AN ENSEMBLE OF NEURAL NETWORK AND MODIFIED GREY WOLF OPTIMIZER FOR STOCK PREDICTION

DEBASHISH DAS

DOCTOR OF PHILOSOPHY
(COMPUTER SCIENCE)

UNIVERSITI MALAYSIA PAHANG

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Doctor of Philosophy.

____

(Supervisor's Signature)

Full Name : DR ALI SAFAA SADIQ

Position : SENIOR LECTURER

Date : APRIL, 2019

____

(Co-supervisor's Signature)

Full Name : DR NORAZIAH BINTI AHMAD

Position : ASSOCIATE PROFESSOR

Date : APRIL, 2019

Universiti
Malaysia
PAHANG
Engineering • Technology • Creativity

STUDENT’S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

____

(Student’s Signature)

Full Name : DEBASHISH DAS

ID Number : PCC15003

Date : APRIL, 2019

AN ENSEMBLE OF NEURAL NETWORK AND MODIFIED GREY WOLF OPTIMIZER FOR STOCK PREDICTION

DEBASHISH DAS

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Doctor of Philosophy

Faculty of Computer Systems & Software Engineering

UNIVERSITI MALAYSIA PAHANG

APRIL 2019

ACKNOWLEDGEMENTS

Foremost, this thesis would not have been possible without the financial support from Doctoral Scholarship Scheme (DSS), University Malaysia Pahang. I would like to thank Faculty of Computer Systems & Software Engineering (FSKKP) and Institute of Postgraduate Studies (IPS) for supporting this great opportunity.

I would like to express my gratitude to my supervisor, Dr. Ali Safa Sadiq, and my co-supervisor, Associate Professor Dr. Noraziah Ahmad for their sincere dedication, useful comments, guidance, advice and support of this thesis. I am really appreciative.

I also would like thank Professor Dr. Kamal Z. Zamli and Dr Junaida for useful suggestions on reviewing my draft thesis and others lecturers that I have learned from. My thanks also go to Dr. Nizam, Associate Professor Dr. Mazlina and FSKKP staffs, for their good support for research students to international level.

I would like to thank UMP for supporting me in receiving Postgraduate Research Grant (PGRS), attending conferences and publishing journals. I also acknowledge with thank to UMP library staffs for their works to provide convenient environments for doing this research.

I would like to thank Asia Pacific University of Technology and Innovation, Malaysia and University of Liberal Arts Bangladesh (ULAB) for assisting me to attain PhD. My warmth thanks also go to my friends and colleagues for encouraging, supporting, assisting and sharing emotions.

Finally, I dedicate this work to my parents, Dilip Kumar Das and Sabita Das, my wife, Tanuza, my son Adrik. Unforgettable thanks to my sisters and families who support and look after me and are always beside me.

ABSTRAK

Pengoptimuman berkaitan dengan proses mencari penyelesaian optimum (sama ada memaksimumkan atau meminimumkan) kepada masalah tertentu yang memenuhi beberapa kekangan yang diberikan. Disebabkan kesederhanaan dan kelenturannya, meta-heuristik telah terbukti berkesan untuk menyelesaikan masalah pengoptimuman. Sehingga kini, terdapat banyak meta-heuristik yang telah dibangunkan dalam bidang penyelidikan. Selaras dengan teorem 'No Free Lunch' yang menunjukkan bahawa tiada meta-heuristik tunggal, yang terbaik untuk semua masalah pengoptimuman, tetapi mencari algoritma yang lebih baik masih merupakan usaha yang membuahkan hasil. Grey Wolf Optimizer (GWO) merupakan algoritma meta-heuristik terkini yang menarik perhatian kebanyakan penyelidik kerana prestasi unggulnya yang disebut dalam kajian literatur. Walaupun GWO menunjukkan prestasi yang tinggi, ia juga ada kelemahannya. Pada masa kini, keoptimuman GWO adalah berat sebelah terhadap GWO jenis alfa dan jenis yang lain (iaitu beta dan delta) masing-masing cuba untuk mengubah kedudukannya ke arah yang terbaik dalam setiap proses ulangan. Proses kemaskini ini boleh menyebabkan algoritma ini bergerak ke optima tempatan terutamanya dalam kes-kes di mana terdapat banyak optima tempatan yang bersaing. Oleh itu, penyelidikan ini cuba mengubahsuai GWO untuk menangani batasan GWO dengan penambahbaikan penerokaan dengan menguatkan proses pencarian melalui beberapa pemimpin rawak dalam setiap lelaran, menghasilkan semula pemimpin rawak dalam setiap lelaran dan memperkenalkan arkib untuk mengesahkan penyelesaian dengan kebarangkalian yang lebih baik untuk teruskan latihan dan penjanaan semula. Pengesahan setiap penyelesaian secara individu oleh Modified GWO, dan bukannya dipertimbangkan sebagai penyelesaian akhir, memudahkan peningkatan penerokaan. Selain itu, penyelidikan mengehadkan bilangan pembolehubah melalui pemilihan ciri untuk meningkatkan prestasi algoritma. Selepas itu, penyelidikan cuba untuk membina model ensemble menggunakan Modified Gray Wolf Optimizer (MGWO) dan rangkaian neural untuk ramalan saham. Model-model yang meluas seperti Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Ant Colony Optimization (ACO), Evolutionary Strategies (ES) dan Probability Based Incremental Learning (PBIL) yang berurusan dengan masalah tertentu juga diterokai dan dibandingkan. Kajian ini melaksanakan analisis ramalan saham sebagai kajian kes untuk melatih rangkaian saraf dengan mengadopsi algoritma MGWO. Dalam kajian ini, data dikumpulkan dari pasaran saham terkenal; New York Stock Exchange (NYSE), NASDAQ dan pasaran baru muncul; Dhaka Stock Exchange (DSE), Bursa Malaysia. Selain itu, pelbagai data faktor seperti harga Dolar, harga Emas, kadar faedah Bank, Pelaburan Langsung Asing, dan Inflasi dikumpulkan untuk mengukur kesan dalam pasaran saham. K-means clustering digunakan untuk memilih syarikat yang sangat menjanjikan; MGWO dilaksanakan untuk pemilihan dan latihan ciri; akhirnya, MGWO-NN digunakan untuk meramalkan harga saham. Model "ensemble" yang dipilih di sini untuk mencapai prestasi ramalan yang lebih baik, digunakan untuk meramalkan harga pasaran masa hadapan. Pendekatan yang dicadangkan mengatasi algoritma meta-heuristik sedia ada. Khususnya, model yang dicadangkan mencapai 97% kadar klasifikasi, 95% ramalan tepat dan kadar kesilapan yang kurang daripada 2.0. Sebagai kesimpulan, kejayaan pelaksanaan model MGWO dan ensemble menjadikan sumbangan yang berharga kepada arena saintifik.

ABSTRACT

Optimization relates to the process of finding the optimum solution (either maximize or minimize) to a particular problem satisfying some given constraints. Owing to its simplicity and flexibility, meta-heuristics have been proven to be effective for solving optimization problems. To date, there are many meta-heuristics have been developed in the literature. In line with the No Free Lunch theorem which suggests that no single meta-heuristic is the best for all optimization problems, the search for better algorithms is still a worthy endeavour. Grey Wolf Optimizer (GWO) is a recently developed meta-heuristic algorithm which is appealing to researcher owing to its demonstrated performance as cited in the scientific literature. Despite its performances, GWO is not without limitation. Precisely, the current best optimal individual of GWO is biased toward alpha and other individuals (e.g. beta and delta) attempt to modify their positions toward this best individual in each iteration process. This update process may cause the algorithm to fall to local optima especially in the cases where there are many competing local optima. Therefore, the research attempts to modify GWO to addresses the limitation of GWO for improvement of exploration by strengthen the searching process via several random leaders in each iteration, re-generating the random leaders in each iteration and introducing archive to verify the solution with better probability to proceed further for training and re-generation. The verification of each solution individually by Modified GWO, instead of considering as a final solution, facilitates the improvement of the exploration. Additionally, the research restricts the number of variables through feature selection to enhance the performance of the algorithm. Subsequently, the research attempts to construct an ensemble model applying Modified Grey Wolf Optimizer (MGWO) and neural network for stock prediction. Widespread models like Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Ant Colony Optimization (ACO), Evolutionary Strategy (ES) and Population-Based Incremental Learning (PBIL) dealing with the specified problems are also explored and compared. The research implements stock prediction analysis as a case study for training the neural network by adopting MGWO algorithm. In this research, data is collected from reputed stock markets; New York Stock Exchange (NYSE), NASDAQ and emerging markets; Dhaka Stock Exchange (DSE), Bursa Malaysia. Moreover, various factors data like Dollar price, Gold price, Bank interest rate, Foreign Direct Investment, and Inflation are collected to measure the effect in stock market. K-means clustering is applied to select the highly promising company; MGWO is implemented for feature selection and training; finally, MGWO-NN is applied to predict the stock price. The “ensemble” model selected here to achieve better predictive performance, is used to predict future market price. The proposed approach outperforms existing available meta-heuristic algorithms. Specifically, the proposed model achieved 97% classification rate, 95% precise prediction and less than 2.0 error rate. In conclusion, the successful implementation of MGWO and ensemble model makes a valuable contribution to scientific arena.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

A	Shared archived score
a_i	Actual Price
$A(k)$	Input of the Neural Network
d	Squared Euclidean Distance
f	Non-linear Function
$\mathrm{G}_{\alpha}^{\mathrm{j}}$	Sum of All the Best Solution
k	Number of Clusters
N	Average Distance between Wolves
p_i	Predicted Price
$P(k)$	Predicted Output
Q_l	Mean
s	Slope
t	Number of Iteration
X	Dataset
$Yn\times k$	Partition Matrix
Π	Vectors

LIST OF ABBREVIATIONS

ACO	Ant Colony Optimization
ANN	Artificial Neural Network
BA	Bat Algorithm
BBO	Bio Geography Based Optimization
CS	Cuckoo Search
DSE	Dhaka Stock Exchange
ES	Evolutionary Strategies
FA	Firefly Algorithm
GA	Genetic Algorithm
GWO	Grey Wolf Optimizer
MAD	Mean Absolute Deviation
MAPE	Mean Absolute Percentage Error
MGWO	Modified Grey Wolf Optimizer
NARX	Non Linear Auto Regressive Exogenous
NFL	No Free Lunch
NYSE	New York Stock Exchange
PBIL	Population Based Incremental Learning
PSO	Particle Swarm Optimization
RMSE	Root Mean Squared Error
RWH	Random Walk Hypothesis
SVM	Support Vector Machine
EA	Evolutionary Algorithm
SIV	Suitability Index Variable
MLP	Multi-Layer Perceptron
HSI	Habitat Suitability Index

CHAPTER 1

INTRODUCTION

1.1 Introduction

This chapter exhibits the foundation to the thesis explaining on the idea of optimization and its pertinence in scientific and industrial procedures everywhere throughout the world. Additionally, the chapter also traces the different sorts of optimization algorithms accessible in literature. Finally, the chapter covers the problem statement, objectives, scope, significance and the organization of the thesis.

1.2 Optimization in Science and Engineering

Computer Science has emerged as a discipline for both theoretical investigation and experimentation which can solve a seemingly difficult problem perhaps by reducing, embedding, transformation, or simulation. Computer science involves solving problems, designing systems and understanding human behavior, by drawing on the concepts fundamental to computer science. It often uses massive amounts of data to speed up computation. An exceptionally regular thought in scientific, business and engineering configuration is the issue of cost and serviceability, in this manner featuring the requirement for optimization. Similarly as business associations are worried about expanding benefit, so engineering-design associations are worried about persistently boosting the productivity of the structured items and researchers are consistently looking into to acquire better outcomes with less contributions of time and materials. There is not really any field of human undertaking today going from medicine, pharmacy, science, engineering to business management that ignores the place of optimization. Optimization is at the core of decision-making in manufacturing and mechanical concerns and is a veritable apparatus in the examination of physical frameworks (Gigenrenzer and Gaissmaier, 2011). Basically, optimization is dealing about finding the best arrangement out of a few possible arrangements. In scientific terms, optimization manages the look for the ideal object among a few items, particularly in circumstances where an entire feasible search is outlandish (Vazan and Tanuska, 2012).

The optimization is such an arena which can be applicable to attain an optimal solution containing discrete or continuous feasible solutions. Taking all things together, it can be stated that the general goal of either continuous or discrete optimization is to maximize or minimize a function. Alternatively, optimization is the economics of science and engineering with the fact of augmenting benefit, limiting expenses, industrial procedures or time utilization (Miller and Rubinovich, 2012).

Various types of well-known optimizations are available in literature such as Combinatorial Optimization (Wolsey & Nemhauser, 2014), Complementarity Problems (Huang & Ni, 2010), Constrained Optimization (Bertsekas, 2014), Unconstrained Optimization (Tuba et al., 2011), Continuous Optimization (Crandall et al., 2011), Discrete Optimization (Kouvelis & Yu, 2013), Global Optimization (Horst & Tuy, 2013), Integer Linear Programming (Morais et al., 2010), Linear Programming (LP) (Bazaraa et al., 2011), Network Optimization (Xie et al., 2010), Non-differentiable Optimization Nonlinear Equations (R. Rao, V. Savsani & D. Vakharia, 2012), Optimization Under Uncertainty (Conti et al., 2009), Quadratically-Constrained Quadratic Programming (QCQP) (Anstreicher, 2012), Quadratic Programming (QP) (Rodriguez-Lujan et al., 2010), Semidefinite Programming (SDP) (Wolkowicz et al., 2012), Semi-infinite Programming (SIP) (Sivaramakrishnan, 2002), Stochastic Linear Programming (SLP) (Higle & Sen, 2013), Second Order Cone Programming (SOCP) (Le et al., 2009), Stochastic Programming (Birge & Louveaux, 2011), Nonlinear Programming (Kuhn, 2014), Nonlinear Least-Squares Problems (Gratton et al., 2007), Mixed Integer Nonlinear Programming (MINLP) (Lee & Leyffer, 2011), Bound Constrained Optmization (Morales & Nocedal, 2011), Mathematical Programs with Equilibrium Constraints (MPEC) (Luo et al., 1996), Multi-Objective Optimization (Deb, 2014) and Derivative-Free Optimization (Rios & Sahinidis, 2013). However, this study has categorized these optimizations into two general categories such as discrete and continuous.

Minimizing or maximizing a function using continuous, real numbers by accepting value points from integer set to other set is known as Continuous optimization that contains negative values, decimals or fractions (Horst & Tuy, 2013). So, continuous optimization can take numerical values to make those values appear both in the real world and in the abstract mathematical world. Therefore, some experts believe that continuous optimization is more accurate and complex than its discrete counterpart (Streiner et al., 2014). However, many other experts oppose the finding (Devenport, 2013).

Conversely, a subclass of optimization, known as discrete optimization that can use integers as opposed to decimals or fractions and execute minimization or maximization of functions. Combinatorial optimization and integer programming are the two subdivisions of discrete optimization (Nemhauser & Bienstock, 2005). Precisely, the current study concentrates on developing Nature-inspired optimization algorithm that achieves the solution for continuous or discrete optimization problems stochastically.

In the previous couple of decades in scientific and engineering research, Nature-inspired algorithms are becoming progressively prevalent everywhere throughout the world. Researchers are getting excited by this improvement and have illustrated a few purposes behind this: a portion of these causes are that they are created to mimic the best elements in biological, chemical and physical processes in nature. This circumstance hurls the issue of deciding appropriate algorithm at whatever point a researcher has an optimization issue to solve. Usually, there is a common belief among the researchers that the decision of the 'best' algorithm to tackle a specific issue depends to a great extent on the kind of issue one is faced with. However, there is no such suggested guidelines on a decision of algorithm available for large-scale, non-linear optimization problems settling (Xu et al., 2012).

Meta-heuristic algorithms are prominent over few decades for solving difficult problems not only in computer science but also for other fields since they are inspired by very simple natural selection concepts. Physical phenomena, animal behaviours and evolutionary concepts are the typical inspirations of meta-heuristic that facilitates the computer scientist to learn meta-heuristic, simulate various concepts, ensemble meta-heuristic with other algorithms, hybridize one with another, or improve existing meta-heuristic. Hence, the application of meta-heuristic algorithm to solve complex prediction problem consisting non-linear nature of data is a distinct research area that requires appropriate investigation. In a nut shell, meta-heuristic algorithms rely on two main components to perform the search process. Exploration is the process of roaming the entire search space to ensure sufficient diversity of the potential solutions. Exploration is the process of exploiting the known best to ensure that the obtained solution is the most optimal. Excessive exploration tend to increase the computation and may lead to poor convergence. On the other hand, excessive exploitation can make the search process trapped in local optima. For these reasons, there is a need to balance between exploration and exploitation.

Given the aforementioned features, meta-heuristic algorithms can be applied for training neural network even though each algorithm has limitations. Some of the prominent meta-heuristic algorithms include Genetic Algorithm (GA) (Garakani et al., 2018, Samadzadegan et al., 2010), Particle Swarm Optimisation (PSO) (Garakani et al., 2018, Bao et al., 2013 and Blondin et al., 2010), Bat Algorithm (BA) (Tuba et al., 2016a), Firefly Algorithm (FA) (Tuba et al., 2016b), Cuckoo Search (CS) (Puntura et al., 2016), and Grey Wolf Optimiser (GWO) (Mirjalili et al., 2014b and Eswaramoorthy et al., 2016). However, no heuristic algorithm is the best suited to solve all optimization problems (Yang, 2012). Moreover, limitations of expensive computational cost, occurrence of premature convergence, mutation rate, crossover rate, time consuming fitness evaluation leads to enhance existing algorithm or propose new one. In machine learning, classification is a supervised learning process to determine appropriate dataset for a new observation based on the performance through training set. Evolutionary or nature-inspired algorithms are good option for classification. Support Vector Machine (SVM) is an efficient supervised learning algorithm that can be applied for classification. The optimization of SVM parameters are possible through algorithms like GA, PSO, BAT, FA, and GWO. The feature selection is a vital part of classification accuracy model and the parameter optimization of SVM through the application of meta-heuristic algorithms which can simultaneously achieve the feature selection. The feature selection through this process is another extension of distinct research dimension (Wei et al., 2017). However, SVM devises limitations such as: computationally expensive, high algorithmic complexity, extensive memory requirements, and selection of appropriate kernel parameters may be tricky (Sagar, 2015 and Patel et al., 2015). Specifically, a problem well-handled by a meta-heuristic may not produce same inspiring result for another problem.

