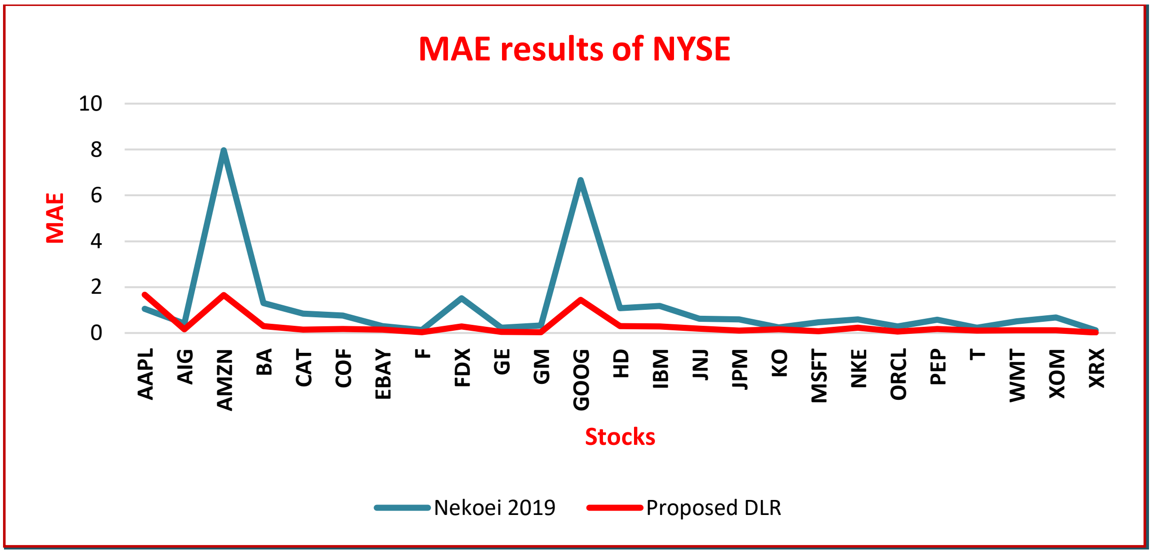


Figure 4.8. The bar chart MAE of 25 stocks on the NYSE

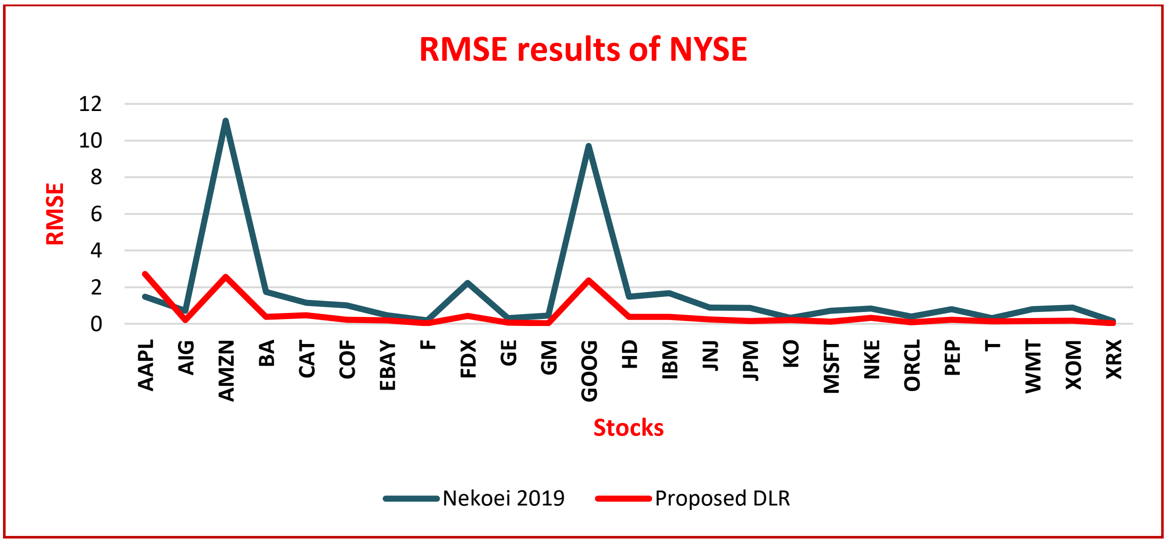


Figure 4.9. The bar chart MAE of 25 stocks on the NYSE

4.7 Evaluating The Portfolio Optimization Model

The active portfolio's success cannot be judged just based on its outcomes. A peer reference is required in which the most essential forms are market indexes. Indices have been used in financial markets for almost a century. They differ in terms of their building method and attributes. Some of them reflect the whole market, while others represent certain company shares or investment philosophies. An active portfolio manager or active investor can pick what he/she considers appropriate based on the nature of his/her portfolio's structure.

The Sharpe ratio, a risk-return evaluation metric, is utilized in this dissertation to provide a unified measure based on a risk rate. In certain circumstances, it is also known as the reward-volatility measure since its composition is based on calculating the excess return compared to the total risk unit.

4.7.1 The Results of ISX

Three models are used to construct an active portfolio for the ISX namely, MVO, EW, and the proposed DPO model in this dissertation. The Sharpe ratio is determined for the three models as well as the general market index for comparison. Table 4.23 shows the preferences and weights of each portfolio generated using the three models, whereas the results of this analysis are displayed in Table 4.24. Figure 4.10 shows the weights of stocks within the portfolio for the three models.

Table 4.22 shows the stock weights of the ISX dataset after applying the three models, it can be deduced that the MVO and the proposed model used the same seven companies (AISP, IITC, IMAP, IMOS, INCP, IRMC, and HBAY), but with different weights to construct their portfolios. The EW model, on the other hand, picked all of the equal assets with equal weight.

Table 4.23. The weights of the assets in portfolio for ISX

Stocks	MVO	EW	DPO	Return	risk	Sharp	Stocks	MOV	EW	DPO	Return	Risk	Sharp
BASH	0	0.023	0	-0.34	0.511	-0.668	SKTA	0	0.023	0	-0.15	0.36	-0.409
BBAY	0	0.023	0	-0.52	0.594	-0.883	SMRI	0	0.023	0	-0.19	0.364	-0.515
BBOB	0	0.023	0	-0.47	0.467	-1.017	AIPM	0	0.023	0	-0.14	0.548	-0.258
BCOI	0	0.023	0	-0.09	0.444	-0.206	AISP	**0.035**	0.023	**0.0459**	0.03	0.531	0.061
BDSI	0	0.023	0	-0.93	0.681	-1.36	IBSD	0	0.023	0	0.01	0.362	0.019
BGUC	0	0.023	0	-0.48	0.452	-1.068	IHLI	0	0.023	0	-0.02	0.486	-0.049
BIBI	0	0.023	0	-0.37	0.4	-0.933	IIDP	0	0.023	0	0.01	0.469	0.018
BIIB	0	0.023	0	-0.32	0.412	-0.784	IITC	**0.332**	0.023	**0.3335**	0.18	0.339	0.532
BIME	0	0.023	0	-0.52	0.555	-0.939	IKLV	0	0.023	0	-0.2	0.487	-0.413
BKUI	0	0.023	0	-0.1	0.562	-0.181	IMAP	**0.106**	0.023	**0.126**	0.11	0.404	0.268
BMFI	0	0.023	0	-0.46	0.698	-0.657	IMIB	0	0.023	0	-0.01	0.608	-0.015
BMNS	0	0.023	0	-0.17	0.36	-0.463	IMOS	**0.17**	0.023	**0.156**	0.27	0.598	0.46
BNOI	0	0.023	0	-0.13	0.567	-0.236	INCP	**0.174**	0.023	**0.159**	0.24	0.515	0.462
BNOR	0	0.023	0	-0.69	0.779	-0.881	IRMC	**0.095**	0.023	**0.088**	0.24	0.725	0.333
BROI	0	0.023	0	-0.33	0.468	-0.697	HBAG	0	0.023	0	-0.03	0.356	-0.085
BSUC	0	0.023	0	-0.13	0.304	-0.418	HBAY	**0.087**	0.023	**0.0902**	0.15	0.517	0.294
BUND	0	0.023	0	-0.67	0.746	-0.902	HISH	0	0.023	0	-0.24	0.624	-0.391
TASC	0	0.023	0	-0.2	0.463	-0.433	HKAR	0	0.023	0	-0.52	0.523	-0.99
NAME	0	0.023	0	-0.14	0.821	-0.167	HMAN	0	0.023	0	-0.34	0.469	-0.714
SBAG	0	0.023	0	-0.92	0.66	-1.391	HNTI	0	0.023	0	-0.21	0.458	-0.45
SBPT	0	0.023	0	-0.3	0.665	-0.454	HPAL	0	0.023	0	-0.14	0.416	-0.327
SILT	0	0.023	0	-0.32	0.489	-0.651							