The Grey Wolf Optimizer (GWO) is very efficient for searching that can contribute for classification, feature selection and learning (Faris et al, 2018, Mirjalili et al., 2014b). For this reason, there is a pressing need to undertake further study to gain complete understanding of the potential offered by this novel algorithm.

This chapter describes the background of study and the challenge statement of the research. This can be within the objectives associated scope of the research to provide an early understanding on the research. The numerous of study and outline of the thesis organization are going to be outlined in this section.

1.3 Problem Statement

In this digital era, huge amount of data is stored and processed all over the world. But, the most challenging task is to extract the useful information from the huge amount of data (Kumar, 2014) and hence an appropriate algorithm is required to be developed for exploring the data. Researchers proposed numerous models to achieve good accuracy in prediction through processing large amount of data although, no single model is dominant over the other (Nguyen et al., 2015).

Thus, many researchers are fascinated to investigate the area of soft computing due to the higher demand of intelligent system in recent times. A portion of the exceptionally prevalent studies incorporate the Ant Colony Optimization (Dorigo, 1992), Bat Algorithm (X.– S. Yang, 2010), Particle Swarm Optimization (Eberhart and Kennedy, 1995), and numerous others. These techniques have been effectively implemented to take care of numerous combinatorial issues, for example, Traveling Salesman's Problem, Job scheduling, and vehicle routing, just to specify a couple.

Neural network and Support Vector Machine (SVM) are good choices of classifiers for data classification and prediction. However, the accuracy of the prediction depends heavily on the learning that needs proper investigation to determine an appropriate training algorithm (Faris et al., 2018, Wang et al, 2016, Mirjalili et al., 2014a, Mirjalili et al., 2014b). SVM classifier is trained for improvement of classification applying Grey Wolf Optimizer algorithm by Eswaramoorthy et al. (2016). However, SVM has challenges like high algorithmic complexity, choosing a kernel function is not so easy and long training time for large dataset. Due to the mentioned challenges, this study concentrates on neural network and it training.

Although various heuristic algorithms can be used to train the neural network, the No Free Lunch theorem (NFL) indicates that there is no single meta-heuristic algorithm is the best suited to solve all optimization problems (Yang, 2012) (i.e. to tune the neutral network). For this reason, the investigation of suitable training of neural network is still deemed necessary. One of the very good approaches for classification is through evolutionary or nature-inspired algorithms which originate from the meta-heuristic search algorithms family (i.e. motivated by the theories and biological evolution and the actions of swarms of nature's creation). Grey Wolf Optimizer (GWO) is one of the recent meta-heuristic algorithms that has demonstrated potential for training neural network and the algorithm can be fine-tuned to perform even better (Faris et al, 2018, Mirjalili et al., 2014a).

The nature provides vast natural wonders with distinct behaviors of animal species. Hence, those unique behaviors and harmonious living of animals can be applied as great inspirations to solve various optimization problems. In this regard, grey wolf optimizer has demonstrated great potential as the algorithm is simple, flexible, derivation-free and able to avoid local optima. Due to the unique intelligence, GWO algorithm has been modified and applied to solve wide variety of optimization problems compared to other swarm intelligence approach (Faris et al., 2018; Mirjalili et al., 2014b; Nur & Ülker, 2018; Turabieh, 2016). Some of the successful applications of GWO algorithm to train the neural network include cloud-based intrusion detection and response based system (Nur & Ülker, 2018), prediction of heart disease (Turabieh, 2016), melanoma detection (Parsian et al., 2017), design static var compensator controller (Mohamed et al., 2015), and classification of sonar data set (Mosavi et al., 2016), just to mention a few.

Despite its reported performance, GWO is not without limitations. Specifically, the current best optimal individual is biased toward alpha and other individuals (e.g. beta and delta) attempt to modify their positions toward this best individual in each iteration process. This update process may cause the algorithm to fall to local optima especially in the cases where there are many competing local optima (Faris et al, 2018; Mirjalili et al., 2014b; Nur & Ülker, 2018; Turabieh, 2016; Mohamed et al., 2015; Mosavi et al., 2016). Hence, the proposed research is an attempt to modify GWO to addresses the deficiency of GWO for improvement of exploration by strengthen the searching process via several random leaders in each iteration, re-generating the random leaders in each iteration and introducing archive to verify the solution with better probability to proceed further for training and re-generation. The verification of each solution individually by Modified

GWO, instead of considering as a final solution, facilitates the improvement of the exploration. Moreover, the feature selection restricts number of variables to enhance the performance of the algorithm. With the mentioned approach, the research is an inspiration from the intelligence of Modified GWO for prevailing optimization algorithm.

1.4 Research Question

The research questions for this research are:

Question 1: Can the Grey Wolf Optimizer be enhanced (as Modified GWO) to improve its exploration and exploitation capabilities?

Question 2: Can the ensemble model incorporating MGWO be effectively developed for prediction analysis?

Question 3: Can the developed ensemble model perform optimally in comparison with existing strategies?

1.5 Aim and Objectives of the Research

The aim of this research is to enhance the GWO algorithm and address its limitation as far as exploration and exploitation capabilities.

The main objectives of the research are:

- To develop a modified GWO algorithm with random selection of leaders
- To adopt the modified GWO algorithm for training of neural network as ensemble model with stock market prediction analysis as case study
- To evaluate the performance of ensemble model against existing strategies in terms of the other developed optimization model in literature

1.6 Scope of the Research

The Grey Wolf Optimizer algorithm models the wolves' navigational ingenuity and implement it to solve optimization problems. The current research work focuses on ensemble intelligent prediction model consisting clustering data mining combined with classification algorithm and neural network that is capable of solving non- linear problems which can predict stock price trend with significant accuracy using historical stock market prices from the stock market.

The scope of the research is limited to the implementation of MGWO for classification, learning and feature selection. The research will take MGWO algorithm as the core implementation. The research adopts Multi Layer Perceptron (MLP) neural network trained with MGWO for stock prediction.

1.7 Significance of the Study

The current research will contribute to the common body of knowledge and research in the boundary of Swarm intelligence to take care of optimization issue in industries, engineering and other genuine issues pertaining to real-life. Moreover, the study intends to build up the Gray Wolf Optimization that will be productive and powerful through persistent exploration and exploitation of the search space. A study of neural network model's efficiency in the selection of models for practical use of stock prediction is another significance of this study. An ensemble of neural network and MGWO is proposed in this research. GWO is modified for feature selection, classification and learning by maintaining an archive to select the best solution that provides better probability to proceed further for training and re-generation. This algorithm will be a supervised method where class information needs to be supplied. The algorithms are also be tested with a benchmark data set.

Neural network performance is enhanced in this research by training it using MGWO to alleviate the problem of over-fitting, entrapment in local minima, result inaccuracy, slow convergence rate. In the proposed model, a clustering model is applied to the training dataset followed by a neural network model. This model is regarded as an ensemble model since it combines the neural network and MGWO one after another. A prototype of the ensemble model is implemented to demonstrate its practical use for stock prediction here.

1.8 Research Framework

This section illustrates the complete research activities to attain the research objectives. Precisely, the section illuminates the stages of research development, design, and evaluation of the proposed model as indicated in Figure 1.1. The research framework is divided into three stages specifically, Literature Review, Research Methodology, and Evaluation.

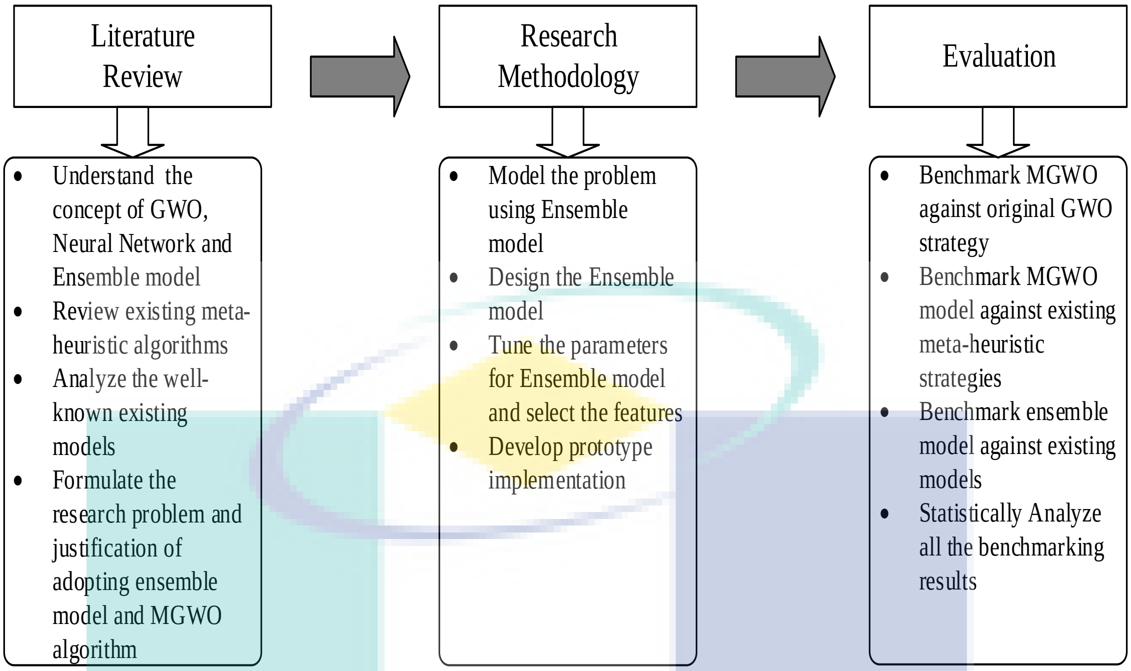

Literature Review
Research Methodology
Evaluation
Understand the concept of GWO, Neural Network and Ensemble model
Review existing meta-heuristic algorithms
Analyze the well-known existing models
Formulate the research problem and justification of adopting ensemble model and MGWO algorithm
Model the problem using Ensemble model
Design the Ensemble model
Tune the parameters for Ensemble model and select the features
Develop prototype implementation
Benchmark MGWO against original GWO strategy
Benchmark MGWO model against existing meta-heuristic strategies
Benchmark ensemble model against existing models
Statistically Analyze all the benchmarking results

Figure 1.1 Research Framework

1.8.1 Literature Review

Literature Review stage involves reviewing the literature by critically comparing the existing work and analyzing theirs strengths and limitations in order to justify for the adoption of ensemble model for stock prediction. Moreover, the requirement of the research, theoretical background of GWO and ensemble model design definitions are established. At this stage, the problem statement is identified and formulated based on the review of existing works.

1.8.2 Research Activity

Research Activity stage involves finding the best model for stock prediction and adoption of ensemble model is established. The ensemble model is implemented at this stage applying MGWO and neural network to achieve the best performance. Then, complete algorithm to construct the ensemble model is designed and developed. Additionally, some related concepts such as feature selection, classification and learning through the application of MGWO are also demonstrated at this stage.

1.8.3 Evaluation

Evaluation stage involves the evaluation of ensemble model. First, MGWO strategy is compared against GWO to evaluate the efficiency of introducing Archive with GWO. Then, MGWO is compared with other meta-heuristic strategies. Next, ensemble model is compared with existing strategies. Finally, the statistical analysis of benchmarking results are performed.

In essence, the research attempts to address three objectives to achieve the aim of the research which is investigation of developing an ensemble model applying MGWO and neural network for stock prediction. Each objective is mapped with several activities need to be conducted in order to achieve all objectives.

1.9 Thesis Organization

This thesis is comprised of five chapters. The remaining chapters are organized as:

Chapter 2 presents the review of the relevant literature, meta-heuristic algorithms, financial prediction, the data exploration stages applying data mining, neural network and several state-of-the-art models including their drawbacks are included in this chapter. It also comprises the related literature on the building of predictive models from stock data and the review of techniques used in this study. Finally, performance improvement of neural network model is reviewed.

Chapter 3 elaborates the research methodology that specifies the details of how the research should be conducted in order to fulfill the research objectives. The chapter also discussed the research approach of the study and the design strategy implemented to carry out the research question. It contains neural network concepts and the different configurations to be investigated. The training algorithm GWO is described. Combinations of parameters and choices of the various associated values, as well as the results of parameter tuning are also provided. It includes the ensemble of neural network and MGWO as well.

Chapter 4 focuses on ensemble model to combine, neural network, classification and MLP neural network are included in this chapter. It also provides the experimentation results, discussion and summary of the stock prediction. Here, comparisons are made with some state-of-the-art models that are well reported for the exploration of data for prediction purpose. The chapter also illustrates the evaluation of the prediction and the computation for measuring the accuracy of a numeric prediction using Mean Absolute Error and statistical test.

Finally, Chapter 5 attempts to draw the conclusion of the thesis. The chapter contains the research findings related to research objectives and the discussion of the significant contributions of this research to knowledge. At the end of this chapter, the chapter includes the recommendation for future research works that can be conducted to improve and fortify the outcome of this research.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Chapter 1 includes the brief background and issues related to optimization, neural network and learning, classification. Constructing from the preceding chapter, the purpose of this chapter is to provide the critical review of relevant literature to reveal the gaps in literature so that the current research turn into the complement. Initially, the study inspects the concept of optimization and determines the optimization algorithms development including both the stochastic and deterministic algorithms. Moreover, various types of optimization algorithms and their applications, strengths and weaknesses are analyzed in this research. Additionally, the study highlights the detail background concepts of the research work, the Grey Wolf Optimizer (GWO), the neural network and its training, the application of meta-heuristic algorithms for training neural network. The chapter also analyses literature that is relevant to ensemble approach, strength of GWO, limitation and enhancement of GWO algorithm. Finally, the chapter provides a critical gap analysis in order to justify the current proposed work.

2.2 Optimization

Industrial and technological advancements are greatly stimulated through optimization all over the world. Optimization strives for better productivity in business, engineering and manufacturing. Optimization searches for the ideal methods to accomplish an end amidst a few means (Faludi, 2013). Fundamentally, optimization includes the maximization or minimization of a function by methodically picking some input values within reasonable set so as to compute the value of the function with the point of deciding the best estimations of the objective function. The general goal of optimization is to guarantee more noteworthy effectiveness utilizing less resources, for example, a computer program could be optimized to utilize less memory, execute quicker or use minimum resources.

Overall, the objective of optimization technique is to guarantee ideal utilization of accessible resources. However, such facility requires significant cost, for example, a computer program may run quicker and acquire more adequacy, most likely because of its utilization of more memory and the other way around. In general, in this manner, designing of an algorithm is required so that well trade-off between different constraints of an optimization process may be guaranteed. The next section will shed the light on some of the optimization algorithms that represent the state of the arts.

2.3 Optimization Algorithms

The requirement for optimization has stimulated the advancement of correct algorithms, prevalently called deterministic or traditional algorithms, for example, finite volume methods (Said and Wegman, 2009), Linear Programming (LP) (Kuhn, 2014), Newton-Raphson (Wooldridge, 2010), Dynamic Programming (Sniedovich, 2010), finite elements (Hughes, 2012).

Probabilistic or random elements are not utilized for the proper functioning of Deterministic algorithms (Motwani and Raghavan, 2010). Thus, these algorithms yield a similar output values for a given input values and the back-end machines probably could utilize a similar succession of states. In contrast, the stochastic algorithms utilize built-in randomness where, distinctive outcome may be produced by the algorithms for a given set of input values and initial conditions (Gentle, 2013; Machairas et al., 2014). Regardless of this, stochastic models have demonstrated to be very fruitful for comparatively bigger problems consist of numerous input parameters and operating conditions. Alongside, stochastic algorithms have also been implemented recently to establish latest algorithms consist of harmonious and self-organized elements in nature, which is categorized as Natural Computing (Păun, 2012).

There are algorithms that basically utilize the computer to generate concepts from nature to create computational frameworks or utilize natural materials, for example, molecules to carry out calculation, are identified as Natural Computing. Hence, the definition of Natural Computing includes that nature is the motivation for such computing, at times termed as Nature-Inspired Computing (NIC) or Computing with Natural Materials (CWN) (Dodig-Crnkovic, 2012). Natural materials based computing is the latest achievement in computing approaches where developers make utilization of natural media as a replacement of silicon for computational instruments such as, hardware and software (Zang et al., 2010).

Many researchers are enthusiastic about NIC algorithms progressively due to well acceptance in the previous couple of decades in scientific and engineering research everywhere throughout the world. The essential reason given for this prominence is that these algorithms are created to produce the best elements in biological, chemical and physical procedures in nature (Rozenberg et al., 2011). This circumstance hurls the issue of determining appropriate algorithm as presently a few algorithms are available at whatever point a researcher needs to solve an optimization problem. The decision of a specific algorithm is reliant on its ability to tackle the current problem. This concept is strengthened in optimization by the No free-lunch theorems (X.- S. Yang, 2011).

Generally, optimization algorithms have the organization as:

$$\textit{Minimize } f_i(x)\ (i = 1,2,3,\ldots, M), \qquad x \in \Re n \qquad 2.1$$

$$\textit{subject to } h_a(x) = 0, \qquad (a = 1,2,3, \ldots., N), \qquad 2.2$$

$$g_b(x) \leq 0, \qquad (b = 1,2,3, \ldots, K) \qquad 2.3$$

Where, $f_i(x)$, $h_a(x)$, $g_b(x)$ are functions of the design vectors.

$$xiL \leq xi \leq xiU \qquad i = 1, N \qquad 2.4$$

In this occurrence, the function $f_i(x)$ where, i = 1, 2, … ., M is known as the objective function. The objective function could be defined as a maximization or minimization problem. For a situation where, M = 1, at that point it is an instance of single objective function and for $M \geq 2$, it is a multi-objective function. Also, the variable $x(i)$ of x is called decision or design variable which could be continuous, discrete or a blend of both (Feist and Palsson, 2010). The space secured by the decision variable is known as the search space ϵ ℜn. Similarly, the space secured by the objective function is known as the solution space, while h_a and g_b are the equality and inequality constraints individually. For the inequality constraints, the maximization has the form ≥ 0, in contrast, the minimization has the form ≤ 0.

In addition, the side constraints are the searchable design space that is characterized by the upper and lower limits, *xiL* and *xiU*, of the design or decision variables. Generally, the objective, goal or cost function can be defined to be linear or nonlinear, implicit or explicit. Integer or discrete optimization problems consist of decision variables with discrete or integer values. Most occasions, conventional optimization technics experience a considerable measure of challenges tackling discrete or integer optimization problems. This is typically the region of solidarity of the stochastic algorithms (Venter, 2010).

2.3.1 Traditional Algorithms

Traditional optimization algorithms are typically deterministic in nature and utilize the gradient-based approach (Davoodi et al., 2014). The example of such algorithms includes, the Simplex Method and Newton-Raphson. The traditional optimization algorithms are exceptionally powerful in smooth mono-modal problems because of utilizing functional values and their corresponding derivatives to determine the appropriate result. But, the algorithms may devise the disturbances to the objective function in some situations which deviate the researchers to select non-gradient methods that utilize Hooke-Jeeves pattern search and Nelder-Mead downhill simplex functional values (Haftka & Gürdal, 2012).

Substantial number of decision variables is well handled by the traditional optimization algorithms that need limited problem-specific parameter tuning. Moreover, those algorithms are typically ready to get the optimum solution in mono-modal environments. But, traditional optimization algorithms include complex optimization strategies and hence they are not suitable for multimodal search environment. Additionally, the algorithms experience complexities in taking care of discrete optimization problems and are not so strong in dealing the circumstances like numerical noise (Toga et al., 2012). The stochastic algorithms are required to be developed due to the mentioned shortcomings, which will be discussed further in the following section.

2.4 Stochastic Algorithms

Specifically, two sorts of stochastic algorithms are available such as, Nature-Inspired Computing (NIC) and Computing With Nature (CWN). The algorithms can create broad utilization of randomness in searching for optimization (Dodig-Crnkovic, 2012).

2.4.1 Nature-Inspired Computing

Nature Inspired Computing (NIC) are motivated by the harmonious co-existence and the complex problem-solving techniques of natural environments (Kefi et al., 2015). Consequently, various scientific investigations are motivated by NIC and such investigations are: neural networks (Mäkisara et al., 2014), cellular automata (Codd, 2014), artificial immune systems (Hemamalini and Simon, 2011), evolutionary computation (Thiele et al., 2009) and swarm intelligence (Ducatelle et al., 2010). Likewise, robotics researchers are also motivated by nature and proposed mechanical artificial intelligence discipline to develop water strider robot, self-configuring robots, robotic salamander and mechanical cockroaches (Dewangan et al., 2014). Biologically inspired algorithms are another subset of NIC that can produce the incredible solutions for complex optimization problems through creation of the collective intelligence with a group of biological agents (Pandiri and Singh, 2015). The motivation and development of NIC includes the field of biology, chemistry, physics and engineering. Commonly, NIC systems contain the simulation of harmonious self-organization, interaction, competition and interdependence of natural elements of the ecosystem. Overall, NIC has been found to acquire answers for issues utilizing heuristics or meta-heuristic standards and this has empowered them to be truly versatile, adaptable and hearty to the degree that they can be implemented to an extensive variety of utilizations with exceptionally competitive results (Fister et al., 2013).

2.4.2 Computing with Nature (CWN)

Computing with Nature (CWN) transformed computing through the utilization of natural materials replacing silicon. The applications of RNA, DNA and quantum computing are some of the examples of CWN based computational processing. Moreover, CWN based computing are also applied in recent times to molecular or bio-computing, bio-chemical computing, bio-molecular computing or DNA computing that utilizes components from molecular biology for data processing operations such as, logical, arithmetic and other computer operations (Rozenberg et al., 2011). CWN based molecular computing has also been implemented effectively to take care of a 7-vertice TSP issue by only exploring different avenues regarding DNA strands in a test-tube, 20-variable 3SAT issues, cryptography, sticker frameworks, joining frameworks and the structure applications for savvy drugs (de Castro, 2007).

Alternatively, quantum computing executes computations through the consideration of data as quantum bits and involving mechanical means, for example, entrapments and super-positioning. A quantum bit, which is also referred as qubit, contains a '0', '1' or a quantum superposition of either a '0' or '1'. Additionally, the quantum computer utilizes logic gates to carry out computing operations on the qubits with the guide of Shor's polynomial algorithm for integers factoring and the Grover's algorithm for quantum database query (Hirvensalo, 2013). Although, quantum computing is still at its earlier stages of advancement, researchers are enthusiastic to investigate the true ability of this computing paradigm as quantum computing has demonstrated its potential to quantum cryptography, quantum teleportation, nuclear magnetic resonance imaging, pattern recognition and classification (Hirvensalo, 2013).

2.4.3 Heuristic and Meta-heuristic Algorithms

NIC utilizes heuristic and meta-heuristics algorithm extensively to enrich computation. A complex problem can be solved by heuristic algorithm through the exploitation of some information. Although, heuristic algorithms are near-exact algorithms that may not produce exact optimal solution, the utilization of heuristic algorithms yet can produce excellent outcome for complex optimization problems at an appropriate time (Safari, 2015). Alternatively, meta-heuristics algorithms, which are also termed as ‘beyond heuristics’, can act superior to heuristics utilizing intelligent memory; experiential and different biases to assist manage the search process (Prakasam and Savarimuthu, 2015). Generally, meta-heuristic algorithms apply local search besides global explorations utilizing randomizations, which assist these algorithms with steering far from being limited in a local environment to a progressively global search. The general goal of any meta-heuristic algorithm is to accomplish the most ideal outcome by utilizing typical mechanisms to accomplish satisfactory exploration and exploitation of the search space (Blum and Roli, 2003). Extensively, meta-heuristics algorithms can be applied to wide range of areas such as bioinformatics, telecommunications, economics, manufacturing and so on. (Osman and Kelly, 2012).

Meta-heuristic algorithms can be commonly categorized into two types namely, population-based and trajectory-based (Behesti and Shamsuddin, 2013). Population based metaheurisctic algorithms can be recognizable to Holland who published his work in 1962 and whose works utilized a mix of theoretical genetics and automata approach. Researchers were inspired to meta-heuristic algorithms because of applying variety and diversification strategies to a population to accomplish results inside a search space. A portion of these approaches can be mentioned as: Schaffer's Vector-Evaluated Genetic Algorithm (VEGA) (Pierre et al., 2011); Farmer, Packard and Pearson's Artificial Immune Systems (Farmer et al., 1986); Holland's and Rosenberg's Evolutionary Strategies (Cuomo et al., 2012); Dorigo and Di Caro's Ant Colony Optimization ACO (Di Caro and Dorigo, 1998) and Grey Wolf Optimizer (Mirjalili et al., 2014a).

2.4.4 Characteristics of Meta-heuristic Algorithms

A decent meta-heuristic algorithm contains two vital features such as capabilities to engage global search mechanism or exploration and local search or exploitation (Osaba et al., 2016). Where, ‘Local Search’ can explore the capable neighboring regions in the hope to determine the optimal solution that is termed as exploitation and ‘Global Search’ facilitates to skip any local optimum that is also referred as exploration. The efficiency of meta-heuristic algorithm may be substantially adjusted by balancing the interaction between local search or exploitation and global search or exploration. However, searching locally a lot may lead the algorithm to be trapped in local optimum, on the other hand, an aggressive global searching may result inefficiency that affects the whole performance of the search (Yang et al., 2014).

Another key component of a decent algorithm is the capacity to recognize the best outcome in iteration and conceivably the best structure vector related with such best outcome. Generally, the rule is identified as ‘The survival of the fittest’. The criterion can be accomplished by continuously refreshing the present best found up until this point (Yang, 2011).

2.4.5 Randomization in Meta-heuristic

Meta-heuristic algorithms contain three key components namely, exploration, exploitation and determining the top performer where, each algorithm is differentiated from one another based on the mechanism engaged to attain the mentioned key components (Li et al., 2010). The incorporation of randomization and a deterministic procedure facilitates the met-aheuristic algorithms to attain the goal where, randomization is a mechanism to define the upper and lower boundaries in a uniformly distributed variable ranging from 0 to 1. Particle Swarm Optimization (PSO) and Firefly Algorithm (FA) have adopted this approach. Cuckoo Search is another strategy used by meta-heuristic that adopts Lévy flight, which is a random process, categorized by step-jumps to look out the disorganised dust particles movement in a fluid (Senthilnath et al., 2012). However, exploration is attained by engaging crossover and mutation for some algorithms such as Genetic Algorithm (GA), Genetic Programming (GP) and Evolutionary Programming (EP). Where, mutation determines the latest solution from initial population and crossover emphasizes the limit on over-exploration (Rani et al., 2012). The exploitation is achieved for the meta-heuristic algorithms by producing different solutions from initial solution. But, the exploitation is attained for Simulated Annealing (SA) algorithm through engaging random walk (Kirkpatrick et al., 1983) and for Harmony Search (HS) algorithm by pitch alteration (Mahdavi et al., 2007). The equation 2.5 denotes the mentioned approach:

$$Xnew = Xold + sw \tag{2.5}$$

Where, *Xnew* is the new solution, *Xold* is the initial solution, *s* is the step size and *w* is zero mean determined from Gaussian distribution. However, the step size should not be too narrow or too wide because too wide step size will support exploration eliminating exploitation and too narrow step size will support exploitation to produce the result trapped into local minima. Therefore, the algorithms should engage Lévy flight or random walk so that the appropriate step size can be determined from Lévy distribution (Kennedy, 2010).

2.5 Broad Classification of Meta-heuristic Algorithms

In Literature, meta-heuristic algorithms are classified in numerous ways and one such way is population-based or trajectory-based. Determining the solution of a problem implementing the population of solutions at a period is known as population-based algorithm (Wong & Moin, 2015). Population-based algorithm determines the initial solution randomly and then iteratively improves the solution. The algorithms are also known as exploration-based approach due to the algorithms excellent ability of the search space diversification. GA and PSO are the two perfect examples of the mentioned approach where, GA utilizes a set of strings and PSO utilizes a number of particles (Kennedy, 2010). Alternatively, trajectory-based algorithm utilizes a single agent to revolve around the search space in zigzag manner iteratively and the example of such approaches includes Simulated Annealing, Great Deluge and Hill Climbing (Kennedy, 2010; Mirjalili et al., 2014b). Population-based and trajectory-based algorithms are different from each other based on number of temporal solutions during each search iteration course where, population-based algorithms utilize multiple agents to produce multiple solutions but trajectory-based algorithms utilize a single agent to produce single solution.

2.5.1 Trajectory-based Algorithms

Trajectory-based algorithms which is also referred as exploitation-based algorithms initially determine a single solution for the current search and then the solution is improved iteratively to produce the final solution (Park et al., 2013). The algorithms generally emphasize the intensification where, the optimal solution is determined by the search agent that moving through the search space to trace the path in the search landscape (Manjarres et al., 2013). Figure 2.1 demonstrates the pseudocode for Trajectory-based algorithms where, the search agents move randomly one solution to another continuously till the stopping criteria is met in a solution space.

```
Begin
    Generate initial solution ($s(0)$)
    $t = 0$
    While (not Termination ($s(t)$))
        Explore neighborhood $s'(t)$ = SelectMove($s(t)$)
        If Move($s'(t)$) accepted
            $s(t)$ = ApplyMove ($s'(t)$)
            $t = t + 1$
        End If
    End While
    Output best result
End
```

Figure 2.1 Trajectory-based Algorithms Pseudocode

The productivity of Trajectory-based algorithm with regard to time and quality of solutions can be enhanced by incorporating parallelism, where parallel multi-start, parallel evaluation and parallel moves are three well-known parallel models (Alba et al., 2005). In literature, various Trajectory-based algorithms are available such as Simulated Annealing, Hill Climbing and Great Deluge.

2.5.1.1 Simulated Annealing

Kirkpatrick et al. established the idea of Simulated Annealing (SA) algorithm to model the cooling and heating processes of materials in metallurgical engineering. The idea of the SA algorithm comprises that the metals become too strong by gradually reducing the temperature so that the system energy can be minimized through the cooling process. The algorithm initiates a random search at high temperature for the cooling process to produce greedy decent till the temperature turns to zero. The algorithm performs well in lower temperature compared to higher because the randomization feature facilitated by the SA algorithm ensures avoiding local optima because the greedy descent may place the algorithm stack into local minima (Downsland & Thompson, 2012).

During each move, SA algorithms search for optimal solutions through the implementation of random variables so that the improvement of the objective function can be determined where; lower objective value is preferable for a minimization problem. Hence, SA algorithms improve the objective function through the avoidance of being trapped to local minima so that global exploration can be maintained. As the algorithms progress, the Annealing schedule consisting both linear and geometric feature ensures the effectiveness through reducing the temperature. Moreover, SA algorithm maintains the search area minimization and earlier convergence features through the reduction of the temperature. Figure 2.2 indicates the pseudocode of SA algorithm that can be implemented for problem minimization (Li, X.-G., & Wei, 2008).

```
Begin
    Initialize population and parameters
    Generate randomly an initial optimal state $S_i$,
    Calculate $f(S_i)$
    Select an initial temperature $T_0$
    Select a terminal temperature $T_f$ or a total number of temperature chant $t_{max}$
    Set temperature change counter $t = 1$
    While $T_i < T_f$ or $t = t_{max}$
        Set repetition counter $L = 0$
        Repeat Until $L = L_t = \beta t$
        Generate new state $S_j$, a neighbor of $S_i$
        Calculate $\Delta E = f(S_j) - f(S_i)$
        If $\Delta E < 0$ then
            $S_i = S_j$
        Else If random $(0,1) < Exp \left(-\frac{\Delta E}{K_b T}\right)$, then
            $S_i = S_j$ where $K_b$ is Bolzmann's constant
        End If
            $L = L + 1$
        End Repeat
            $t = t + 1$
            $T_i = \alpha T_i$, where α is the cooling rate
    End While
End
```

Figure 2.2 SA Algorithms Pseudocode

SA algorithm can be implemented to solve various problems for example, Artificial Neural Networks Training (Ledesma et al., 2008), Quadratic Assignment Problem (Bilbao & Alba, 2009), Job Shop Scheduling (Van et al., 1992), Traveling Salesman's Problems (Malek et al., 1989), N-Queens Problem (Tambouratzis, 1997). SA algorithm is having the capability to avoid local minima through the implicit manipulation of the temperature cooling which is the main strength of the algorithm. But, SA algorithm may not be very efficient to implement for smooth energy landscape and also for the problems with few local minima. Additionally, SA algorithm may not reach for appropriate outcome within certain period of time because of including many cost function evaluations iteratively (Kumbharana & Pandey, 2013).

2.5.1.2 Hill Climbing Algorithm

Hill Climbing (HC) algorithms iteratively determine the solution for a problem by picking an arbitrary solution initially and modify the single solution to determine better solution (Hoffmann, 2010). The modification will continue to determine a new solution until no further improvement is possible. But an extra modification on HC algorithm's effort will be attempted to determine the optimal solution if, any modification leads poor solution.

There are numerous ways by which HC algorithm is different from similar algorithms like Gradient Descent such as, HC algorithms fine-tune only a single value for the current solution but Gradient Descent modifies multiple values for current solution for the subsequent iteration. Hence, HC algorithms are considered as a type of Depth-First search (Fisher, 1987). However, HC algorithms apply feedback mechanism to estimate the closeness or latest solution so that the next search direction can be determined, which is different from Depth-First search that rejects or accepts the solution out-rightly. The pseudocode for HC algorithm is demonstrated in Figure 2.3 that can be applied for the minimization problem. HC algorithms can be implemented to various arena to obtain effective outcome such as configuring application servers (Xi et al., 2004), Traelling Salesman’s Problem (Selman & Gomes, 2006), signature verification (Galbally et al., 2007).

```
Begin
    Initialize population and parameters
    $x_k = LB_k + (LB_k - LB_k)$ x $U_{(0,1)}$ , $k = 1,2, \dots N$ ( The initial state of $x$)
    Calculate $f(x)$
    While (until termination condition)
        $x' = improved\ (x)$
        If $f(x') \leq f(x)$ then
            $x = x'$
        End If
    End While
End
```

Figure 2.3 Hill Climbing Algorithms Pseudocode

HC algorithm is useful compared to other similar algorithms due to lower consumption of computer resources for searching as the algorithms store only current solution. In addition, the algorithms produce better results in comparison to other algorithms for the unexpected interruption during the execution of the algorithm. But, HC algorithms may have some limitations such as low speed for ridges instance, alleys and plateau, probability of getting stuck into local minima (Minton et al., 1992). Although, the issue of slower speed and getting stuck into local minima is the main concern for many trajectory algorithms and hence many researchers successfully investigated to minimize the issue (Sharma et al., 2016).

2.5.1.3 The Great Deluge

The Great Deluge (GD) algorithm is proposed by Dueck that includes the concept of a person's activities to move in various directions upwards to the hill during a deluge so that his feet can be avoided to become wet if the water level rises (Özcan et al., 2012). GD algorithms initially assign a value similar to the initial objective function for the parameter and the value is decreased iteratively during the progression of the search. The algorithms produce the final solution if the determined value is nearly equivalent to the objective function.