Table 4.24. Sharp Ratio Results for ISX

	return	risk	Sharp Ratio
General Index	0.200409	14.69128	0.013641
Proposed DPO	0.191152	0.204522	**0.944627**
MVO	0.19693	0.211225	0.932326
EW	-0.22198	0.08241	-2.693

According to the results presented in Table 4.24, the portfolio formed by the proposed DPO model has the best Sharpe ratio in comparison to other models and the general market index. The DPO model selects seven assets out of forty-three to form the portfolio. Table 4.23 shows that these companies are profitable and have the lowest risk rate in comparison to other companies. On the other hand, the observed results indicate that the EW model produced the worst outcomes. The rationale reason for this is that out of a total of 43 companies, 36 are losing and these had a substantial influence on the portfolio's performance. As a result, the contribution of these companies to the formation of the general market index is also a factor in their poor performance.

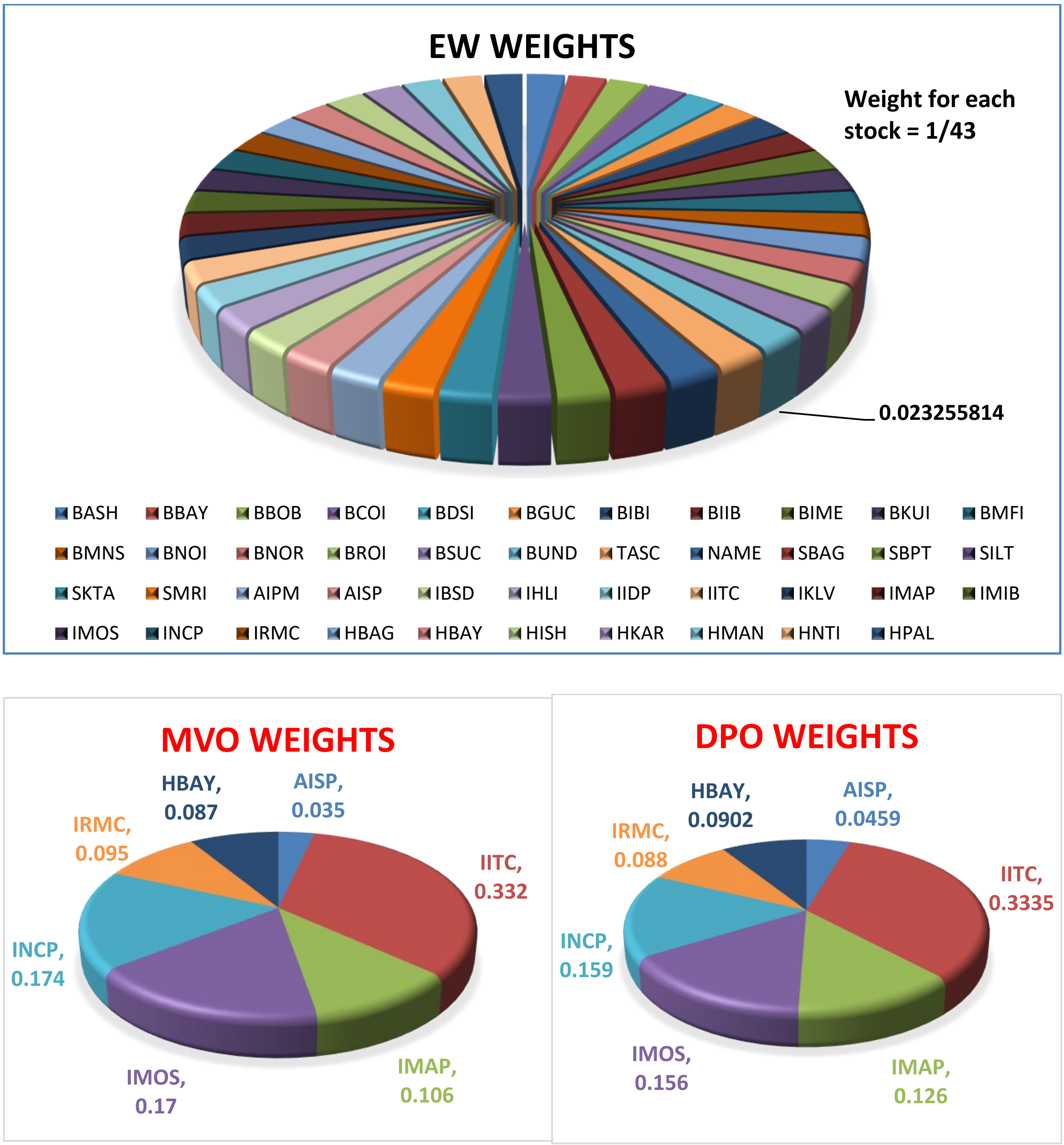


Figure 4.10. The weights of the assets in a portfolio for ISX

4.7.2 The Results of NYSE

To obtain an optimal allocation of assets, three models are implemented namely, MVO, EQ, and the proposed DPO model. This is to highlight the strength of DPO model as shown in Table 4.25. Based on the results presented in Table 4.25, it is clear that DPO model outperformed the other models. The PDO model produced the largest Sharpe ratio, maximized profits, and reduced risks in the portfolio. In another research (NekoeiQachkanloo et al., 2019), the fuzzy logic model was used to build the portfolio in which it was close to but not better than MVO, whereas DPO result is better than MVO. Figure 4.11 shows the weights of stocks within the portfolio for the three models.

Table 4.25. Weights for stocks after applying three models for NYSE

Stocks	MVO	DPO	EW	Expected Return	Volatility
AAPL	0	0	0.04	0.000399	0.02624
AIG	0	0	0.04	0.000677	0.021613
AMZN	**0.20301**	**0.197282**	0.04	0.001186	0.020395
BA	0	**0.031196**	0.04	0.000694	0.015165
CAT	**0.13476**	**0.168609**	0.04	0.000787	0.02357
COF	0	0	0.04	0.000621	0.018091
EBAY	0	0	0.04	0.000452	0.023107
F	0	0	0.04	0.000261	0.018235
FDX	0	0	0.04	0.000576	0.015513
GE	0	0	0.04	0.0005	0.013725
GM	0	**0.042849**	0.04	0.000116	0.017901
GOOG	0	0	0.04	0.000354	0.019843
HD	**0.66223**	**0.521644**	0.04	0.000959	0.012856
IBM	0	0	0.04	0.000201	0.012075
JNJ	0	**0.002754**	0.04	0.000367	0.008889
JPM	0	0	0.04	0.000544	0.017126
KO	0	0	0.04	-2.52E-05	0.015289
MSFT	0	0	0.04	0.000502	0.014607
NKE	0	0	0.04	0.000199	0.022715
ORCL	0	0	0.04	0.000364	0.015195
PEP	0	**0.035666**	0.04	0.000344	0.008884
T	0	0	0.04	0.000274	0.009831
WMT	0	0	0.04	0.000191	0.01031
XOM	0	0	0.04	0.000223	0.011993
XRX	0	0	0.04	0.000182	0.01864
Sharpe Ratio	**1.24**	**1.406013**	**0.747867**		

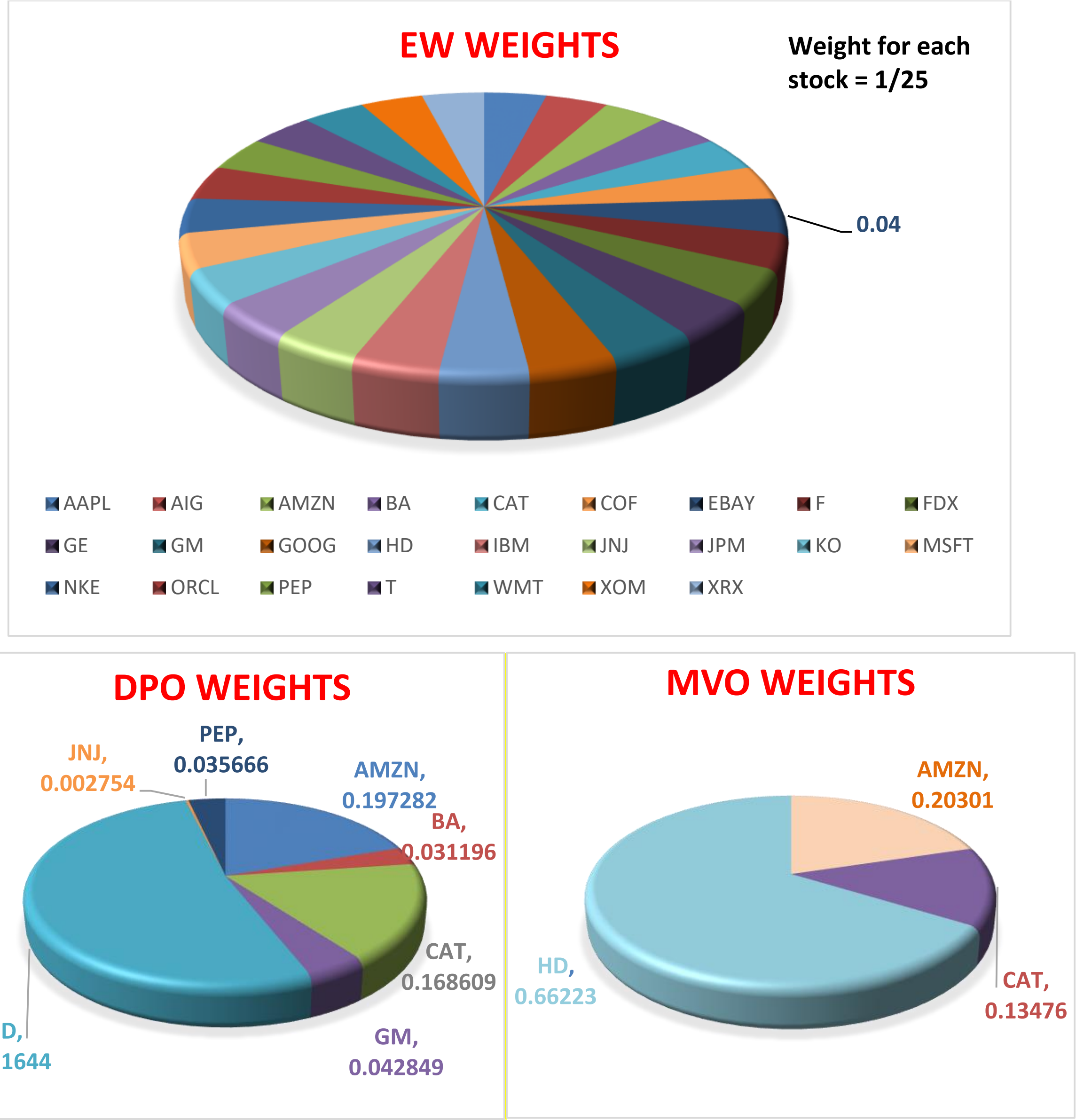


Figure 4.11. The weights of the assets in a portfolio for NYSE

Table 4.25 illustrates that the DPO model selected seven stocks, whereas MVO chose three stocks only. As it is clear in Table 4.25, the MVO model chose three stocks (HD, AMZN, and CAT) out of Twenty-five stocks with weights (0.66223, 0.20301, and 0.13476) respectively, to form the portfolio with a Sharpe ratio of (1.24). The DPO model selects seven stocks (HD, AMZN, CAT, GM, PEP, BA, and JNJ) with weights (0.521644168, 0.19728178, 0.168609017, 0.042849363, 0.035665882, 0.031195716, and 0.002754074) respectively, to form the portfolio with a Sharpe ratio of (1.406).

4.8 Discussion and Justification

The main purpose of the proposed system is to improve the accuracy and execution time of SM prediction, furthermore it aimed to find the best portfolio that can provide the best returns with the least risks. Several experiments are performed on the global stock exchange dataset represented by NYSE and the local stock exchange dataset represented by ISX to prove the efficiency of the proposed system.

4.8.1 Training Model

As shown in Tables 4.3, 4.4 and 4.10, the accuracy results seem to be significantly affected by the training model. The QN optimization method, which is based on the second derivative, is used to train DLC and DLR models. It had a very powerful effect in obtaining very high accuracy with a very fast convergence time. The rationale from these high accuracy results is that the GN method avoids getting trapped in local minima. Moreover, the learning rate parameters can be adapted automatically. As a result, this method reduced the generalization error and made the DNN fully trained to provide the best weights.

4.8.2 Technical Indicators

Employment of technical indicators represented by the TIs generation layer has an effective impact on the performance of the proposed system. These effects are summarized in the following points:

1- At the general level, and according to the results shown in tables (4.6, 4.12, 4.17 and 4.19), the performance of the DLC and DLR models are improved after applying the TIs generation layer and for both ISX and NYSE markets.

2- For ISX dataset, TIs generation layer has a strong impact on improving the accuracy of prediction, especially with the DLC model, while they were not as strong when applying the DLR model. The reason for the strong effect on the DLC model is because technical indicators help guide the classifier to predict stock movement by generating features related with class attribute, which leads to improving the accuracy of the decision. Whereas the reason for weak impact on performance of DLR is for two issues, the first relates to the fact that the price of stocks is low, and the second relates to the fact that the change in the market is very little, and this leads to the result that the general error rate was relatively low, so the effect of this layer on performance did not appear.

3- For NYSE dataset, TIs generation layer has a strong impact on improving the accuracy of prediction for both the DLC and DLR models.

4- At the company level, the TIs generation layer strongly improves the results of disturbed companies that have little accuracy but it has a fluctuating effect with stable companies that have high accuracy.

4.8.3 Activation Function

Activation functions also had an effective role in improving the accuracy of prediction as shown in tables (4.15 and 4.21). The reason for this improvement is because the outputs from bipolar sigmoid and hyperbolic Tanh functions are within the range between [-1,1] while the output of unipolar sigmoid function is in range [0, 1]. Our conclusion is that to fit the inputs to the function, which fall within the range [-1, 1] after applying the standardization layer with the outputs that fall within the same range. In addition, the proposed approximate activation functions play a role in improving the execution time while maintaining an accuracy very close to the original.

4.8.4 The Prediction Results

Due to the absence of previous studies applied to the ISX, the research resorted to comparing the proposed DLC model with three standard classifiers to prove the performance efficiency of the model. According to Table 4.9, the results show that the model DLC outperforms the three classifiers of forty-three companies with a very large percentage, as the accuracy rate is 97.942, 83.36, 62.35, and 56.44 for the classifiers DLC, DT, KNN, and Naïve Bayes, respectively. The best accuracy of the models was 99.482, 93.313, 89.6657, and 80.8511, while the worst accuracy was 92.657, 47.416, 46.8085, and 27.9635. furthermore, the DLR model also proved its efficiency in predicting the prices of stocks with a very small error rate, as the results showed in the tables 4.17 and 4.18.