The GD algorithms have been advanced later by allowing the algorithms to receive all downhill moves and also hybridizing GD with Hill Climbing for better effectiveness (Burke & Bykov, 2017). The GD algorithms can be applied by choosing an optimum solution from an approximate solution J. Later, the algorithms select K as a random value of *badness* so that the desired approximate solution can be estimated. This way, J will produce adverse solution for greater assessment of the *badness* value. The algorithms implement another parameter called *tolerance* that can assess numerous factors to select J' as an approximate solution for a neighbor J. The calculation for J' solution's *badness* is determined at this phase to compare the outcome with *tolerance* parameter. GD algorithms initiate recursively for any outcome better than *tolerance*. But, any outcome worse than *tolerance* will result to choose J'' as a neighboring solution for J that will allow the process to be continued till better results than *tolerance* are determined for all neighbors of J. Finally, GD algorithm will be concluded with a final solution J (Dhouib, 2010). Figure 2.4 demonstrates the pseudocode for GD algorithm that can be applied for minimizing a problem (Nabeel, 2010; Othman et al., 2013).

```
Begin
    Initialize population and parameters
    Generate random solution $Sol$
    Set $SolutG = Sol$
    Set $Level = f(SolutG)$
    Set estimated quality of Solution: $Estimated.Quality$
    Set number of iteration: $IterNo$
    Calculate increase rate $\beta = \frac{Estimated.Quality}{IterNo}$
    Set Iteration = 0
    While ($iteration < IterNo$)
        Generate a random new Solution $Sol_{new}$ in the neighborhood of $Sol$
        Calculate $f(Sol_{new})$
        If $f(Sol_{new}) > f(SolutG)$ then
            $Sol_{new} = SolutG$
            $Sol = Sol_{new}$
            $f(Sol) = f(Sol_{new})$
            $f(SolG) = f(Sol_{new})$
        End If
        If $f(Sol_{new})$ >Level then
            $Sol = Sol_{new}$
            $f(Sol) = f(Sol_{new})$
        End If
        If $f(Sol_{new}) \leq Level$ then
            $Level = Level + \beta$
        End If
        $Iteration = Iteration + 1$
    End While
    Return $SolutOptimalGD$
End
```

Figure 2.4 Great Deluge Algorithms Pseudocode

GD algorithms can be applied to various extents such as prediction for protein structure (Burke et al., 2007), problems of facility layout (Nahas et al., 2010), issues of patient admission (Kifah & Abdullah, 2015), course timetabling, sports, examination and other similar areas. The algorithms are unlike Hill-Climbing and Simulated Annealing due to receive neighborhood candidate solution. GD algorithms are more effective compared to HC and SA as the algorithms allow to explain two characteristics earlier for a search process such as processing time and processing region of the estimated solution (Burke et al., 2003). However, the algorithms are having limitation such as possibility of being trapped to local minima that creates the variations of the algorithm (Mcmullan, 2007). In spite of GD algorithms advancement with variations, the efficiency of the algorithms is yet a main concern.

Overall, various limitations of trajectory algorithms such as possibility of getting trapped to local minima, slower speed due to accept single solution at a time and inefficient for mixed-objective optimization to maximize or minimize objectives (Yang et al., 2006) motivates the researchers to opt for population-based approaches which accept multiple-optimal solutions at one iteration due to the application of multiple search agents.

2.5.2 Population-based Meta-heuristic Algorithms

Population-based approaches usually implement a set of decision vectors, which can be expressed by Equation 2.6 where, N denotes the size of population and the number of design variables is denoted by n (Kothari, 2012).

$$X = \{x_{1\,,}\ x_{2,}x_{3\,,}\ \dots\ x_{N}\ \} \qquad 2.6$$

Where, $$Xi = (x_i 1, x_i 2, \dots, x_i \mathrm{n}) \qquad 2.7$$

Meta-heuristic term was first introduced and applied by Glover (1986), who has also proposed an algorithm called TS (1989). Escaping from local optima is the main endeavor of Meta-heuristic algorithm that can explore the search space proficiently by trial and error basis. The majority of meta-heuristic algorithms are considered as population based algorithms, where finding a result begins from numerous places of solution space which is different from traditional algorithms. Consequently, every individual of the population may be the candidate to determine the optimal solution. Meta-heuristic algorithms can guide the searching and movement in the search space which can facilitate to proficiently explore the complete search space and the same may not be possible by other search algorithms. The guidance for search is problem specific, which can be referred as fitness function, where parameters can be maximized or minimized through the fitness function in light of issue nearby.

Population-based meta-heuristics algorithms generally share a similar structure consist of four components such as, main algorithm, extension to deal with constrained optimization problems, extension to retain promising solutions and a component to halt the program. However, the key algorithms implement three features namely, crossover, mutation and selection for most cases (Nozohour-leilabady & Fazelabdolabadi, 2016). Figure 2.5 represents the Pseudocode for Population-based Meta-heuristic algorithms (Karaboga & Bastruk, 2007).

```
Begin
    Initialize population and parameters
    Evaluate the objective function
    While (until termination condition)
        Evaluate the population quality
        Apply the variation operator
        Evaluate the objective function
    End While
    Output best result
End
```

Figure 2.5	Population-based Algorithms Pseudocode

In the literature, many useful meta-heuristic algorithms have been proposed over the last few decades. Some of the most popular meta-heuristic algorithms implemented, but not limited to, are: PSO (Cura, 2009, Seidy et al., 2016 and Li et al., 2014), GA (Delnavaz, 2014 and Razali et al., 2011), ACO (Mohapatra et al., 2013 and Yang et al., 2014), ES (Wen et al., 2015, Bliss et al., 2014 and Bisoi et al., 2014), PBIL (Ali et al., 2014), BBO (Mirjalili et al., 2014a) and GWO (Mirjalili et al., 2014b).

‘Local Search’ and ‘Global Search’ are the two key components of meta-heuristic algorithm. Where, ‘Local Search’ can explore the capable neighboring regions in the hope to determine the optimal solution that is termed as exploitation and ‘Global Search’ facilitates to skip any local optimum that is also referred as exploration. The efficiency of meta-heuristic algorithm may be substantially adjusted by balancing the interaction between local search or exploitation and global search or exploration. However, searching locally a lot may lead the algorithm to be trapped in local optimum, on the other hand, an aggressive global searching may result inefficiency that affects the whole performance of the search (Yang et al., 2014).

Population-based meta-heuristic algorithms usually exploit the prior knowledge from the solution space and the search agent is moved towards the feasible region by utilizing the solution at the initialization phase. If the necessary information is unavailable, the distribution of the decision vectors is taken place uniformly at the search space (Wong & Moin, 2015). The algorithms are generally inspired by harmonious co-existence of nature such as bio-inspired and swarm-based however, some algorithms like Grey Wolf Optimizer, Biogeography-based Optimization, Black Hole Optimization, Harmony Search are inspired by physics, chemistry or geography.

2.6 Swarm Intelligence

Group of researchers investigated and specified a new discipline named Swarm Intelligence that consists of a simple mobile agents set to solve essential issues collectively by direct or indirect communications (Binitha & Sathya, 2012; Kennedy et al., 2001; Mahale & Chavan, 2012). Swarm Intelligence utilizes basic rules consisting emergence of intelligent behavior for secret single agent to treat a group of individual natural and artificial systems for organizing by decentralization. Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) and Honey-Bee Mating Optimization are some of the commonly used swarm intelligence algorithms. As Grey Wolf Optimizer (GWO) is one of the most recent meta-heuristics swarm intelligence approach, the algorithms of this type will be investigated further in detail.

2.6.1 Swarm-based Approaches

Swarm Intelligence consists of combined social interactions of creatures that implements group's cooperative intelligence for simple agents like ants, animals, plants and other elements of ecosystem depending on the real-life behavior (Pandiri & Singh, 2015). Swarm-based approaches comprise various features like:

- Swarm-based approaches are population-based and use multi agents for searching
- The population agents are homogeneous

- The outcomes of the system yield from individual interactions with each other in the environment
- The movement of the individual agents are mobile and chaotic
- The control structure is decentralized where; each iteration is performed by the action of individual leader (Parpinelli & Lopes, 2011).

Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Bee Colony Optimization (BCO) and Firefly Algorithm (FA) are some of the prominent and frequently applied Swarm-based approaches. The mentioned common Swarm-based approaches may be applied for stock prediction. Enormous amount of information is processed and stored every day in stock markets worldwide. However, it is not always possible to make an appropriate decision about the stock investment using this information. Sometimes, it may not be even possible to receive the desired return through the stock investment from this vast amount of information applying various predictive models. Stock market remains best investment alternatives for few decades despite being unpredictable and uncertain. Prediction of stock price is extremely complicated due to the nonlinear form of stock data. As the economic condition of a country greatly depend on stock market, researchers are investigating endlessly to determine the best predictive model for stock market. Prediction of stock market is significant in finance and is gaining more attention of the researchers, due to the fact that the investors may be better guided through successful prediction of the stock price. Exploring stock data needs to build a predictive or descriptive model such that hidden information lies in data can be unfolded. Consequently, building a predictive model from the stock data is a complicated task. Information from large databases can be extracted through a well-known technology called data mining that facilitates the organizations to retrieve the vital information from data repositories (Witten et al., 2016). From the existing models and recent developed algorithms, extracting the best subset of features that helps in accurately identifying the labeled action from stock market massive amount of data 2^N subsets is not an easy task and tends to be non-polynomial complex problem during the raise of searching space (Chandrashekar et al., 2014). Considering the limitations and prospects, research should be focused to develop an algorithm for stock prediction that is applicable not only for a single market but also generalized for various stock markets. However, applying machine learning model does not guarantee a good accuracy, besides the accuracy of machine learning models similarly with data mining as well as neural network may get affected by numerous factors (Negnevitsky, 2005). Moreover, the selection of input parameters may result inconsistent output.

2.6.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is one of the common choices for solving complex and intricate problems that cannot be solved by traditional methods. Stock prediction issue can be well addressed through PSO that facilitates maximizing profit and minimizing risk. The application of PSO is presented by Cura (2009) and Particle Swarm with Center of Mass (PSOCoM) to move the particles to the best predicted position is proposed by Seidy (2016) that can train the adaptive linear combiner to form a stock market prediction (Seidy, 2016).

PSO has been combined with other models to propose ensemble model for stock prediction in recent work (Khajavi et al., 2017, Seidy, 2016). Many optimization problems has been addressed through PSO. The natural process of swarm behaviors such as bird and fish swarm for searching food is mimicked by PSO. The local search by individual experience with the global search by neighboring experience can be balanced by PSO.

The pseudo code of PSO algorithms is indicated through Figure 2.6. PSO applies swarm (population) of particles (individuals), which can be moved to the search space over numerous iterations. Each particle indicates a candidate solution for the problem, which is also considered as a point in M-dimensional space. The status of the particle is portrayed by its position and velocity. PSO is accomplished by adjusting a swarm of random particles where particle flying along the direction that will be balanced through local best (position of one particle) and global best (ever found by all particles). The particle is updated in each iteration by two best values or fitness namely, *pBest* (local best) and *gBest* (global best).

```
Begin
While terminating condition is not reached do
    For $i$ = 1 to number of particles do
        Calculate the fitness function $f$
        Update personal best and global best of each particle
        Update velocity of the particle using Equation, $V(t+1) = wV(t) + c_1 \times rand\,( ) \times (X_{pbest} - X(t)) + c_2 \times rand\,( ) \times (X_{gbest} - X(t))$
        Update the position of the particle using Equation, $X(t+1) = X(t) + V(t+1)$
    End For
End While
End
```

Figure 2.6 PSO Algorithms Pseudocode

PSO has been applied for nonlinear function optimization, pharmaceutical and biomedical applications, communications and combinatorial optimization problems successfully (Poly, 2007). The critical assessment of PSO algorithms confirm that PSO algorithms are better than Ant Colony Optimization (ACO) as PSO algorithms are simple for implementation and a small number of parameters are needed to adjust (Pereira, 2011). Also, PSO provides greater diversification. Additionally, the memory capacity of PSO is efficient and better than GA. But, PSO does not utilize evolution operators such as mutation, crossover, inversion and selection. PSO is similar to ES, GA and GP in terms of initialization and updating generations. Although PSO algorithms are efficient for searching both continuous and multimodal however, there are some limitations of the PSO algorithm such as error rate is high in some situations and the performance may not be that well after the implementation of internal and external factors (Khajavi et al., 2017). Moreover, utilization of multiple parameters by PSO algorithms may affect the efficiency and speed (Tanweer et al., 2015). In addition, limitations such as easily falling into local optimum in high-dimensional space and having a low convergence rate in the iterative process motivates the researcher to either use different algorithms or improve PSO (Li et al., 2014).

2.6.3 Ant Colony Optimization

Ant Colony Optimization (ACO) is an evolutionary algorithm that mimics the behavior of Ant Colony which can solve the complicated combination optimization problems i.e. Travelling Salesman Problem (TSP). Initially, an ant locates the food source and return to the nest. Ants observe four possible ways extensively, but the runway is consolidated in a way that the route is not less attractive than shortest route. Though, ants lose their trail pheromones, they follow the shortest route (Mohapatra et al., 2013). Yang et al., (2014) applied combinatorial model to predict short-term electricity price of New South Wales in Australia by ACO algorithm based on the generalized autoregressive conditional heteroskedasticity (GARCH) model and SVM. The forecasting accuracy is improved through their model (Yang et al., 2014).

ACO has been integrated with other models to form the ensemble approach for stock prediction in recent investigations (Cai et al., 2015, Yang et al., 2014). ACO can be applied to determine an appropriate partition of stock data through engaging ants for searching. The balance between exploration and exploitation can be made through ACO algorithm where pheromone intensification of paths and exploitation is the main focus (Cai et al., 2015). Better predication accuracy can be availed through ACO integrated with other model (Cai et al., 2015).

The pseudo code of ACO algorithm is indicated in Figure 2.7. In ACO algorithm, ConstructAntSolutions is a partial solution extended by adding an edge based on stochastic and pheromone considerations. Update pheromone is a process to increase pheromone of good solutions, decrease that of bad solutions, which is also known as pheromone evaporation.

```
Begin
Set parameters, initialize pheromone trails
While terminating condition is not reached do
    ConstructAntSolutions applying pheromone trail
    Update Pheromones
End While
End
```

Figure 2.7 ACO Algorithms Pseudocode

ACO has some advantages like inherent parallelism, efficiency for TSP and similar problems and suitable for dynamic application. ACO can be implemented to solve various issues like Machine Learning Problems, network problems, stochastic optimization problems and Travelling Salesman's Problems (Stützle et al., 2011). ACO can be hybridized or ensemble with other algorithms to form robust and efficient algortihms. Moreover, ACO is very effective for distributed environment. However, loss of diversity and increased chance of premature convergence are some of the limitations of ACO algorithm (Cai et al., 2015). Moreover, ACO algorithms utilize multiple parameters such as pheromone quantity, pheromone update rule, evaporation rate, and pheromone reinforcement rate, which need to be tuned properly. In addition, difficulty in theoretical analysis and changing the probability distribution after each course of iteration has motivated the researchers to investigate for suitable algorithms (Mohapatra et al., 2013).

2.6.4 Genetic Algorithm

Genetic Algorithm (GA) is a population based meta-heuristic evolutionary algorithm that mimics biological evolution and it can solve constrained and unconstrained optimization problem through natural selection process. The population of individual solution is repetitively modified through GA algorithm. An individual from current population is randomly chosen by the algorithm, which is used as a parent to generate children for next generation. The successive generations finally produces the optimal solution through the evolving of the population. Delnavaz (2014) applied GA and fuzzy-neural network algorithm to predict the stock price for Tehran Stock Market. The result through the combinatorial algorithms is encouraging (Delnavaz, 2014).

GA has been integrated with other models to form the ensemble approach for stock prediction in recent investigations (Göçken et al., 2016, Delnavaz, 2014). GA can be applied to overcome the limitation of input variable selection and also it is potential for search and optimization problem. During evolution, GA can generate new and better population among different species. GA is capable to exploit the unknown search space through the collected information (Göçken et al., 2016).

The pseudo code of GA for stock data classification is directed in Figure 2.8. Selection, crossover and mutation are the three basic operators of GA. However, GA can be extended for better performance through the adjustment of elitism (best individuals in a population can be propagated to the next generation) or random immigrants (worst individuals in a population can be replaced by random one).

```
Begin
Generate an initial random population
Evaluate each candidate solution
While terminating condition is not reached do
    Select individuals for the next generation
    Recombine pairs of parents
    Mutate the resulting offspring
    Evaluate each candidate solution
End While
End
```

Figure 2.8 GA Algorithm Pseudocode

Nevertheless, there are some limitations persist with GA algorithm such as hidden layer has to remain fixed due to time-consuming training, transfer and training function need to be fixed as the combination of both may deteriorate the quality (Göçken et al., 2016). Moreover, difficulty in identifying the fitness function because of occurrence of premature convergence, complexity in choosing population because of mutation rate, crossover rate, time consuming decoding, and fitness evaluation may be experienced with GA. Hence, the limitations of GA diverts the researcher to investigate for other good algorithms (Razali et al., 2011).

2.6.5 Evolutionary Strategy

Evolutionary Strategy (ES) algorithm is capable to optimize and search the space through the simulation of the genetic evolutionary process proposed by Darwin theory consisting selection, mutation, recombination and reproduction. The algorithm measures the performance through the application of fitness function and efforts the evolution place in better search space regions.

ES algorithm has been integrated with other models to form the ensemble approach for stock prediction in recent investigations (Hu et al., 2015, Bliss et al., 2014, Bisoi et al., 2014). Hu et al. (2015) investigated and proposed that EA algorithm can be applied for rule discovery in stock algorithm trading (AT) (Hu et al., 2015). Bliss et al. (2014) applied EA algorithm to predicting future links in social networks by applying the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) to optimize weights (Bliss et al., 2014). Bisoi et al. (2014) proposed use of combinatorial evolutionary dynamic neural network for stock market trend analysis and prediction using unscented Kalman filter.

The pseudo code of ES is indicated in Figure 2.9. In ES, a population of individual is created so that it represents the solution of a problem. The process is considered as a solution of some problem and it is similar to genes in natural evolution. The fitness of each gene is assessed to measure the capability in solving problems. The best gene is chosen at this stage to reproduce for next generation and new child solution is formed from the parent through reproduction. The mutation on the child solution is executed and the process is iterated till suitable solution has been achieved.

```
Begin
Create a population of individuals
While terminating condition is not reached do
    Evaluate the fitness of each of the individuals
    Select the best individuals for reproduction to form the next generation
    Perform reproduction on the parent solutions to form new child solutions
    Perform mutations on the child solutions
End While
End
```

Figure 2.9 ES Algorithm Pseudocode

However, ES algorithm has some limitations such as finding optimal solutions in a finite amount of time is not guaranteed, parameter tuning mostly by trial-and-error, and population approach may be expensive in terms of other meta-heuristic algorithms, influences the researchers to look for alternate solution (Bisoi et al., 2014).

2.6.6 Probability Based Incremental Learning

Probability Based Incremental Learning (PBIL) is an evolutionary optimization algorithm, which is able to create the real valued probability vector for the object of the algorithm, and it can produce high evaluation solution vectors with high probability through sampling (Baluja, 1994). PBIL algorithm is a better algorithm for solving real-world problems than GA and hill-climbing and it is formed through the generalization of GA to preserve the statistics of population produced by GA.