As for America's dataset, the goal to apply proposed DLC and DLR models on NYSE for two reasons, the first to prove the validity of our construction of Iraq's dataset, and the second to prove the efficiency of the proposed model by applying it to previously assessed data by previous studies. According to Table 4.16, which presents the accuracy results of DLC with two previous studies (Yuan Song, 2018) and (NekoeiQachkanloo et al., 2019) on the same data with the same period for comparison. The results show that the DLC model outperforms the two classifiers of twenty-five companies, where the accuracy rate is 98.276, 90.9576, and 59.19 for the models DLC, (NekoeiQachkanloo et al., 2019), and (Yuan Song, 2018), respectively. The best accuracy of the models was 99.074, 95.65, and 61.67, while the worst accuracy was 97.108, 83.76, and 56.21. On the other hand, Table 4.22 shows the results of the DLR model, after applying it to the data of NYSE, the results also show the superiority of the proposed model in terms of the RMSE and MAE over the results of (NekoeiQachkanloo et al., 2019) in Twenty-four stocks. The only exception was for AAPL stock. The rationale of DLR results in comparison with (NekoeiQachkanloo et al., 2019) is that the results of all companies that had similar behavior are in the same range. Therefore, the error rate of these stocks is more than one according to the evaluation criteria. This is because the price is relatively stable and not high where it did not exceed 150 dollars. Unlike the other three stocks AAPL, AMZN, and GOOG, the price reached some periods to 1200 dollars. In (NekoeiQachkanloo et al., 2019), however, there is a clear inconsistency in the findings of AAPL, AMZN, and GOOG stocks, although they had similar behavior.

Four possible reasons are behind the high prediction accuracy obtained in this dissertation. The first reason is to choose the appropriate topology for the neural network with adjusting the number of layers, the number of nodes within each layer, and other parameters. The second reason is the extraction of new TIs from the original features. The experimental analysis shows that these new extracted attributes were more relative to the target feature (close price). The third reason is to use the QN optimization method instead of GD to train the network (layer three). The fourth reason is the application of other activation functions that worked better than the traditional activation function.

4.8.5 Portfolio Optimization Results

According to Table 4.23 which shows the stock weights of the ISX dataset after applying the three models, it can be deduced that the MVO and the proposed model used the same seven companies (AISP, IITC, IMAP, IMOS, INCP, IRMC, and HBAY), but with different weights to construct their portfolios. According to the results is presented in Table 4.24, the portfolio consisting of the proposed DPO model has the best Sharpe ratio in comparison to other models and the general market index. The results also show that there is little improvement in the percentage of Sharpe Ratio, and the reason for this is that 37 out of 43 companies are losers.

After presenting and analyzing the results of ISX data, the portfolio consisted of seven companies, five of which belonged to the industry sector, one to the hotel sector, and the other to the agricultural sector. As a result, the DPO model recommends nominating the industry sector to invest in it. Likewise, after allocating the weights, it was found that (IITC) company was the best company that the DPO model could recommend investing in.

The same is true for the American Stock Exchange. Table 4.25 illustrates that the proposed DPO model is selected seven stocks, whereas MVO choses three stocks only. According to the results is presented in Table 4.25, the portfolio consisting of the proposed DPO model has the best Sharpe ratio in comparison to other models. The possible explanation for obtaining better results is the use of the prediction return from phase one in the optimization phase. The other reason is the exploitation of GRG optimization in training the weights of NN that were also allocated for stocks in the portfolio. As it is clear in Table 4.25, the MVO model chose three stocks (HD, AMZN, and CAT) out of Twenty-five stocks with weights (0.66223, 0.20301, and 0.13476) respectively, to form the portfolio with a Sharpe ratio of (1.24). While our proposed model selects seven stocks (HD, AMZN, CAT, GM, PEP, BA, and JNJ) with weights (0.521644168, 0.19728178, 0.168609017, 0.042849363, 0.035665882, 0.031195716, and 0.002754074) respectively, to form the portfolio with a Sharpe ratio of (1.406).

4.8.6 Implications of Dissertation

This research has many theoretical and practical implications. From the theoretical side, this dissertation enriches the theoretical base of previous research on portfolio optimization with return prediction. It developed an integrated Deep Neural Network system to provide clear guidance for fund managers and individual investors. The proposed system, therefore, can be used in the selection of the right strategies for making investment decisions, and this, in turn, can enhance revenue during uncertainty periods. The key phases of the proposed models can be applied with any other stocks to improve prediction accuracy and portfolio optimization.

In addition to the good results obtained, there are several advantages of the proposed system related to the practical aspect. first, employing the multithread principle in most layers of the DNN had a role in reducing the time for the TIs generation layer by 5 to 1 through the operation of technical indicators simultaneously at the same time instead of applying them sequentially. In the second layer (Standardization layer), the percentage of reduction in time was 10 to 1. As for the network as a whole, the application of this principle had an effective role in reducing the time taken from n to 1. Second, employing second derivative method (GN) in the prediction layer had the merit of reducing the convergence time for network training from 45 minutes to 0.75 seconds. Third, the suggestion of an approximate activation function based on the principle of the Toiler series also led to a reduction in the execution time because the original activation functions use the exponential function in their equation, and the calculation of this function is expensive. Finally, although Markowitz's model plays an important role in high profitability, it suffers from some problems including execution time. Since to apply it, the quadratic programming problem of the trade-offs between risk and return must be solved to obtain an efficient frontier. This dissertation tried as much as possible to reduce the execution time by applying the GRG method to improve the portfolio, which in turn has a solution close to the optimal solution, unlike Markowitz's method, which searches for the exact solution and requires a long time to achieve it.

Chapter Five

The Conclusion and Future Works

Conclusions and Future Works

5.1 Conclusions

The main purpose of the proposed system is to improve the accuracy and execution time of SM prediction, in addition to finding the best portfolio that can give the best returns with the least risks. It, therefore, extends previous literature on DNN-based price prediction and portfolio optimization models. The proposed topology used DNN in two phases, in which the expected return prediction process is achieved in the first phase, whereas the second represented the portfolio optimization process. The first layer of phase one is used to increase the accuracy of forecasting daily earnings. the features values in the second layer are standardized within a certain range, so that the results do not deviate from one feature to another. The third layer forecasted daily profit returns. The second phase which included one layer only assigned weights to each item in the portfolio to maximize profitability. Using the ISX and NYSE dataset, the suggested model achieved high prediction accuracy as measured by the MAE, RMSE and Accuracy metrics. It also outperformed the EW and MVO models, resulting in better earnings and fewer risks in the overall portfolio.

The most important conclusions of this dissertation that were discovered through the design and implementation of the proposed system are:

1. The proposed multi-layer, multi-purpose (DNN) system has been shown to be effective in identifying and creating the best features that have a positive effect on the accuracy of the buy, sell or hold stock decision by applying the DLC model. In addition, it gives promising results in price prediction and decision-making by employing the DLR model. Finally, the proposed system has proven its efficiency by assigning weights to the stocks within the portfolio to give the best profits in relation to risk by using the DPO model.

2. For the first time, a benchmark for the ISX is established and organized. Building a database of eighty-nine Iraqi enterprises in eight different sectors — banks, communications, investment, insurance, services, agriculture, industry, and hotels — will be necessary to achieve this goal successfully.
3. In general, the DLC model shows the better results for all datasets according to Accuracy evaluation measure. The DLR model gives better results for all datasets according to RMSE and MAE evaluation measures. The logical reason for these results are firstly the selection of appropriate topology parameters of the deep neural network, secondly the use of an effective training methodology, thirdly the creation of new technical indicators, and fourthly the application of various activation functions.
4. The DPO model also achieves the best Sharpe ratio, outperforming the MVO model for the two stock exchanges. This dissertation also addresses two limitations in the MVO model which are the mixed-integer quadratic problem and the short-term trading. The first issue is solved by using the general reduced gradient optimizer, whereas the second is addressed by using the predict expected return instead of the historical mean return. Moreover, such solutions are reconstructed as a deep learning model which has not been applied in earlier research.
5. In terms of implementation time, the dissertation attempted to deal with this issue in several points:
   - Employing enhanced training methods that accelerated convergence time.
   - Proposing approximated activation functions and applying them to find the output of the prediction layer.
   - Exploiting the principle of parallel computing by applying the multi-thread approach.
   - Optimizing portfolio by exploiting a hybrid strategy, which combines the diversification advantages of the Markowitz model with the local search (exploitation) advantages of the GRG method.

6. The standard features satisfy if the change in daily prices values are slight else, the TIs are necessary.

5.2 Recommendations

As for the ISX, after studying and analyzing the results, the dissertation reached several recommendations:

- At the level of the general market, the long-term investment in the ISX is not profitable, it may be very dangerous because of the 43 stocks, 37 stocks are losing within the dataset of the study sample, and this indicator is not good compared to the NYSE, which stocks in the study sample are all profitable. On the other hand, ISX may be more stable than the NYSE at the level of short-term investment because the rate of changes was little and the reason for this is that stock prices range between (0.3-100) Iraqi dinars, while the prices of the American stock exchange are between (100-1200) US dollars.
- At the level of sectors, if the investor desires high venture from a principle, the more turbulent the market, the higher profits are achieved. The banking sector achieves this principle, due to the high volume of trading, as the banking sector represents 78% of the total trading volume of the ISX in the last ten years. But if he wants not to take risks, and according to what DPO model and Markowitz's model have found, the industrial sector is the best option to invest in.
- At the level of stocks, according to MPO model, it is found that IITC company is the best company that could recommend investing in.

5.3 Limitations

Although the dissertation findings are promising, this dissertation is not without limitations that invite further research. The proposed model relies solely on TIs and input features in the inventory forecasting process. However, there are certain restrictions. When it comes to predicting stock prices, DNNs may benefit from the inclusion of financial and economic variables. Moreover, in the stage of selecting stocks as candidates for portfolio formation, they were chosen on the basis of reputation or randomly in this dissertation, they may be improved when they are selected using optimization methods such as fuzzy logic. Furthermore, in portfolio improvement modeling, different risk metrics may be more suitable than a variance. Creating portfolio models based on forecasts and increasing out-of-sample performance will allow future research to include more input functions and better quantify risk. Finally, despite many improvements to the issue of execution time, the proposed system was limited to working with the stock exchange’s offline data.

5.4 Future Works:

Some future works can be viewed below:

1. Applying the proposed model to online SM dataset.
2. Studying other statistical indicators as well as external indicators and indicate their impact on the accuracy of the decision.
3. Studying other methods of optimizing NNs.
4. Applying the proposed model on other financial datasets.
5. Studying the deletion of zeros from the Iraqi currency and the extent of their impact on the recovery of the economy.
6. Applying fuzzy logic instead of activation function.
7. Using pivot (average price) instead of only close price to find the class label.
8. Studying the micro-economic (inflation, political instability) indicators in ISX.
9. Studying different risk metrics may be more suitable than a variance in portfolio optimization model.
10. Selecting the optimal portfolio using technical and fundamental analysis with fuzzy logic.

References

References

Abiodun, O. I., Jantan, A., & Omolara, A. E. (2018). Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. *State-of-the-Art in Artificial Neural Network Applications: A Survey. Heliyon*, *4*(11), e00938.

Abreu, M., & Mendes, V. (2010). Financial literacy and portfolio diversification. *Quantitative Finance*, *10*(5), 515–528.

Adnan, M. H., & Isma'eel, M. M. (2021). *Estimating Stock Returns Using Rough Set Theory: An Exploratory study With An Evidence From Iraq Stock Exchange*.

Aggarwal, C. C. (2018). Neural networks and deep learning. *Springer*, *10*, 973–978.

Aggarwal, R. (2017). Different avenues of capital market (secondary market) available for investing in market of yamuna nagar. *International Research Journal of Management, IT and Social Sciences*, *4*(3), 34–50.

Ahmed, M. K., Wajiga, G. M., Blamah, N. V., & Modi, B. (2019). Stock market forecasting using ant colony optimization based algorithm. *Am J Math Comput Model*, *4*(3), 52–57.

Akbay, M. A., Kalayci, C. B., & Polat, O. (2020). A parallel variable neighborhood search algorithm with quadratic programming for cardinality constrained portfolio optimization. *Knowledge-Based Systems*, *198*, 105944. https://doi.org/10.1016/j.knosys.2020.105944

Alalaya, M. M., Al Rawashdeh, H. A., & Alkhateb, A. (2018). Combination Method between Fuzzy Logic and Neural Network Models to Predict Amman Stock Exchange. *Open Journal of Business and Management*, *6*(3), 632–650.

Almahdi, S., & Yang, S. Y. (2017). An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown. *Expert Systems with Applications*, *87*, 267–279.

Alrabadi, D. W. H. (2016). Portfolio optimization using the generalized reduced gradient nonlinear algorithm: an application to Amman stock exchange. *International Journal of Islamic and Middle Eastern Finance and Management*.

Althelaya, K. A., El-Alfy, E.-S. M., & Mohammed, S. (2018). Evaluation of bidirectional LSTM for short-and long-term stock market prediction. *2018 9th International Conference on Information and Communication Systems*

(ICICS), 151–156.

Appel, G. (2005). *Technical analysis: power tools for active investors*. FT Press.

Asaad, Z. (2014). Testing the Bank Sector at Weak Form Efficiency in Iraq Stock Exchange for Period (2000-2014): An Empirical Study. *Available at SSRN 3069561*.

Bailey, D. H., & Lopez de Prado, M. (2012). The Sharpe ratio efficient frontier. *Journal of Risk*, *15*(2), 13.

Baschab, J., & Piot, J. (2007). *The executive's guide to information technology*. John Wiley & Sons.

Bengio, Y., Roux, N., Vincent, P., Delalleau, O., & Marcotte, P. (2005). Convex neural networks. *Advances in Neural Information Processing Systems*, *18*.

Boenninghoff, B., Hessler, S., Kolossa, D., & Nickel, R. M. (2019). Explainable authorship verification in social media via attention-based similarity learning. *2019 IEEE International Conference on Big Data (Big Data)*, 36–45.

Browne, M. W. (1975). Predictive validity of a linear regression equation. *British Journal of Mathematical and Statistical Psychology*, *28*(1), 79–87.

Chakraborty, M. (2012). Artificial neural network for performance modeling and optimization of CMOS analog circuits. *International Journal of Computer Applications November 2012*. https://doi.org/10.5120/9380-3731

Chen, X., & He, Z.-J. (2015). Prediction of stock trading signal based on support vector machine. *2015 8th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, 651–654.

Chen, Y., & Zhou, A. (2019). MOEA/D with an improved multi-dimensional mapping coding scheme for constrained multi-objective portfolio optimization. *2019 IEEE Congress on Evolutionary Computation (CEC)*, 1742–1749.

Chesney, M., Stromberg, J., Wagner, A. F., & Wolff, V. (2020). Managerial incentives to take asset risk. *Journal of Corporate Finance*, *65*, 101758.

Delen, D., & Ram, S. (2018). Research challenges and opportunities in business analytics. *Journal of Business Analytics*, *1*(1), 2–12.

Démurger, S., Fournier, M., & Yang, W. (2010). Rural households' decisions towards income diversification: Evidence from a township in northern China. *China Economic Review*, *21*, S32–S44.

Deng, L., Li, J., Huang, J.-T., Yao, K., Yu, D., Seide, F., Seltzer, M., Zweig, G., He, X., & Williams, J. (2013). Recent advances in deep learning for speech research at Microsoft. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 8604–8608.