PBIL algorithm has been integrated to form the ensemble approach for stock prediction (Monteiro et al., 2018, Ali et al., 2014). Monteiro et al. (2018) investigated the application of probability based ensemble model to predict for a day ahead Iberian Electricity Market. Ali et al. (2014) applied PBIL algorithm for determining the Egyptian stock market trend through the enhancement of the performance of multi-layer perceptron and achieved better result. Numerous researches have been concentrated on incremental learning than selective learning.

The pseudocode for PBIL algorithm is indicated through Figure 2.10. PBIL uses an initial probability vector initialized to 0.5 for every entry. The reason for choosing 0.5 is that the probability of generating 1 or 0 for each course of iteration is equal (Monteiro et al., 2018) but the values will be updated through learning as the search continues.

```
Begin
Initialize the probability vector $P(i)$ = 0.5
While terminating condition is not reached do
    $M$ = generate samples from probability vector $P$
    Evaluate samples($M$)
    $B$ = select best solutions from($M$)
    $P(i) = (1-\alpha) * P(i) + \alpha * B(i)$
End While
End
```

Figure 2.10 PBIL Algorithm Pseudocode

However, some of the drawbacks of PBIL algorithm such as PBIL algorithm depends on the inversion of information matrix, PBIL can only converge to local optima though in case of unimodal functions PBIL can converge to the global optimum, and PBIL uses single probability vector which may have less expressive power (Monteiro et al., 2018) can be addressed either by implementing other optimization algorithm or improving PBIL (Ali et al., 2014).

2.6.7 Bio-geography Based Optimization

The optimization of neural network can be performed through the application of Biogeography-Based Optimization (BBO) (Mirjalili et al., 2014a) in training MLPs. BBO is an evolutionary algorithm that applies evolutionary mechanisms to each individual in a population. BBO can provide more flexible training procedures compared to others for the search space of MLP that is changeable for different datasets. It tends to outperform GA due to applying various evolutionary operators.

The pseudocode of BBO is as indicated in Figure 2.11. BBO algorithm will initially outline the island modification probability, mutation probability, and elitism parameter and initialize the population. The immigration rate and emigration rate will be calculated for each island, provided that the solution will be considered as good if it has high emigration rates and low immigration rates. Otherwise, if it has low emigration rates and high immigration rates then the solution will be treated as bad. Here, the immigration islands will be chosen based on the immigration rates probabilistically and roulette wheel selection will be used based on the emigration rates to select the emigrating islands. Then, randomly selected Suitability Index Variables (SIVs) will be migrated based on the selected islands where the migration will take place randomly. BBO performs mutation based on the mutation probability for each island probabilistically. Finally, fitness of each individual island will be calculated and the process continues until the target is achieved.

```
Begin
Generate an initial random population
While terminating condition is not reached do
    Calculate the immigration rate and emigration rate for each island
    Choose the immigration islands based on the immigration rates probabilistically
    Migrate randomly selected Suitability Index Variables (SIVs) based on the
    selected islands
    Perform mutation based on the mutation probability for each island
    Probabilistically
    Calculate the fitness of each individual island
End While
End
```

Figure 2.11 BBO Algorithm Pseudocode

BBO is a meta-heuristic algorithm that applies evolutionary mechanisms to each individual in a population. BBO can provide more flexible training procedures compared to others for the search space of MLP that is changeable for different datasets. It tends to outperform GA due to applying various evolutionary operators (Mirjalili et al., 2014a). Usually, heuristic algorithms are employed for solving a particular problem by determining a combination of weights and biases that provide the minimum error for an MLP. The architecture does not change during the learning process in this method. For minimizing the overall error of MLP, the training algorithm needs to discover proper values for all connection weights and biases. Generally, there are three methods of using a heuristic algorithm for training MLPs. Firstly, heuristic algorithms are utilized for searching. Secondly, heuristic algorithms are employed to find a proper architecture for an MLP in a particular problem. The last method is to use a heuristic algorithm to tune the parameters of a gradient-based learning algorithm, such as the learning rate and momentum. The weights and biases are encoded using vector to train an MLP. The encoding is easier in this way though the decoding is a bit complicated. This method is used often for simple neural network structure and it is appropriate for the problem, which can't deal with complex MLP structure (Haykin S., 1994).

BBO algorithm can be integrated to form ensemble approach for better classification and solving prediction problem. Moreoever, BBO has much scope to grow as this research community is quite young. Significant challenging tasks can be addressed through BBO by exploring new approach (Zhang et al., 2016, Mirjalili et al., 2014a).

However, BBO algorithm may have some limitations such as poor in exploiting the solutions, no provision for selecting the best members from each generation and a habitat doesn’t consider its resultant fitness while immigrating the features may result the generation of many infeasible solutions. The extension of BBO and ensemble with other models may be investigated to address the limitations (Ammu et al., 2013).

2.7 Artificial Neural Network and Its Training

The architecture of human brain consists parallel neurons network, which can enormously control human intelligence. Essentially, an appropriate outline of an ANN which is also known as Neural Networks (NN) can substantially contributes to its learning. Node, weight and layers can be adjusted to construct an appropriate ANN based on a problem at hand. ANN can be single layer or multi-layer. However, single layer consists of one input and an output layer which is appropriate for solving linear problems. On the other hand, a multi-layer consists of input layer, output layer and one or more hidden layer that can solve non-linear problems. ANN and its improvement for time series prediction has been investigated recently by many researchers for numerous investigations (Wanto et al., 2017; Lahmiri et al., 2016; Balabanov et al., 2011; Neukukar et al., 2010). Back-propagation algorithm is used before to train multi-layer neural network but appropriate learning algorithm can significantly improve the performance of neural network for pattern recognition, prediction and many other diverse applications.

The neural network needs to be trained to generate the output or target much closer to desired one. Berry et al. (1997) defined the “Training” as a process of producing, finding or setting the weights in a neural network to produce good prediction result. Numerous algorithms are offered by neural network for the purpose of training but back-propagation is the widely accepted one for training multilayer perceptron network among the available options (Wu et al., 2012) due to its ability to faster convergence and mathematical compliance. Levenberg–Marquardt and Gradient Decent are extensively used for training through back-propagation however they are computationally expensive to support large neural networks especially, when the network to be trained consists substantial amount of adaptive weights (Schmidhuber, 2015). Figure 2.12 indicates the Back-propagation algorithm.

```
Input: D, a dataset consisting of the training tuples and their associated target values
L, the learning rate Network, a multilayer feed forward network
Output: A trained network
Begin
Initialize all network weights and biases
While terminating condition is not satisfied do
  For each training tuple X in D
    For each input layer unit j
      Output of an input unit, $O_j$ = actual output value, $I_j$
    End For
    For each hidden or output layer unit j
      Compute the net input of unit j with respect to the  previous layer, i
      $I_j = \sum i w_{ij} O_i + \Theta_j$
      Compute the unit of each unit j, $O_j = \frac{1}{1+ e^{-1_j}}$
    End For
    For each unit j in the output layer
      Computer the error, $Err_j = O_j (1- O_j)( T_j - O_j)$
    End For
    For each unit j in the hidden layer,
      Compute the error with respect to the next higher layer, k
      $Err_j = O_j (1- O_j) \sum_k Err_k w_{jk}$
    End For
    For each weight $w_{ij}$ in network
      Weight increment, $\Delta w_{ij} = (l) Err_j O_i$
      Weight update, $w_{ij} = w_{ij} + \Delta w_{ij}$
    End For
    For each bias $\Theta_j$ in network
      Bias increment, $\Delta \Theta_j = (l) Err_j$
      Bias update, $\Theta_j = \Theta_j + \Delta \Theta_j$
    End For
  End For
End While
End
```

Figure 2.12 Back Propagation Algorithm

Source: Han et al., 2012

The basic structure of a Multi-Layer Back-Propagation Neural Network is indicated in Figure 2.13.

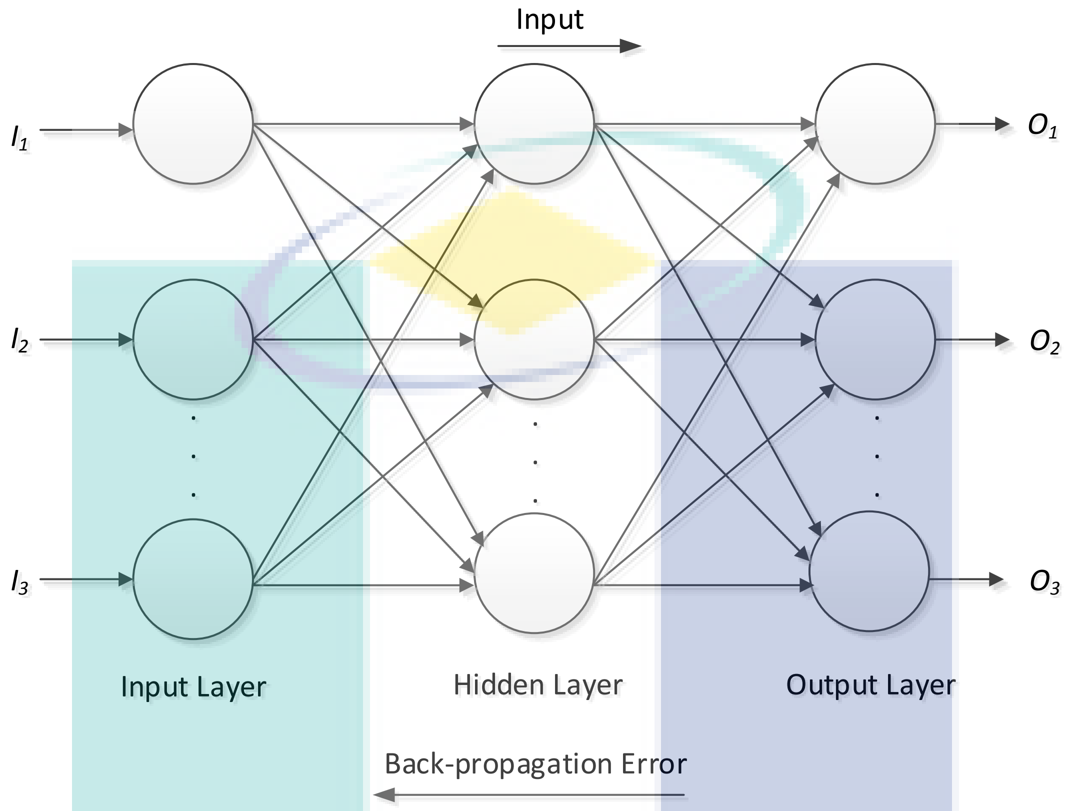


Figure 2.13 Multi-Layer Back-Propagation Neural Network Structure

As indicated earlier, meta-heuristic algorithm is an effective and efficient approach to optimize neural network as the algorithm can balance both exploration and exploitation. As a result, the complex and non-linear problem can be solved through this approach. However, it is always better to improve, ensemble or hybrid meta-heuristic to avail maximum outcome through meta-heuristic (Ojha et al., 2017). Although, many meta-heuristic algorithms are used for training neural network, Bio-Geography Based Optimization and Grey Wolf Optimizer can be some of the better choices for training MLP neural network (Mirjalili et al., 2014a, 2014b).

2.8 Meta-heuristic Algorithms for Training Neural Network

Neural Network can be trained applying meta-heuristic algorithms such as Genetic Algorithm, Particle Swarm Optimization, and Ant Colony Optimization. However, the current study explains only Grey Wolf Optimizer for training NN as per the scope of the research.

In the context of the current work, the combination of multiple algorithms to improve the accuracy and stability of a classification model is known as an ensemble model. Researchers attempted for prediction applying ensemble algorithms acclaimed that an ensemble model built efficiently can concurrently make accurate prediction and manage the prediction errors well in diverse areas of the input space (Jothimani et al., 2016). The ensemble and similar approaches in financial prediction have been very popular and successful in recent years (Sujatha et al., 2018, Niu et al., 2016, Lamhiri, 2014). Co-operative ensemble approach is the attention in this research where the prediction task can be divided into numerous sub-tasks to gain the prediction accuracy and the outcome of the prediction is sum of all sub-tasks. Figure 2.14 represents the architecture of an Ensemble Model, where input is forwarded to the sub task 1, sub task 2 and sub task 3. The output produced by the various sub tasks will be combined together and then forwarded to the final ensemble output.

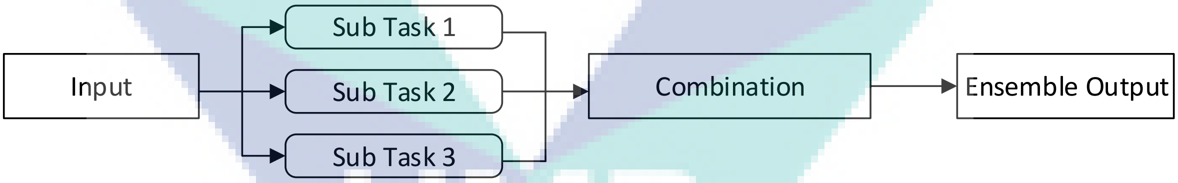


Figure 2.14 Ensemble **Model** Architecture

2.8.1 Review on Grey Wolf Optimizer

Grey Wolf Optimizer (GWO) is a new meta-heuristic algorithm, which is emulated from grey wolves (Canis lupus), and it mimics the leadership hierarchy and hunting mechanism of grey wolves in nature (Mirjalili et al., 2014b). The leadership hierarchy is simulated applying four types of grey wolves i.e. $\alpha, \beta, \delta, \omega$ and hunting is implemented applying three main steps i.e. searching for prey, encircling prey, attacking prey. The GWO algorithm is applicable to challenging problems in unknown search spaces which produces better result than PSO, GSA, DE, EP and ES (Faris et al., 2018, Gupta et al., 2017, Mirjalili et al., 2014b). GWO algorithm can be applied as a training algorithm for Multi-layer perceptron. The algorithm improves the wolves attack strategy. It calculates the weights based on wolves fitness function and gives the highest weight to the dominant wolf concurrently to improve the convergence. The empirical results confirm the power of GWO and demonstrate that the algorithm can faster decide the suitable thresholds, provide good classification rate, efficiency and accuracy. The flow of GWO algorithm is indicated in the Figure 2.15.

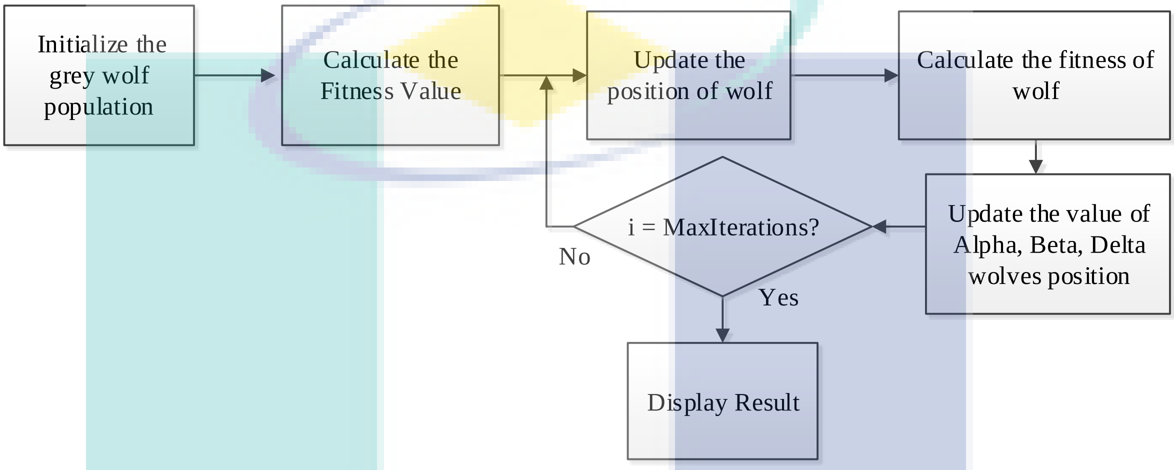

Initialize the grey wolf population
Calculate the Fitness Value
Update the position of wolf
Calculate the fitness of wolf
Update the value of Alpha, Beta, Delta wolves position
i = MaxIterations?
No
Yes
Display Result

Figure 2.15 Flow of GWO Algorithm

The solution of a problem through meta-heuristic algorithm needs to address two conflicting processes known as exploration and exploitation (Emary et al., 2018). The exploration process facilitates the algorithm to discover new areas into the problem search space through engaging abrupt alterations to the solutions. The promising areas may be explored to the search landscape and solution may be exempted from stagnation into local optimum through exploration. On the other hand, the exploitation process entails the algorithm to discover the neighboring zone so that expected solutions attained through exploration can be improved. Exploitation performs gradual adjustment to the solution so that the solution converges to the global optimum. GWO algorithm can be adjusted to make a good balance between both exploration and exploitation (Faris et al, 2018, Mirjalili et al., 2014b).

2.8.1.1 The Original Grey Wolf Optimizer Algorithm

GWO is inspired through the searching by grey wolves in nature for optimal way of hunting preys and the algorithm is developed by Mirjalili et al. in 2014 (Mirjalili et al., 2014b). The detail overview of GWO algorithm is explained in this section.

2.8.1.2 Basic Characteristics of Grey Wolf Optimizer Algorithm

Grey wolves (Canis lupus) belong to Canidae family and they are considered as apex predators, meaning that they are at the top of the food chain. They have a very strict social dominant hierarchy. The algorithm divides the wolves into four types: *α, β, δ, and ω*, whereas, each type of wolves displays the following social behavior.

The leaders are a male and a female, called alpha. Decision making about hunting, sleeping place, and time to wake, is made by alpha. Due to these reasons, alpha becomes the leader in the pack and others follow its orders. The best solution of a problem can be determined identifying the location of alpha, as it is the best member in managing the pack.

The second level in the hierarchy of grey wolves is beta. The betas are subordinate wolves that help the alpha in decision-making or other pack activities. The beta maintains the discipline of the pack that enforces the alpha’s commands in the pack and hence, it takes the role of the advisor to the alpha.

The third level in the hierarchy of grey wolves is delta and they need to submit to alphas and betas, but they dominate the omega. They are responsible for watching the boundaries of the territory, warning the pack in case of any danger, protecting and guaranteeing the safety of the pack, helping the alphas and betas when hunting prey, and providing food for the pack and caring for the weak, ill, and wounded wolves in the pack.