Dennis John E, J., & Moré, J. J. (1977). Quasi-Newton methods, motivation and theory. *SIAM Review*, *19*(1), 46–89.

Dhaini, M., & Mansour, N. (2021). Squirrel search algorithm for portfolio optimization. *Expert Systems with Applications*, *178*, 114968.

Diebold, F. X., & Yilmaz, K. (2009). Measuring financial asset return and volatility spillovers, with application to global equity markets. *The Economic Journal*, *119*(534), 158–171.

Doering, J., Kizys, R., Juan, A. A., Fitó, À., & Polat, O. (2019). Metaheuristics for rich portfolio optimisation and risk management: Current state and future trends. *Operations Research Perspectives*, *6*, 100121.

Du, K.-L., & Swamy, M. N. S. (2013). *Neural networks and statistical learning*. Springer Science & Business Media.

Edwards, R. D., Bassetti, W. H. C., & Magee, J. (2012). *Technical analysis of stock trends*. CRC press.

Edwards, R. D., Magee, J., & Bassetti, W. H. C. (2018). *Technical analysis of stock trends*. CRC press.

Elton, E. J., & Gruber, M. J. (1997). Modern portfolio theory, 1950 to date. *Journal of Banking & Finance*, *21*(11–12), 1743–1759.

Fama, E. F. (1965). Portfolio Analysis in a Stable Paretian Market. *Management Science*, *11*(3), 404–419. https://doi.org/10.1287/mnsc.11.3.404

Fama, E. F., & French, K. R. (2021). *Multifactor explanations of asset pricing anomalies*. University of Chicago Press.

Farrell, J., & Saloner, G. (1985). Standardization, compatibility, and innovation. *The RAND Journal of Economics*, 70–83.

Ferni Ukrit, M., Saranya, A., & Anurag, R. (2020). Stock Market Prediction Using Long Short-Term Memory. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems* (pp. 205–212). Springer.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

Fombellida, J., Martín-Rubio, I., Torres-Alegre, S., & Andina, D. (2020). Tackling business intelligence with bioinspired deep learning. *Neural Computing and Applications*, *32*(17), 13195–13202.

Freitas, F. D., de Almeida, A. R., & de Souza, A. F. (2006). A prediction-based portfolio optimization model. *Fifth International Symposium On Robotics and Automation - ISRA 2006, January 2002*, 520–525.

Friedlob, G. T., & Plewa Jr, F. J. (1996). *Understanding return on investment*. John Wiley & Sons.

Gans, P., & Vacca, A. (1974). Application of the Davidon—Fletcher—Powell method to the calculation of stability constants. *Talanta*, *21*(1), 45–51.

García, S., Luengo, J., & Herrera, F. (2015). *Data preprocessing in data mining* (Vol. 72). Springer.

Glaser, J., & Stone, J. (2008). Effective use of business intelligence: leveraging your organization's business data could improve financial and operational performance--and quality of care. *Healthcare Financial Management*, *62*(2), 68–73.

Gökgöz, F., & Atmaca, M. E. (2012). Financial optimization in the Turkish electricity market: Markowitz's mean-variance approach. *Renewable and Sustainable Energy Reviews*, *16*(1), 357–368.

Gold, S. (2015). The Viability of Six Popular Technical Analysis Trading Rules in Determining Effective Buy and Sell Signals: MACD, AROON, RSI, SO, OBV, and ADL. *Journal of Applied Financial Research*, *2*.

Gosens, J., & Lu, Y. (2013). From lagging to leading? Technological innovation systems in emerging economies and the case of Chinese wind power. *Energy Policy*, *60*, 234–250.

Grinold, R. C., & Kahn, R. N. (2000). *Active portfolio management*.

Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques Third Edition [M]. *The Morgan Kaufmann Series in Data Management Systems*, *5*(4), 83–124.

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Hao, C., Wang, J., Xu, W., & Xiao, Y. (2014). Prediction-based portfolio selection model using support vector machines. *Proceedings - 2013 6th International Conference on Business Intelligence and Financial Engineering, BIFE 2013*, *5*, 567–571.

https://doi.org/10.1109/BIFE.2013.118

Harlow, W. Van. (1991). Asset allocation in a downside-risk framework. *Financial Analysts Journal*, *47*(5), 28–40.

Haykin, S. (1994). Neural Networks: A Comprehensive Foundation, MacMillan College Publishing Co. *New York*.

Hiransha, M., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2018). NSE stock market prediction using deep-learning models. *Procedia Computer Science*, *132*, 1351–1362.

Hyndman, R. J., & Koehler, A. B. (2006). Effect of Question Formats on Item Endorsement Rates in Web Surveys. *International Journal of Forecasting*, *22*(4), 679–688.

Johnson, D. H., Narayan, S., Flask, C. A., & Wilson, D. L. (2010). Improved fat–water reconstruction algorithm with graphics hardware acceleration. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, *31*(2), 457–465.

Kalayci, C. B., Polat, O., & Akbay, M. A. (2020). An efficient hybrid metaheuristic algorithm for cardinality constrained portfolio optimization. *Swarm and Evolutionary Computation*, *54*, 100662.

khalid Chyad, A. (2021). ARIMA Model To Forecast The ISX60 Indicator: An Applied Study An Iraqi Financial Market. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(7), 2549–2554.

Khanna, T., & Palepu, K. G. (2010). *Winning in emerging markets: A road map for strategy and execution*. Harvard Business Press.

Klioutchnikov, I., Sigova, M., & Beizerov, N. (2017). Chaos theory in finance. *Procedia Computer Science*, *119*, 368–375.

Kohonen, T. (1988). *Self organization and associative memory, Springer-Verlag Berlin Germany*.

KON, S. J. (1984). Models of Stock Returns—A Comparison. *The Journal of Finance*, *39*(1), 147–165. https://doi.org/10.1111/j.1540-6261.1984.tb03865.x

Kraus, M., & Feuerriegel, S. (2017). Decision support from financial disclosures with deep neural networks and transfer learning. *Decision Support Systems*, *104*, 38–48.

Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European*

Journal of Operational Research, *259*(2), 689–702. https://doi.org/10.1016/j.ejor.2016.10.031

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *25*.

Laffont, J.-J., & Maskin, E. S. (1990). The efficient market hypothesis and insider trading on the stock market. *Journal of Political Economy*, *98*(1), 70–93.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Lee, S. Il, & Yoo, S. J. (2020). Threshold-based portfolio: the role of the threshold and its applications. *Journal of Supercomputing*, *76*(10), 8040–8057. https://doi.org/10.1007/s11227-018-2577-1

Leonard, J., & Kramer, M. A. (1990). Improvement of the backpropagation algorithm for training neural networks. *Computers & Chemical Engineering*, *14*(3), 337–341.

Levy, H., & Levy, M. (2014). The benefits of differential variance-based constraints in portfolio optimization. *European Journal of Operational Research*, *234*(2), 372–381.

Lim, S., Kim, M.-J., & Ahn, C. W. (2020). A Genetic Algorithm (GA) Approach to the Portfolio Design Based on Market Movements and Asset Valuations. *IEEE Access*, *8*, 140234–140249.

Lin, C. M., Huang, J. J., Gen, M., & Tzeng, G. H. (2006). Recurrent neural network for dynamic portfolio selection. *Applied Mathematics and Computation*, *175*(2), 1139–1146. https://doi.org/10.1016/j.amc.2005.08.031

Luo, X., O'Brien, W. J., & Julien, C. L. (2011). Comparative evaluation of Received Signal-Strength Index (RSSI) based indoor localization techniques for construction jobsites. *Advanced Engineering Informatics*, *25*(2), 355–363.

Ma, Y., Han, R., & Wang, W. (2020). Prediction-Based Portfolio Optimization Models Using Deep Neural Networks. *IEEE Access*, *8*, 115393–115405. https://doi.org/10.1109/ACCESS.2020.3003819

Malkiel, B. G. (2003). The efficient market hypothesis and its critics. *Journal of Economic Perspectives*, *17*(1), 59–82.