The last grey wolf in the hierarchy is omega and it may not be an important individual in the pack, but it has been observed that the whole pack face internal fighting and problems in case of losing the omega, which is harmful to the group structure.

Group hunting is another interesting social behavior of grey wolves in addition to the social hierarchy. The main phases of grey wolf hunting are as follows: searching for the prey; tracking, chasing, and approaching the prey; pursuing, encircling, and harassing the prey until it stops moving; attacking toward the prey.

The fittest solution is considered as the alpha (*α*) in designing mathematical model of the social hierarchy of wolves for designing GWO (Mirjalili et al., 2014b). Beta (*β*) and delta (*δ*) are respectively the second and third fittest solution. Omega (*ω*) is considered for the rest of the candidate solutions. In the GWO algorithm, the hunting (optimization) is guided by *α*, *β*, and *δ*. The *ω* wolves follow these three wolves. GWO can be summarized as per the Figure indicated in 2.16.

```
Begin
Initialize the grey wolf population, $X_i$ ($i$ = 1,2,3,...,$n$)
Initialize $a$, $A$ and $C$
Calculate the fitness of each search agent, where, $X_\alpha, X_\beta, X_\delta$ are the best, second best and third best search agent consecutively
While $i$<MaxIterations
    Update the position of current search agent for each search agent by equation
    $\vec{X}(t+1) = (\vec{X}_1 + \vec{X}_2 + \vec{X}_3)/3$
    Update $a$, $A$ and $C$
    Calculate the fitness of all search agents
    Update $X_\alpha, X_\beta$, and $X_\delta$
    $t = t + 1$
End While
Return $X_\alpha$
End
```

Figure 2.16 Psudocode of GWO Classification Algorithm

The original Grey Wolf Optimizer (GWO) algorithm is illustrated in Figure 2.16, where, X_i represents the initial population of grey wolf; the GWO parameters such a, A, C are the vectors; t represents the maximum number of iteration.

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a} \qquad 2.8$$

$$\vec{C} = 2.\vec{r}_2 \qquad 2.9$$

In Equation 2.8, the values of $\vec{a}$ are linearly decreased from 2 to 0 over the course of iterations. At this stage, the estimation of the fitness for each search agents is made and the hunt agents are identified such as best hunt agent X_{α}, the second best hunt agent X_{β} and the third best hunt agent X_{δ}. The updating of the location for the current hunt agent is made using the Equation, $\vec{X}(i+1) = (\vec{X}_1 + \vec{X}_2 + \vec{X}_3\,) \,/\, 3.$ Then, the vectors are updated. Next, the fitness value of all hunts are estimated and the value for $X_{\alpha}, X_{\beta}\ and\ X_{\delta}$ are updated. The stopping condition are checked here to determine whether the iteration (t) reaches max number of iterations, if yes, then return and print the best value of solution X_{α}, otherwise, the algorithm will start through the same Equation, $\vec{X}(i+1) = (\vec{X}_1 + \vec{X}_2 + \vec{X}_3\,) / 3.$

2.8.1.3 Mathematical Model of Grey Wolf Optimizer Algorithm

GWO can be formed as per the mathematical equations below (Mirjalili et al., 2014b):

$$\vec{D} = \left| \vec{C} \,.\vec{X}_p(t) - \vec{X}(t) \right| \qquad 2.10$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}.\vec{D} \qquad 2.11$$

Where, t denotes the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\vec{X}_p$ is the position vector of the prey, and $\vec{X}$ denotes the position vector of a grey wolf.

The vectors $\vec{A}$ and $\vec{C}$ are calculated as follows:

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a} \qquad 2.12$$

$$\vec{C} = 2.\vec{r}_2 \qquad 2.13$$

Where, components of $\vec{a}$ are linearly reduced from 2 to 0 over the number of iterations and used for controlling the trade-off between exploitation and exploration. The following equations will be employed for updating the value of variable:

$$\vec{a} = 2 - t\ (2 \ / \ X_i\ ) \tag{2.14}$$

$$\vec{X}(t+1) = (\ \vec{X}_1 + \ \vec{X}_2 + \ \vec{X}_3\ )\,/\,3 \tag{2.15}$$

Where, X_i denotes the number of iterations, $\vec{r}_1$ and $\vec{r}_2$ are random vectors between [0, 1] which are employed to find the optimal solution. Appropriate idea about the potential location of prey can be availed by Alpha, Beta and Delta, where they help the Omega to follow the suitable positions. The values of $\vec{X}_1$, $\vec{X}_2$ and $\vec{X}_3$ can be obtained through the equations below:

$$\vec{X}_1 = \ \vec{X}_\alpha - \ \vec{A}_1 . \vec{D}_\alpha \tag{2.16}$$

$$\vec{X}_2 = \ \vec{X}_\beta - \ \vec{A}_2 . \vec{D}_\beta \tag{2.17}$$

$$\vec{X}_3 = \ \vec{X}_\delta - \ \vec{A}_3 . \vec{D}_\delta \tag{2.18}$$

In iteration *t*, the best 3 solutions are respectively, $\vec{X}_1$, $\vec{X}_2$ *and* $\vec{X}_3$. Where, the values of $\vec{D}_\alpha, \vec{D}_\beta$ and $\vec{D}_\delta$ are as below:

$$\vec{D}_\alpha = \ |\ \vec{C}_1 . \vec{X}_\alpha - \vec{X}\ | \tag{2.19}$$

$$\vec{D}_\beta = \ |\ \vec{C}_2 . \vec{X}_\beta - \vec{X}\ | \tag{2.20}$$

$$\vec{D}_\delta = \ |\ \vec{C}_3 . \vec{X}_\delta - \vec{X}\ | \tag{2.21}$$

Exploration and exploitation in GWO can be expressed as follows:

Parameter $\vec{C}$ is the key element to facilitate exploration in terms of local optima stagnation as it contains random values between [0, 2] that offers random weights for prey to stochastically emphasize *C* > 1 and deemphasize *C* < 1. As a result, the solution inclines closer to the prey. Whereas, parameter $\vec{A}$ is another source of exploration as the value of the parameter is controlled by a, that can be linearly declined from 2 to 0. The range of parameter $\vec{A}$ alters between the interval of [-2, 2] as it contains random element. The value of $\vec{A}$ > 1 and $\vec{A}$ < -1 ensures exploration so that GWO algorithm starts searching globally. Conversely, the value of $\vec{A}$ > -1 and $\vec{A}$ < 1 ensures exploitation.

2.8.1.4 Variations of Grey Wolf Optimizer Algorithm

In the course of the most recent couple of years, different variations of GWO have been acquainted owing with various improvement issues. The modification of GWO algorithm has been proposed to comply with the difficult real-world optimization problem. Due to the constraint of GWO to handle real-world problems, some modifications are proposed through update mechanism, some proposed to improve GWO operations, some proposed to enable the exploration and exploitation through ensemble or hybridization and some proposed to handle parallel computing platforms. This section is planned to give a quick overview about proposed GWO's variations and upgrades.

Mittal et al., (2016) proposed the improvement of GWO exploration through application of exponential decay function as indicated in Equation 2.22. The approach recommended to reduce the value of parameter *a* exponentially replacing linear modification. The proposed solution was tested over 27 benchmark functions and attained better result compared to other prominent meta-heuristic algorithms such as PSO, BA, CS and GWO. Whereas, Long et al., (2017) investigated the ensemble of Modified Augmented Lagrangian (MAL) with Improved Grey Wolf Optimizer (IGWO) to adapt the parameter *a* applying the Equation indicated in 2.23. The study attained a better result through the nonlinear adaptation with an appropriate balance between exploration and exploitation.

$$a = 2\left(1 - \frac{Iteration^2}{MaxIteration^2}\right) \qquad 2.22$$

$$a = \left(1 - \frac{Iternation}{MaxIteration}\right).\left(1 - \mu.\frac{Iternation}{MaxIteration}\right)^{-1} \qquad 2.23$$

Where, μ is a nonlinear modulation index at the interval (0, 3).

In the recent time, GWO algorithm has been implemented for feature selection with the objective of selecting most appropriate features, decreasing number of features and removing irrelevant, noisy and redundant features. Li et al., (2017) investigated the ensemble of binary GWO and wrapper-based method for feature selection. The study attempted to address medical diagnosis problem through application of a classifier called Kernel Extreme Learning Machine. Emary et al., (2016) proposed an ensemble of binary GWO and k-nearest neighbor (KNN) where GWO is applied as a feature selection approach. The study attained better result with faster convergence compared to GA and

PSO. Another study for feature selection applying GWO was attempted by Emary et al., (2016). The study was successful to produce encouraging results with an option to avoid local minima.

Currently, another application of GWO attracted the researchers' attention in training neural network or ANN integrating GWO. The most common neural network is MLP, which is applied for classification. Mosavi et al., (2016) applied GWO-based training in combination with MLP for three different data sets and attained reasonable result compared to PSO, Gravitational Search Algorithm (GSA) and PSOGSA. Similar model was applied by Mohamed et al., (2015) for training MLP. The study was successful in producing lower error rate with faster convergence for MLP. In another study, Mirjalili et al., (2014b) investigated the application of GWO to train MLP and produced better result in comparison with PSO, GA, ACO, ES and PBIL.

As featured earlier, GWO has been utilized by numerous researchers because of its benefits over others, GWO depends on parameters which can balance between exploration and exploitation. Moreover, GWO is simple and flexible which utilizes basic analogy including the grey wolves in nature for hunting preys. Consequently, their usage is clear. At this juncture, the standard GWO has been demonstrated its potential for taking care of unimodal optimization issues, although when confronting complex multimodal optimization issues with substantial amount of local minima, the GWO is generally getting stuck into a local minima because of shortcoming of its population's decent variety (Faris et al., 2018).

Keeping in mind the shortcoming of GWO, the ensemble can be an effective option to enhance GWO's performance (Faris et al., 2018, Mosavi et al., 2016, Mohamed et al, 2015). Notwithstanding the reality, ensemble can enhance the GWO's performance, an excess of ensemble may invoke more complicacy to the algorithm. Moreover, exploration and exploitation are the key operations to improve meta- heuristic algorithm's proficiency. In the literature, different types of approaches have been adopted by various algorithms such as crossover, mutation, and elitism operators in GA, random walks in CS and parameter adaptation in GWO. To balance the exploration and exploitation, all meta-heuristic algorithm utilizes directly/indirectly a mechanism or operator.

2.8.2 Review on Neural Network Training by Grey Wolf Optimizer

Neural Network (NN), which is motivated by biological systems can be applied for information processing effectively. NN has been used extensively over a long period of time due to its dynamic behavior and excellent ability to handle nonlinear data. But, the performance of NN heavily depends on the weights and structure of the network. Moreover, the training of NN is another important issue that needs appropriate algorithm to produce better outcome. Generally, new meta-heuristic algorithms are explored to determine the algorithm's ability to optimize the NN. In this regard, researchers have been investigated recently the optimization ability of NN by GWO.

As a part of investigation to optimize NN applying GWO, Mirjalili (2015) attempted to apply GWO for training multi-layer perceptron. The study gained high exploration and exploitation that could outperform other popular trainers such as PSO, GA, ACO, ES and PBIL. The investigation is able to produce very competitive results and improve local optima avoidance. The classification accuracy is also very good for the study. However, the study recommended investigating the application of GWO to determine the optimal structure of MLP and fine-tune GWO to produce better solution (Mirjalili, 2015).

In another study, Nur and Ülker (2018) investigated the application of GWO for optimizing NN to propose a hybrid cloud-based Intrusion Detection and Response System (IDRS). The study achieved good result, which could successfully detect intrusion over the cloud. Moreover, GWO-NN produced better classification accuracy compared to other classification algorithms such as Naïve Bayes (NB) and Gravitational Search Algorithm with NN (GSA-NN) for two different data sets. However, the classification accuracy of GWO-NN was lower than Multi-layer Perceptron with Back propagation (MLP-BP) for one data set and lower than both MLP-BP and Particle Swarm Optimization with NN (PSO-NN) for another data set. In addition, GWO-NN approach was slower in convergence compared to NB during training. Hence, the study recommended modifying GWO to improve the grey wolf performance (Nur & Ülker, 2018).

Parsian et al., (2017) attempted to optimize NN applying GWO for melanoma detection. The study trained NN applying GWO to determine the optimal initial weights where the GWO-NN produced better classification rate of 90% compared to ordinary MLP that produced 88%. Moreover, the convergence speed of GWO-NN was really faster and Root Mean Square Error (RMSE) was also reduced through this approach. However, the study did not attempt to compare the performance of GWO-NN with other classification algorithms such as PSO, GA and ACO. Additionally, the investigation did not modify GWO to improve the classification performance, which can be attempted for future research (Parsian et al., 2017).

In another investigation, Turabieh (2016) attempted to optimize NN applying GWO to predict heart disease. The study produced better prediction result for the heart disease related medical data set through the parameter tuning by GWO-NN and locating initial weights and biases by GWO. Moreover, GWO-NN produced lower RMSE value i.e. close to 0 and converged much faster. The study also compared the performance of GWO-NN with standard NN and identified that GWO-NN performs much better than standard NN in terms of prediction accuracy, convergence speed and local minima avoidance. However, the study applied back-propagation algorithm for training and did not attempt to compare the performance of GWO-NN with other classification algorithms such as PSO, GA and ACO. Additionally, the investigation did not modify GWO to improve the classification performance and hence suggested to perform further research to determine the optimal NN structure through the modification of GWO (Turabieh, 2016).

Meanwhile, Mosavi et al., (2016) conducted a study to perform classification of sonar data set applying GWO-NN approach where GWO was implemented for training NN. The investigation was successful to overcome the limitations such as improper classification accuracy, slow convergence speed and trapping in local minimum through GWO-NN. The performance of GWO-NN was also compared in this study with other classification algorithms such as Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA) and the hybrid algorithm (i.e. PSOGSA) applying convergence speed, the possibility of trapping in local minimum and classification accuracy metrics for three data sets where, GWO-NN outperformed for all data sets. However, the study did not attempt to modify GWO to investigate the classification performance. Additionally, the study recommended applying GWO or its modification to determine the optimum structure of NN as a future research (Mosavi et al., 2016).

Similar kind of NN optimization through application of GWO was investigated by the authors in (Mohamed et al., 2015) for designing of the static VAR compensator (**SVC**) controller for damping power system oscillations. GWO-NN based approach produced better outcome for this research with lower error values and faster convergence. However, the study did not perform the comparison of GWO-NN approach with other classification algorithm such as PSO, GA and ACO. Moreover, modification of GWO could be investigated to produce better classification outcome and determine optimum NN structure (Mohamed et al., 2015).

To sum up, GWO has demonstrated great potential for optimization of NN as a recent swarm-intelligence based meta-heuristic algorithm. However, some limitations of the algorithm needs further investigation such as GWO cannot solve all optimization problems by way of NFL suggestion, GWO can solve only single-objective problems, multi-modal search landscape is difficult to be handled by GWO because the operators are converged to identical solution, more number of variables worsens the performance of GWO due to entrapment in local solutions, GWO may produce local solutions for a problem containing large number of variables and local solutions due to faster convergence and exploitation, GWO has also limitation in terms of exploration rate as it has the possibility of being stagnant with its limited operators' alpha, beta and delta. Moreover, the encircling model recommended by GWO may performs the exploration to limited extent only, so GWO needs more operators to increase the exploration rate. Hence, GWO needs to be modified or extended to solve complex problems (Faris et al., 2018; Gupta et al., 2017; Mirjalili, 2015; Mirjalili et al., 2014b; Nur & Ülker, 2018).

2.9 Gap Analysis on the Need for Modified Grey Wolf Optimizer

Table 2.1 presents the gap analysis in the current adoption of GWO for tuning ANN where the common approach GWO is applied to optimize ANN for different type of investigations and data sets.

Table 2.1 Gap Analysis Findings Summary

Strategy	Main Features	Limitations and Recommendations
GWO-MLP (Mirjalili, 2015)	• Addressed the exploration and exploitation • Better performance than popular trainer such as PSO, GA, ACO, ES and PBIL • Improved local optima avoidance • Good classification accuracy for selected data set	• Investigation is required to apply GWO for determining optimal structure of MLP • Fine tuning of GWO is required to produce better solution
GWO-NN (Mosavi et al., 2016)	• The investigation addresses the limitations such as improper classification accuracy, slow convergence speed and trapping in local minimum • Good performance for selected data sets	• Classification performance is not at 95% confidence level for different data set • Modification of GWO is required for different data set to enhance performance
GWO-NN (Turabieh 2016)	• Produced good prediction result for the heart disease related medical data set • Produced lower RMSE value • Better performance than standard NN	• Classification performance is not at same level for other data set • GWO needs to be modified to improve the classification performance for different data set
GWO-NN (Parsian et al., 2017)	• Produced better classification rate compared to ordinary MLP • Faster convergence • Reduced Root Mean Square Error (RMSE)	• Classification performance is not at 95% confidence level for various data sets • Modification of GWO is recommended for better performance
GWO-NN (Nur and Ülker 2018)	• Able to detect intrusion over the cloud • Balanced exploration and exploitation • Better classification accuracy compared to Naïve Bayes (NB) and Gravitational Search Algorithm (GSA)	• Classification accuracy is lower for some data set • Slower in convergence compared to NB • Modification of GWO is suggested

As seen from the analysis, the demand for proposing a new algorithm or improving earlier algorithm is enormous to deal with the limitation of existing algorithms. The requirement of a new optimization algorithm is also highlighted by No Free Lunch (NFL) theorem that a single algorithm cannot solve all the optimization problems optimally (Wolpert & McReady, 1997). In line with the gap analysis discussed earlier, this research will endeavor to address the gap indicated above by planning and executing another attempt in view of modified Grey Wolf Optimizer (MGWO) applying ensemble approach. Grey Wolf Optimizer (GWO) is one of the most recent swarm intelligence-based meta-heuristic algorithms shaped for addressing the problem of global optimization. Grey wolves' hunting and leadership hierarchy in nature motivates the inspiration of such algorithm. In light of supplementing existing work on meta-heuristic based ensemble strategies, adopting Modified Grey Wolf Optimizer (MGWO) has all the earmarks of being an appealing choice. Specially, GWO has benefits over other meta-heuristic algorithms (Faris et al., 2018, Gupta et al., 2017, Mirjalili et al., 2014b):

(i) GWO is simple and flexible SI-based algorithm that produces random population of grey wolves. The computation facilitated by GWO is lightweight compared to other meta-heuristic algorithm like GA and PSO.