Maringer, D., & Kellerer, H. (2003). Optimization of cardinality constrained portfolios with a hybrid local search algorithm. *Or Spectrum*, *25*(4), 481–495.

Markowitz, H M. (1952). Portfolio Selection/Harry Markowitz. *The Journal of Finance*, *7*.

Markowitz, Harry M. (2010). Portfolio theory: as I still see it. *Annu. Rev. Financ. Econ.*, *2*(1), 1–23.

Markowitz, S., & Grossman, M. (1998). Alcohol regulation and domestic violence towards children. *Contemporary Economic Policy*, *16*(3), 309–320.

Melicher, R. W., & Norton, E. A. (2013). *Introduction to finance: markets, investments, and financial management*. John Wiley & Sons.

Michaud, R. O., & Michaud, R. O. (2008). *Efficient asset management: a practical guide to stock portfolio optimization and asset allocation*. Oxford University Press.

Mohamed, E. A., Ahmed, I. E., Mehdi, R., & Hussain, H. (2021). Impact of corporate performance on stock price predictions in the UAE markets: Neuro-fuzzy model. *Intelligent Systems in Accounting, Finance and Management*, *28*(1), 52–71.

Moré, J. J., & Sorensen, D. C. (1982). *Newton's method*. Argonne National Lab., IL (USA).

Navale, G. S., Dudhwala, N., Jadhav, K., Gabda, P., & Vihangam, B. K. (2016). Prediction of stock market using data mining and artificial intelligence. *International Journal of Computer Applications*, *134*(12), 9–11.

Nazerfard, E., & Cook, D. J. (2015). CRAFFT: an activity prediction model based on Bayesian networks. *Journal of Ambient Intelligence and Humanized Computing*, *6*(2), 193–205.

NekoeiQachkanloo, H., Ghojogh, B., Pasand, A. S., & Crowley, M. (2019). Artificial counselor system for stock investment. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 9558–9564.

Norton, E., & Tenenbaum, B. H. (1993). Specialization versus diversification as a venture capital investment strategy. *Journal of Business Venturing*, *8*(5), 431–442.

Norton, M., Khokhlov, V., & Uryasev, S. (2019). Calculating CVaR and bPOE for common probability distributions with application to portfolio optimization and density estimation. *Annals of Operations Research*, 1–35.

Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *ArXiv Preprint ArXiv:1811.03378*.

O'Grady, J. G., Alexander, G. J. M., Hayllar, K. M., & Williams, R. (1989). Early indicators of prognosis in fulminant hepatic failure. *Gastroenterology*, *97*(2), 439–445.

Ogneva, M. (2012). Accrual quality, realized returns, and expected returns: The importance of controlling for cash flow shocks. *The Accounting Review*, *87*(4), 1415–1444.

Omisore, I., Yusuf, M., & Christopher, N. (2011). The modern portfolio theory as an investment decision tool. *Journal of Accounting and Taxation*, *4*(2), 19–28.

Orsolic, I., Pevec, D., Suznjevic, M., & Skorin-Kapov, L. (2017). A machine learning approach to classifying YouTube QoE based on encrypted network traffic. *Multimedia Tools and Applications*, *76*(21), 22267–22301.

Oztemel, E., & Gursev, S. (2020). Literature review of Industry 4.0 and related technologies. *Journal of Intelligent Manufacturing*, *31*(1), 127–182.

Paiva, F. D., Cardoso, R. T. N., Hanaoka, G. P., & Duarte, W. M. (2019). Decision-making for financial trading: A fusion approach of machine learning and portfolio selection. *Expert Systems with Applications*, *115*, 635–655.

Pandey, D. K., & Kumari, V. (2020). Effects of merger and acquisition announcements on stock returns: an empirical study of banks listed on NSE & NYSE. *The Review of Finance and Banking*, *12*(1).

Pang, X., Zhou, Y., Wang, P., Lin, W., & Chang, V. (2020). An innovative neural network approach for stock market prediction. *The Journal of Supercomputing*, *76*(3), 2098–2118.

Panigrahi, S. S., & Mantri, J. K. (2015). A text based Decision Tree model for stock market forecasting. *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, 405–411.

Perry, A. (1978). A modified conjugate gradient algorithm. *Operations Research*, *26*(6), 1073–1078.

Platanakis, E., & Urquhart, A. (2020). Should investors include bitcoin in their portfolios? A portfolio theory approach. *The British Accounting Review*, *52*(4), 100837.

Riedmiller, M., & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. *IEEE International Conference on Neural Networks*, 586–591.

Robertson, A. (1959). The sampling variance of the genetic correlation coefficient. *Biometrics*, *15*(3), 469–485.

Rubin, E., & Rubin, A. (2013). The impact of business intelligence systems on stock return volatility. *Information & Management*, *50*(2–3), 67–75.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *ArXiv Preprint ArXiv:1609.04747*.

Sahoo, S., & Mohanty, M. N. (2020). Stock market price prediction employing artificial neural network optimized by gray wolf optimization. In *New paradigm in decision science and management* (pp. 77–87). Springer.

Schlossberg, B. (2006). *Technical analysis of the currency market: classic techniques for profiting from market swings and trader sentiment*. John Wiley & Sons.

Sharpe, W. F. (1994). The sharpe ratio. *Journal of Portfolio Management*, *21*(1), 49–58.

Silva, Y. L. T. V., Herthel, A. B., & Subramanian, A. (2019). A multi-objective evolutionary algorithm for a class of mean-variance portfolio selection problems. *Expert Systems with Applications*, *133*, 225–241. https://doi.org/10.1016/j.eswa.2019.05.018

Singer, T. (2001). Information Engineering. *Plant Engineering*, *55*(1), 30–32.

Skolpadungket, P., Dahal, K., & Harnpornchai, N. (2007). Portfolio optimization using multi-obj ective genetic algorithms. *2007 IEEE Congress on Evolutionary Computation*, 516–523.

Starace, F. (2008). The utility industry in 2020. In *Handbook utility management* (pp. 147–167). Springer.

Sun, Y., Wang, X., & Tang, X. (2014). Deep learning face representation from predicting 10,000 classes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1891–1898.

Svozil, D., Kvasnicka, V., & Pospichal, J. (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems*, *39*(1), 43–62.

Sze, V., Chen, Y.-H., Yang, T.-J., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, *105*(12), 2295–2329.

Ta, V.-D., Liu, C.-M., & Tadesse, D. A. (2020). Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading. *Applied Sciences*, *10*(2), 437.

Talmon, S. (2013). *The Occupation of Iraq: Volume 2: The Official Documents of the Coalition Provisional Authority and the Iraqi Governing Council*. Bloomsbury Publishing.

Tan, P.-N., Steinbach, M., & Kumar, V. (2016). *Introduction to data mining*. Pearson Education India.

Wang, S., Liu, W., Wu, J., Cao, L., Meng, Q., & Kennedy, P. J. (2016). Training deep neural networks on imbalanced data sets. *2016 International Joint Conference on Neural Networks (IJCNN)*, 4368–4374.

Waweru, N. M., Munyoki, E., & Uliana, E. (2008). The effects of behavioural factors in investment decision-making: a survey of institutional investors operating at the Nairobi Stock Exchange. *International Journal of Business and Emerging Markets*, *1*(1), 24–41.

Willmott, C. J., Ackleson, S. G., Davis, R. E., Feddema, J. J., Klink, K. M., Legates, D. R., O'donnell, J., & Rowe, C. M. (1985). Statistics for the evaluation of model performance. *J. Geophys. Res*, *90*(C5), 8995–9005.

Wood, R. A., McInish, T. H., & Ord, J. K. (1985). An investigation of transactions data for NYSE stocks. *The Journal of Finance*, *40*(3), 723–739.