(ii) GWO implements intense activities controlled by two parameters to balance exploration and exploitation so that local optima stagnation can be avoided.

(iii) The mathematical model offered by GWO is novel, although the estimation of global optimum is analogous to other population-based algorithm. Moreover, GWO has the ability to displace a solution to another n-dimensional search space.

(iv) GWO requires less memory contrasted with PSO as it contains only one vector because, PSO requires two vectors namely, position and velocity. Additionally, GWO retains just three best solutions, while PSO retains one best solution gained through all particles. The mathematical calculations of PSO and GWO are dissimilar. GWO is considered as a standout amongst the most developing SI algorithms. The success of GWO algorithm propels different scientists to apply the algorithm for various optimization problems. Till date, GWO has been utilized successfully for solving numerous problems but not limited to, global optimization problems, electric and power engineering problems, scheduling problems, power dispatch problems, control engineering problems, robotics and path planning problems, environmental planning problems (Faris et al., 2018).

However, GWO can be improved to address few shortcomings:

(i) Balancing of convergence and exploitation is required for GWO to avoid local optima. Because, the current best optimal individual is biased toward alpha and other individuals (e.g. beta and delta) attempt to modify their positions toward this best individual in each iteration process. Consequently, this update process may cause the algorithm to fall to local optima especially in the cases where there are many competing local optima.

(ii) Poor search agents among alpha, beta, delta wolves of GWO deteriorate exploitation. Hence, the exploitation needs to be improved by avoiding such search agents.

(iii) Poor search agents around search space worsen exploration. So, the exploration is required to be enriched by avoiding similar search agents.

(iv) The positions of wolves are updated mostly based on the experience of alpha, beta, and delta leaders in GWO that lead to premature convergence. Thus, measure should be taken to keep away premature convergence.

(v) More number of variables degrade the performance of the GWO algorithm due to the entrapment of the initial population in a local solution. So, number of variables need to be controlled.

2.10 Summary

This chapter presents the explanation of various algorithms in light of proposed research and related areas. Natural computing is at the central focus for optimization problems related to engineering, industrial and scientific arena for few decades. However, Nature-inspired computing has received special attention due to its efficiency in producing quality search outcome applying trajectory or population-based meta-heuristic or heuristic.

The problem of searching has been addressed in a different way by population-based meta-heuristic and Trajectory-based algorithms. Initially, the Trajectory-based algorithms make a random assumption for the solution and then the solution is further improved at the later stages of the algorithm whereas, the population-based algorithms apply incremental approach to estimate the solution with an option to backtrack at the later stages of the algorithm. The advantage of implementing backtracking facilitates the population-based meta-heuristic algorithms to function well in multi-modal search environments as each solution can be tracked. Conversely, Trajectory-based algorithms can track the problems pertaining to solutions where, the algorithms continuously modify the attained outcome till the terminating conditions are met for the search process. Trajectory-based algorithm can also apply different built-in method to resolve the issue that facilitates the algorithms to repeatedly visiting the locations/states, which is a very challenging issue in combinatorial searching. On the other hand, population-based algorithms are the appropriate choice for handling such issue due to the algorithms capability to track the areas explored earlier. Moreover, another good characteristic of population-based algorithm is the capability of solving the problem with greater efficiency. Overall, the key reason to choose this research is due to the efficacy of the population-based algorithms.

This chapter has reviewed meta-heuristic algorithms. Population based meta-heuristic algorithms PSO, GA, ACO, ES, PBIL and BBO has been presented. Next, neural network and its training with GWO algorithm has been explained. Then after that, ensemble model has been discussed.

The intensive research investigated that there are still limitations with existing meta-heuristic based models in terms of entrapment in local minima, balance between exploration and exploitation, and convergence. The next chapter will devise an MGWO based ensemble model for stock prediction to address the gaps persist with existing model.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In the previous chapter, meta-heuristic algorithms, neural network, ensemble model, and existing model were reviewed. The chapter also highlighted the research gap, the strength and limitation of existing meta-heuristic algorithms including Grey Wolf Optimizer (GWO).

Proceeding from the preceding chapter, this chapter portrays the research methodology applied for designing, implementing and assessing the ensemble approach consisting Modified Grey Wolf Optimizer (MGWO) and neural network. This chapter likewise depicts tuning of MGWO’s exploration to accomplish the optimal outcome. The chapter also illuminates the three research phases to fulfill the aims and objectives of the research namely k-means clustering for categorizing the stock data, classification to determine the best features applying meta-heuristic algorithm and neural network model to predict the stock price. In this chapter, GWO is modified and enhanced to better suit with stock data. The main focus here is to develop an ensemble model through the review and enhancement of algorithm such as clustering, classification and prediction. The chapter further demonstrates the neural network architecture, experimentations and validation of the constructed neural network. Data collection method and pre-processing for k-means clustering, neural network and MGWO is also elaborated in this chapter.

3.2 The Proposed Research

The overall research plan is presented in Figure 3.1. The first section of the figure represents the detail discussion about GWO and Modified GWO. Where, the basic characteristics of original GWO, mathematical model, exploration and exploitation through GWO will be discussed initially. Then, the modification of GWO, parameters of the algorithm, mathematical model, balance of exploration and exploitation through Modified GWO will be presented. The ensemble model applying neural network and Modified GWO is discussed in the next section, where data preprocessing and the feature selection will be explained as well. Next, application of ensemble model and evaluation will be presented. Finally, the performance evaluation of the ensemble model will be discussed.

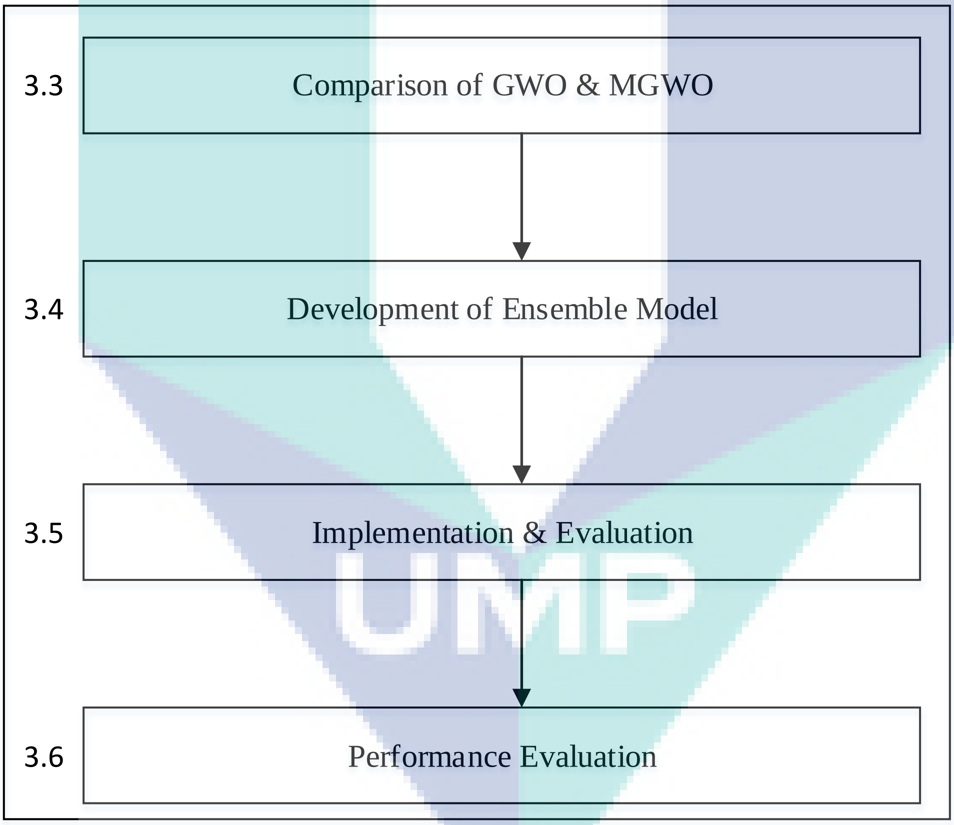


Figure 3.1 Overall Research Plan

Figure 3.2 portrays the research activities for design and application of Modified Grey Wolf Optimizer (MGWO) where, the MGWO is designed by studying the organization and movement of Grey Wolf from literature besides the development of Grey Wolf based mathematical model. The MGWO is a population-based meta-heuristic algorithm where the solution of a problem is produced through the movement of agents by balancing the exploration and exploitation to the search space. The proposed algorithm mimics the behavior of Grey Wolf in this research, where the algorithm is capable to balance the exploration and exploitation to the search space as it simulates the democratic and communicative behavior of Grey Wolf for solutions to their search.

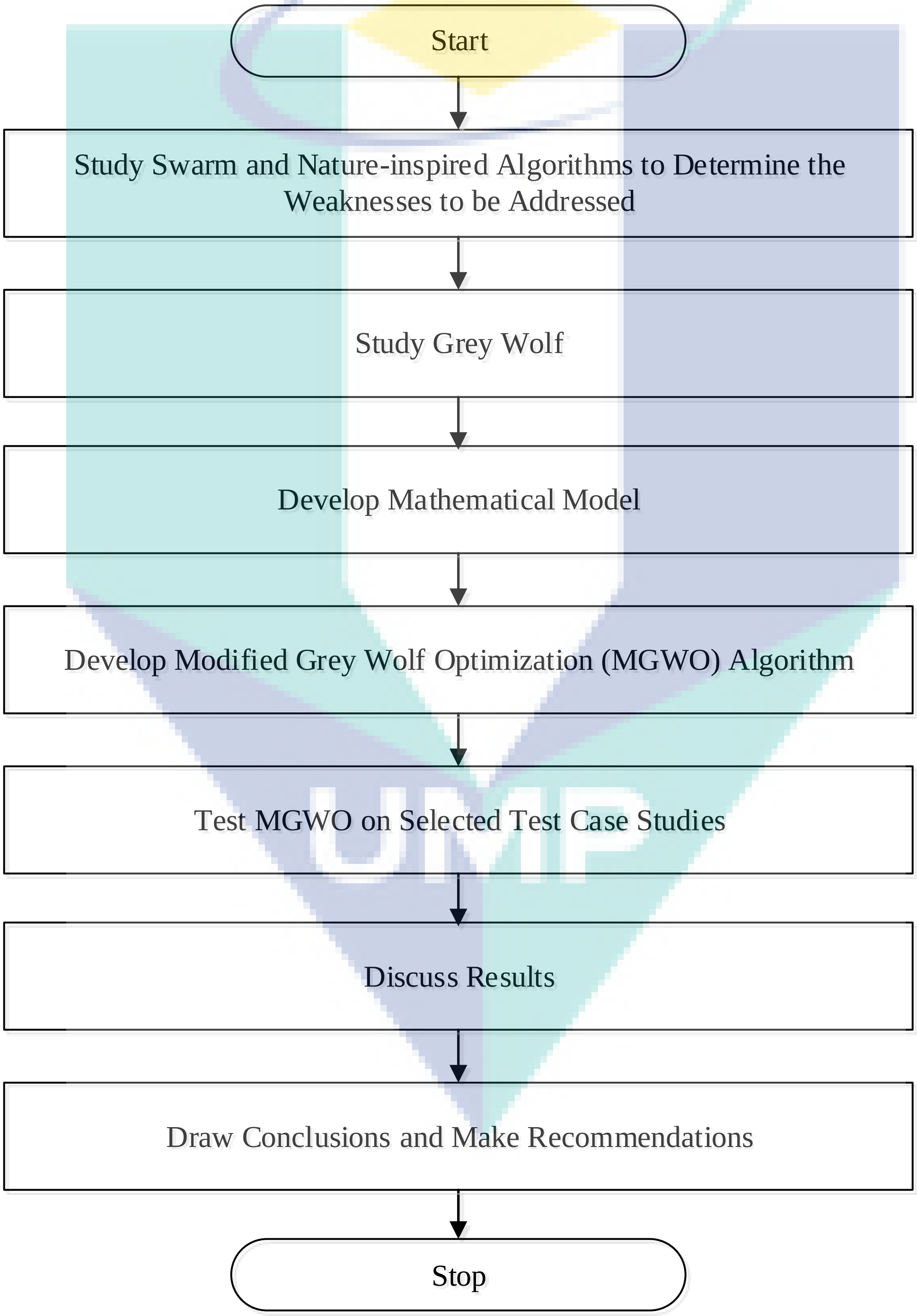

Start
Study Swarm and Nature-inspired Algorithms to Determine the Weaknesses to be Addressed
Study Grey Wolf
Develop Mathematical Model
Develop Modified Grey Wolf Optimization (MGWO) Algorithm
Test MGWO on Selected Test Case Studies
Discuss Results
Draw Conclusions and Make Recommendations
Stop

Figure 3.2 MGWO Research Activities Flowchart

3.3 Modified Grey Wolf Optimizer Algorithm

This section illustrates the design of proposed algorithm, MGWO. In the current research, MGWO is an approach based on GWO for feature selection and prediction. Original GWO starts by initializing the grey wolf population. Later, the algorithm updates the position of each search agent by the locations of the best solutions, and assessed the objective function of the algorithm.

The proposed algorithm improves the wolves attack strategy. It calculates the weights based on wolves fitness function and gives the highest weight to the dominant wolf concurrently to improve the convergence, decide the suitable thresholds faster, provide good classification rate, efficiency and accuracy. Table 3.1 describes the definition of each MGWO's parameter where, the parameter value for maximum iteration and population size are maintained with GWO and other algorithm for fair comparison. The value for coefficient vectors $\vec{a}$, $\vec{A}$ and $\vec{C}$ are reduced here in MGWO from original GWO algorithm in order to ensure closer exploration and exploitation. As the probability value lies between 0 and 1, the value for Probability vector is chosen as 0 to 1. Finally, the probability of generating 1 or 0 for each iteration is equal and hence, the Threshold value is chosen as 0.5 (Monteiro et al., 2018). The complete pseudo-code of MGWO algorithm can be represented in Figure 3.3.

Table 3.1 Details of MGWO's Parameters

Parameter	**Values**
Maximum number of iterations (*max_iter*)	300
Population size (*p_size*)	1000
$\vec{a}$	Linearly decreased from 1 to 0
$\vec{A}$	Random values in -a to a
$\vec{C}$	Random values in 0 to 1
Probability Vector (*P*)	0 to 1
Threshold	0.5
Archive (*A*)	Collected solution in each iteration

```
Begin
Initialize p_size, max_iter, n, pos, flag
Generate the init_pos of grey wolves randomly
Construct A archive of collected solutions in each iteration
Initialize $a$, $\vec{A}$ and $\vec{C}$
Initialize Controlling parameter of selecting/removing solutions from archive A
(Selecting ${G_\alpha}^n P_1=1/N_i$ , Removing ${G_\alpha}^n$ $P_2 = N_i$)
$G_\alpha$= The grey wolf with the first highest fitness
$G_\beta$= The grey wolf with the second highest fitness
$G_\delta$= The grey wolf with the third highest fitness
Threshold = 0.5
$P({G_\alpha}^n)$ = 1, the probability vector, ($\Pi = \{{G_\alpha}^1, {G_\alpha}^2, {G_\alpha}^3, \ldots\ldots\ldots, {G_\alpha}^n\}$)
While $i$<max_iter
  Calculate the fitness of grey wolves
  If fitness $i_{th} < G_\alpha$
    Update $G_\alpha$with new fitness $i_{th}$ value
    Update $G_\alpha$Position
  Else If fitness $i_{th} > G_\alpha$and fitness $i_{th} < G_\beta$
    Update $G_\beta$with new fitness $i_{th}$ value
    Update $G_\beta$Position
  Else If fitness $i_{th} > G_\alpha$and fitness $i_{th} > G_\beta$ and $i_{th} < G_\delta$
    Update $G_\delta$ with new fitness $i_{th}$ value
    Update $G_\delta$ Position
  End If
  For x = 1:p_size
    For y = 1:n
      If pos(x,y) > Threshold
        flag(y) = 1
      Else
        flag(y) = 0
      End If
    End For
  End For
Update the position of current grey wolf by, $\vec{G}(i+1) = (\vec{G}_1 + \vec{G}_2 + \vec{G}_3)/3$
Calculate the probability, $P({G_\alpha}^n) = \frac{N({G_\alpha}^n,\ A)}{\sum_{j=1}^{n} N\ ({G_\alpha}^j,\ A)}$
If $P({G_\alpha}^n)$<Threshold
  $v$=abs(Max $(G_\alpha) - fitness\_value)/fitness\_value$   // Normalization
  For z = 1: p_size
    Find non-dominated $P_1(G_\alpha)$< $v$
    Update the archive A, remove $G_{\alpha_{th}}$
    Re-Generate the init_pos of $G_{\alpha_{th}}$randomly
  End For
Else
  Update $a$, $\vec{A}$ and $\vec{C}$
  Calculate the fitness of grey wolves including selected features
  Update $G_\alpha, G_\beta,$ and $G_\delta$
  End If
  i = i + 1
End While
Return $G_\alpha$, selected features
End
```

Figure 3.3 Pseudocode of MGWO Algorithm

In psudo-code of MGWO illustrated as Figure 3.3, p_size represents the grey wolf population to initialize the MGWO parameters; *max_iter* denotes maximum number of iteration. The initial position of grey wolves are generated at this stage. The initialization of an archive (*A*) will be made to collect the solution for each iteration. Then, the initialization of the vectors *a*, *A*, *C* are performed through the Equation 3.1 and Equation 3.2.