Wu, S., Li, G., Deng, L., Liu, L., Wu, D., Xie, Y., & Shi, L. (2018). $ L1 $-norm batch normalization for efficient training of deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(7), 2043–2051.

Wu, S. X., & Banzhaf, W. (2010). The use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing*, *10*(1), 1–35.

Yang, F., Chen, Z., Li, J., & Tang, L. (2019). A novel hybrid stock selection method with stock prediction. *Applied Soft Computing Journal*, *80*, 820–831. https://doi.org/10.1016/j.asoc.2019.03.028

Yefimochkin, O. (2011). Fundamental: using macroeconomic indicators and genetic algorithms in stock market forecasting. *Unpublished Master's Thesis, Department of Computer Engineering, The Technical University of Lisbon, Portugal*, 1–120.

Yeniay, O. (2005). A comparative study on optimization methods for the constrained nonlinear programming problems. *Mathematical Problems in Engineering*, *2005*(2), 165–173.

Yong, B. X., Abdul Rahim, M. R., & Abdullah, A. S. (2017). A stock market trading system using deep neural network. *Asian Simulation Conference*, 356–364.

Yuan Song. (2018). *Stock Trend Prediction:Based on Machine Learning Methods* [University of California,]. https://escholarship.org/uc/item/0cp1x8th#author

Zaman, U., Nadeem, R. D., & Nawaz, S. (2020). Cross-country evidence on project portfolio success in the Asia-Pacific region: Role of CEO transformational leadership, portfolio governance and strategic innovation orientation. *Cogent Business & Management*, *7*(1), 1727681.

Zellweger, T. (2007). Time horizon, costs of equity capital, and generic investment strategies of firms. *Family Business Review*, *20*(1), 1–15.

Zhang, J., Cui, S., Xu, Y., Li, Q., & Li, T. (2018). A novel data-driven stock price trend prediction system. *Expert Systems with Applications*, *97*, 60–69.

Zhou, X. Y., & Li, D. (2000). Continuous-time mean-variance portfolio selection: A stochastic LQ framework. *Applied Mathematics and Optimization*, *42*(1), 19–33.

الملخص

تعد صعوبة تطوير وإدارة محفظة استثمارية فعالة والتنبؤ بسوق الأوراق المالية من أكثر القضايا تعقيدًا التي تواجه القطاع المالي اليوم ، خاصة في ظل المنافسة المتزايدة والتغيرات الاقتصادية العالمية على المستويين الوطني والدولي. وبالتالي ، تعد إدارة المحافظ وتحسينها جزءًا مهمًا من نظام ذكاء الاعمال (BI).

على الرغم من حقيقة أن التعلم العميق قد تناول مجموعة واسعة من القضايا في العديد من المجالات ، إلا أنه لا يزال يواجه مشاكل في بعض مجالات التمويل ، مثل إدارة المحافظ. نتيجة لذلك ، تبحث هذه الأطروحة في كيفية استخدام التعلم العميق لحل مشكلات تحسين المحفظة والتنبؤ بأسعار أسهمها ، وكذلك ما إذا كان ينبغي استبدال مبادئ نظرية المحفظة التقليدية بالكامل بشبكة عصبية عميقة متكاملة شاملة أو تضمينها في هيكلها.

للتعامل مع هذه القضايا ، تهدف هذه الأطروحة إلى اقتراح نظام شبكة عصبية عميقة متكاملة لتوفير إرشادات واضحة لمديري الصناديق والمستثمرين الأفراد. يمكن للنظام المقترح التنبؤ بالسعر في سوق الأوراق المالية. بالإضافة إلى ذلك ، يمكن أن يؤدي إلى اختيار الاستراتيجيات الصحيحة لاتخاذ قرارات الاستثمار ، وهذا بدوره يمكن أن يعزز الإيرادات خلال فترات عدم اليقين. يشتمل النظام المقترح على ثلاثة نماذج هي تصنيف التعلم العميق (DLC) ، وانحدار التعلم العميق (DLR) ، وتحسين المحفظة العميقة .(DPO) يتم تطبيق هذه النماذج الثلاثة تلقائيًا من المدخلات إلى اتخاذ القرار.

تعمل نماذج DLR و DLC على تحسين سرعة التقارب في تدريب الطبقات الكثيفة الكاملة باستخدام طريقة Quasi Newton بدلاً من Gradient Descent . وفقًا لذلك ، يمكن لهذه النماذج التنبؤ بسعر إغلاق الأسهم بالإضافة إلى اتخاذ قرار البيع أو الشراء أو الاحتفاظ بالأسهم وإنتاج دقة عالية في وقت معقول. بينما ، تم تصميم نموذج DPO لمعالجة اثنين من المشاكل في نظرية المحفظة التقليدية وهما المشكلة التربيعية المختلطة ذات العدد الصحيح والتداول قصير الأجل. يتم حل المشكلة الأولى باستخدام مُحسِّن التدرج العام المنخفض ، بينما تتم معالجة المشكلة الثانية باستخدام التنبأ بالعائد المتوقع بواسطة نموذج DLR بدلاً من متوسط العائد التاريخي. علاوة على ذلك ، يتم إعادة بناء هذه الحلول كنموذج تعلم عميق الذي لم يتم تطبيقه في البحوث السابقة.

يتم تطبيق مجموعة بيانات عالمية ومحلية حقيقية ، ممثلة في بورصة نيويورك (NYSE) وسوق العراق للأوراق المالية (ISX) ، لتحديد سلوك النظام المقترح. تم اختيار خمسة وعشرين سهماً مشهوراً من بورصة نيويورك من عام 2010 إلى عام 2016 وثلاثة وأربعين سهماً من بورصة العراق من 2013 إلى 2020 ، والتي تم بناؤها وهيكلها.

تدعم النتائج التجريبية متانة النماذج المقترحة ، أولاً أثبت نموذج DLC كفاءته من حيث الدقة ووقت التنفيذ ، وبالتالي بلغ معدل الدقة (97.942) و (98.116) لمجموعة بيانات ISX و NYSE على التوالي. ثانيًا ، تفوق نموذج DLR أيضًا على دقة التنبؤ للبحث السابق بناءً على مقاييس MAE و RMSE حيث وصل (0.32094 و 0.49347) في بورصة نيويورك بينما (0.084 ، 0.1879) في بورصة العراق وفقًا لـ MAE وRMSE . ثالثًا ، أنتج نموذج DPO أكبر نسبة شارب ، مع زيادة الأرباح ، وتقليل المخاطر في المحفظة ، حيث شكل المحفظة بنسبة شارب (1.406 ، 0.944) لمجموعة بيانات ISX و NYSE.

بالإضافة إلى النتائج الجيدة التي تم الحصول عليها ، هناك العديد من المزايا للنظام المقترح المتعلقة بالجانب العملي ، بما في ذلك أولاً: استخدام مبدأ تعدد الخيوط (multithread) في معظم طبقات DNN ، وكان لتطبيق هذا المبدأ دور فعال في تقليل الوقت المستغرق من n إلى 1. ثانيًا: استخدام طريقة المشتقة الثانية GN في طبقة التنبؤ له ميزة تقليل وقت التقارب لتدريب الشبكة من 45 دقيقة إلى 0.75 ثانية. ثالثًا ، أدى اقتراح وظيفة التنشيط التقريبية بناءً على مبدأ سلسلة Toiler أيضًا إلى تقليل وقت التنفيذ بنسبة 100٪ تقريبًا.

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعـــة بـابــل

كلية تكنولوجيا المعلومات -قسم البرمجيات

تحليل البيانات وإدارة ذكاء الأعمال باستخدام التعلم العميق المطور

اطروحة مقدمة إلى

مجلس كلية تكنولوجيا المعلومات -جامعة بابل كجزء من متطلبات
نيل درجة الدكتوراه في تكنولوجيا المعلومات- برمجيات

من قبل

أمير الحق عادل صاحب علي

بإشـراف

أ.د. إيمان صالح صكبان الشمري

1443 ه

2022 م