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a} \qquad 3.1$$

$$\vec{C} = 2.\vec{r}_2 \qquad 3.2$$

The values of $\vec{a}$ are linearly decreased from 1 to 0 over the course of iterations as per the original algorithm. The initialization of the controlling parameter for selecting or removing solutions from archive (*A*) is made where, (Selecting ${G_\alpha}^n P_1=1/N_i$ and Removing ${G_\alpha}^n$ $P_2 = N_i$). Then, the estimation of the fitness for each search agents is made and best hunt agents are identified where, the best hunt agent is G_α, the second best hunt agent is G_β and the third best hunt agent is G_δ. Here, the initialization of the Threshold value is assigned to 0.5 as each of the steps may produce different values ranging between 0 and 1. The reason for choosing Threshold value 0.5 is that the probability of generating 1 or 0 for each iteration is equal for such value (Monteiro et al., 2018). Hence, the probability for all best search agent (G_α) is initialized to 1. Then, the location of the hunt agents are updated based on fitness. Here, the position of the agent is checked in comparison with threshold value 0.5, if the threshold value is more than 0.5, then feature will be selected by updating flag value to 1 otherwise feature will not be selected and updating flag value to 0. The location of the current hunt agent is updated using Equation, $\vec{G}(i+1) = (\vec{G}_1 + \vec{G}_2 + \vec{G}_3\ )\ /\ 3.$ The probability for all the best search agent is calculated applying the Equation, $P({G_\alpha}^n) = \frac{N({G_\alpha}^n,\ A)}{\sum_{j=1}^{n} N\ ({G_\alpha}^j,\ A)}$. Then, the probability of best search agent will be compared with threshold, if the probability value is below threshold then the non- dominated values will be removed from the archive and others will remain with archive for re-generation, otherwise the vectors *a*, $\vec{A}$ *and* $\vec{C}$ will be updated, the fitness value of all hunts including selected features will be estimated and the value of search agents G_α, G_β and G_δ will be updated. Hence, the stopping condition will be checked whether the iteration (*i*) reaches max number of iterations. Finally, the best value of solution G_α will be returned with selected features.

3.3.1 Exploration and Exploitation in MGWO

As the algorithm needs to address the exploration and exploitation, MGWO is tuned to balance the exploration and exploitation. To emphasize the exploration, the algorithm determines the new area in the problem search space through the application of rapid alteration in the solution so that the algorithm does not stack in local minimum. The exploitation is balanced by improving the accomplished expected solution in exploration by determining the neighborhood of every solution. In MGWO, $\vec{C}$ and $\vec{A}$ are the main controlling parameter to ensure exploration by returning a random values between 0 to 1 for $\vec{C}$ and 1 to 0 linearly decreased value of *a* for $\vec{A}$. Whereas, the original GWO, promotes the exploration through larger range of random values between 0 to 2 for $\vec{C}$ and 2 to 0 linearly decreased value of a for $\vec{A}$. The value of parameter $\vec{A}$ ranges between -1 to 1 where, exploration is achieved through the value A < 0 and exploitation is achieved through the value $A > 0$. The balance between exploration and exploitation is maintained through this algorithm by setting random values for $\vec{C}$ and linearly decreased values for $\vec{A}$. Moreover, original GWO searches for best G_α to emphasize the exploration and exploitation with higher ranges of values whereas, MGWO attempts to determine multiple G_α to explore and exploit with relatively shorter range of values. If unsuccessful with one G_α, the algorithm proceeds to find another G_α to further explore and exploit. This way, MGWO makes a good balance between the local and global solutions which also balances the exploration and exploitation.

3.3.2 Mathematical Model of MGWO

The mathematical model of MGWO can be formed as per the equations below:

$$\vec{D} = \left| \vec{C} \, . \vec{G}_a(i) - \vec{G}(i) \right| \tag{3.3}$$

$$\vec{G}(i+1) = \vec{G}_a(i) - \vec{A}.\vec{D} \tag{3.4}$$

Where, i denotes the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\vec{G}_a$ is the position vector of the prey, and $\vec{G}$ denotes the position vector of a grey wolf.

The vectors $\vec{A}$ and $\vec{C}$ are calculated as follows:

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a} \qquad 3.5$$

$$\vec{C} = 2.\vec{r}_2 \qquad 3.6$$

Where, components of $\vec{a}$ are linearly reduced from 1 to 0 over the number of iterations and is used for controlling the trade-off between exploitation and exploration. The following equations will be employed for updating the value of variable:

$$\vec{a} = 2 - i\,(2\,/\,X_i\,) \qquad 3.7$$

$$\vec{G}(i+1) = (\,\vec{G}_1 + \vec{G}_2 + \vec{G}_3\,)\,/\,3 \qquad 3.8$$

Where, X_i denotes the number of iterations, $\vec{r}_1$ and $\vec{r}_2$ are random vectors between [0, 1] which are employed to find the optimal solution. Appropriate idea about the potential location of prey can be availed by Alpha, Beta and Delta, where they help the Omega to follow the suitable positions. The values of $\vec{G}_1$, $\vec{G}_2$ and $\vec{G}_3$ can be obtained through the equations below:

$$\vec{G}_1 = \vec{G}_\alpha - \vec{A}_1.\vec{D}_\alpha \qquad 3.9$$

$$\vec{G}_2 = \vec{G}_\beta - \vec{A}_2.\vec{D}_\beta \qquad 3.10$$

$$\vec{G}_3 = \vec{G}_\delta - \vec{A}_3.\vec{D}_\delta \qquad 3.11$$

In iteration *i*, the best 3 solutions are respectively, $\vec{G}_1$, $\vec{G}_2$ and $\vec{G}_3$. Where, the values of $\vec{D}_\alpha, \vec{D}_\beta$ and $\vec{D}_\delta$ are as indicated in Equation 3.12, Equation 3.13 and Equation 3.14:

$$\vec{D}_\alpha = \left|\,\vec{C}_1.\vec{G}_\alpha - \vec{G}\,\right| \qquad 3.12$$

$$\vec{D}_\beta = \left|\,\vec{C}_2.\vec{G}_\beta - \vec{G}\,\right| \qquad 3.13$$

$$\vec{D}_{\delta} = \left| \vec{C}_3 . \vec{G}_{\delta} - \vec{G} \right| \qquad 3.14$$

Here, *n* random parameter vectors are formed to further explore the best solution for every iteration selected on to update vectors as per Equation 3.15.

Vectors, *Π* = $\{G_{\alpha}{}^{1}, G_{\alpha}{}^{2}, \ G_{\alpha}{}^{3}, \ldots\ldots\ldots\ldots, G_{\alpha}{}^{n}\}$ 3.15

If *N* is average distance between wolves, *A* is shared archived score and $\mathrm{G_{\alpha}}^{\mathrm{j}}$ is sum of all the best solution then the following probability Equation 3.16 can be formed as per meta-population distribution.

Probability, $P(G_{\alpha}{}^{n}) = \frac{N(G_{\alpha}{}^{n}, \ A)}{\sum_{j=1}^{n} N\,(G_{\alpha}{}^{j}, \ A)}$ 3.16

In MGWO, two important steps are included that comprises: firstly, select the features and train the neural network using MGWO to determine the optimal initial weights and secondly, test the results of the proposed MGWO approach. This approach can improve the efficiency of the back-propagation to seek global optima in the search space. For the proposed MGWO approach, the weights are achieved as a vector of variables.

For this approach, Root Mean Square Error (RMSE) is the cost function which can determine the error between the actual value and predicted value which can be expressed by Equation 3.17:

$$RMSE = \sqrt{\sum_{i=1}^{n}(a_i - p_i)^2} \qquad 3.17$$

Where, *n* is the number of observations, a_i is the number of actual values and p_i is the number of predicted values from neural network. The lower RMSE value is expected for determining acceptable prediction and making the model acceptable.

3.4 Stock Market Prediction

Stock market remains best investment alternatives for few decades despite being unpredictable and uncertain. The economy of a country can be greatly affected by stock market as it plays a significant role to the economy. Investors invest in the stock market to acquire the profit and for that they purchase the security bond of different company. The selection of the security bond of different company is made based on the different factors such as company's information analysis and prediction, and dividend declaration. If emerging stock market is considered as an example, it is observed that, most investors do not have adequate information about the market analysis and prediction of the future prices. Investors purchase the security bond based on rumor, manipulated financial report of companies and without any idea about data analysis and prediction. As a result, the stock market becomes unpredictable due to extreme ups and downs in the daily share price indices. Investors lose their capital in the unstable stock market which creates a big crisis in the capital market and national economic growth is greatly hampered due to such crisis. Therefore, a good model is required that will provide real scenario of stock market and facilitate the investors to predict the prices in advance. All these will contribute towards the solidity of the national economy.

Despite prediction of stock price is extremely complicated due to the nonlinear form of stock data, stock price prediction is an exciting research area. Consequently, researchers are continuously striving to improve the existing prediction models. Individual and institutional investors are not leaving even single efforts to make an accurate stock investment plan. They are devising their own strategy to perform the daily and future investment. However, due to the complex nature of stock data and stock market, stock price prediction still remains one of the most complicated jobs in financial forecasting (Wei, 2013). Investors are grabbing any forecasting method that assists them in making decent profit and minimize investment risk through stock investment. Consequently, it enables researchers' abundant motivation to either develop a new or enhance various stock prediction models (Atsalakis et al., 2011). Different types of prediction models have proved to be effective for stock market as the investors can avail the profit through those stock prediction models. Neural network is widely used by many researchers due to its ability to learn from unknown hidden patterns and capability to produce solution from unknown data. Some stock prediction works related to neural network are included here. ANN and ARIMA models are used for forecasting next day stock market by Merh et al. (2011). Future index value of Sensex (BSE 30) was also forecasted by them through those models to determine the forecasting accuracy. Mahajan et al. (2015) proposed a Neuro–Fuzzy model for BSE India which could guide investors to have profitable script in their portfolio. However, integration of multiple approaches are gaining priority currently instead of single approach in order to improve the stock price prediction model where distinct feature of each model is combined together to build a rigorous stock prediction model (Wang et al., 2012). Due to unpredictable behavior of stock market there is always some risk involved to the investment (Hassan et al., 2005). Moreover, stock prediction is even more complex due to influence of different factors: positive or negative news of the company, political turbulences, rate of interest, dollar price and natural disasters (Bonde et al., 2012).

Random Walk Hypothesis (RWH) believes that stock price is not affected by historical price and tomorrow's price is predictable through the analysis of today's price. The researchers who support RWH also established that stock prices cannot be predicted as they follow random behavior and it is unnecessary to apply fundamental analysis or machine learning for predicting stock market. Efficient Market Hypothesis (EMH) is another divisive model that also explains RWH states that stock price of a security is the reflection and determination of all relevant information. According to the EMH, buying stock is a game of chance and hence investors may not be able to analyze the information with better efficiency. In spite of controversy with EMH, researchers progress the stock prediction research forward through numerous research publications in this area (Tilakaratne et al., 2009). The RWH also suggests that stock data do not follow patterns, and is therefore not eligible for prediction. Extensive research on the topic implies the opposite that technical analysis can produce positive results in terms of prediction (Wong et al., 2012). Research on reward of technical analysis on the Singapore market suggests that a significant part of member companies rely on technical analysis (Wong et al., 2012).

The behavior of stock market may not be well known by financial analysts and eventually they will not be able to judge the exact time to buy or sell stocks for making profit through stock investment. However, decision making is a critical and vital process in stock trading as it has to be made correctly and at the right time (Gamil et al., 2007). Due to higher profit through stock investment, stock exchange is a prevalent investment destination though recent experience has demonstrated that the higher the expected return, the greater the risk consequences (Kuo et al., 2001 and Vincent et al., 2013). Thus, various studies have led to different decision support models in order to provide investors with optimal prediction. Internet plays vital role to make the huge stock information available to the investors, but the investor's tasks become quite tough due to various responsibilities such as collection, analysis, filtration and making correct decision from several information (Van et al., 2004).

Different stock prediction models have been developed over the years to understand, monitor and predict the stock market worldwide. The applications of various artificial intelligence based models to the stock market has drawn the researchers' attention apart from the statistical models that have been used to understand and predict fluctuations in the stock market. Many researchers have also focused on technical analysis as the procedure to improve the investment rate in stock market. Kozdraj (2009) attempted to apply neural network in predicting stock price for Warsaw Stock Exchange, Wu et al. (2012) applied neural network for the purpose of training multilayer perceptron network, Lopez et al. (2012) performed classification of data to build a model through placing similar data in a same group whereas disparate data is separated through clustering, Lertyingyod et al. (2016) proposed a stock prediction model through the analysis of historical price of stock and applied Data Mining techniques to predict one, five and ten day periods stock price trend, Narayanan et al. (2015) applied combination of classification model applying SVM and Naïve Bayes which gave them more accuracy with significant reduce of classification error, Navale et al. (2016) applied Artificial Neural Network (ANN) for prediction and Patel et al. (2015) applied SVM for stock price prediction. Therefore, understanding the market and being able to predict what will happen at the near future are desirable skills for every investor.

Each stock market adopts unique characteristics and the information gained from one perhaps implemented in another. Hence, the unique features of stock market needs to be studied through research. Despite neural network proved its potential in modeling non-linear relationship of stock market, adopting it for stock market is still challenging (Lertyingyod et al., 2016; Navale et al., 2016; Narayanan et al., 2015; Patel et al., 2015). The challenges through the neural network includes determining appropriate neural network architecture, the selection of representative input vectors of features from the time series data of the market and the availability of sufficient data for training. Stock market has adopted various algorithms for prediction over the years, i.e. GA, PSO, and SVM. The important question may raise as to, which model is the most effective for stock prediction?

Neural network and similar models are applied extensively for stock prediction by most studies. Extensive domain knowledge is the utmost priority in this regard to determine an appropriate choice of data and models for building neural network based stock prediction model. In spite of abundant investigations towards design and development of the standard model for stock prediction, no generic stock prediction model has been revealed yet (Lahmiri et al., 2015). Morepover, many investigators have developed several models in neural network to predict stock market, but most has failed to provide appropriate prediction due to various issues such as entrapment in local minima, result inaccuracy, slow convergence rate, uncertain and instable market situations. Researchers investigated various heuristic models over the years for training neural network to propose a good stock prediction model such as PSO, GA, ACO, ES and PBIL.

From the above discussions, it is clear that further investigations can be attempted to implement a good prediction model for stock market. Hence, the present study selects stock prediction as a case study for the implementation of ensemble model so that an effective stock prediction approach can be proposed that can predict average daily price of listed companies for stock market.

Figure 3.4 presents the block diagram of the activities for stock prediction where, k-means clustering algorithm is applied in the pre-processing block for selection of organization based on growth. However, the pre-processing block will be briefly explained in this research. The main contribution of the research is stock prediction and evaluation block and hence, the block will be explained in detail. The prediction of stock market applying ensemble model consists of the steps:

1. Select and preprocess datasets
2. Create a data mining based decision support model applying k-means clustering to categorize the organization based on growth
3. Create a classification based decision support model which can evaluate the suitability of data, select the features and train the neural network for prediction
4. Create an MLP neural network based decision support model to forecast the stock price
5. Create and benchmark a comparison strategy to evaluate the performance of the prediction model

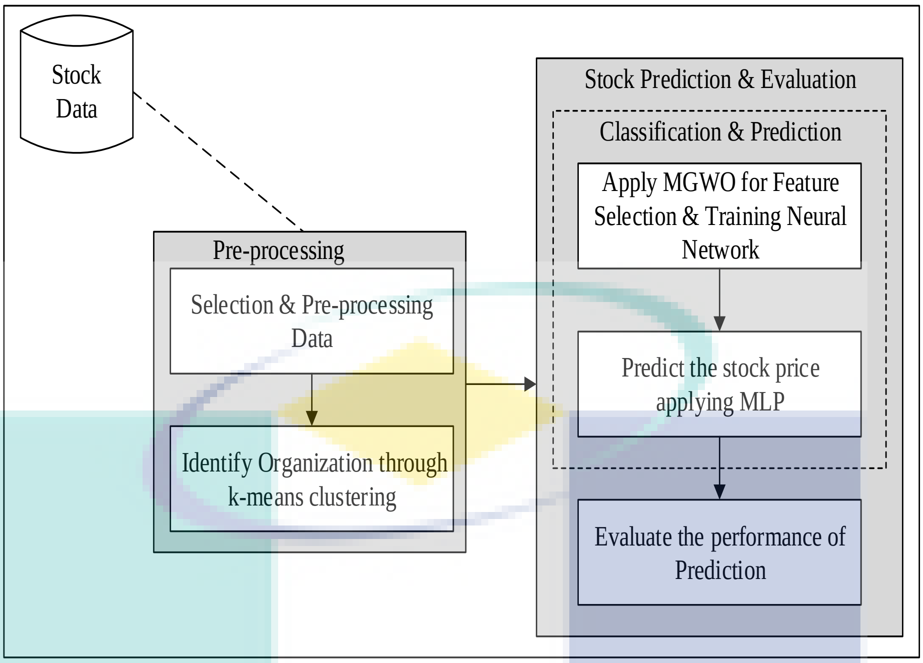

Stock
Data
Pre-processing
Selection & Pre-processing Data
Identify Organization through k-means clustering
Stock Prediction & Evaluation
Classification & Prediction
Apply MGWO for Feature Selection & Training Neural Network
Predict the stock price applying MLP
Evaluate the performance of Prediction

Figure 3.4 Block Diagram of Stock Prediction Activities

3.5 MGWO Implementation

MGWO consists of three basic operations such as initialization; calculation of fitness value, update accordingly and determine the best position of the grey wolf; and finally, validation of the terminating conditions.

3.5.1 Data Collection and Analysis

This study uses 6 years of end-of-day data beginning in January 2011. The research will apply the collected data to evaluate the proposed stock prediction ensemble model consisting of neural network and MGWO. The proposed model is evaluated in the context of the stock markets NYSE, NASDAQ, Bursa Malaysia, and DSE, Bangladesh. Therefore most of the analyzed data originates from the Yahoo Finance, Bursa Malaysia Library and DSE Library. The factors data is collected from Bangladesh Bank Website, Bangladesh Jewelry Samity and Banks in Bangladesh. In the following, the data source, preprocessing steps and arrangement of datasets are described in detail. All data is preprocessed and back adjusted as per requirement.

Table 3.2 shows typical examples of stock data set from DSE, Bangladesh, where data set belongs to one particular company of pharmaceutical sector named ACI for a certain date as mentioned in the table. Table 3.3 shows factors data set used for DSE, Bangladesh, where dataset belongs to the price for the factors such as gold price (per grams), dollar price (1 unit), bank interest rate, foreign direct investment (FDI) and inflation.

Table 3.2 Stock Data Set from DSE

DATE	TRADING CODE	LTP*	HIGH	LOW	OPENP*	CLOSEP*	YCP	TRADE	VALUE (mn)	VOLUME
1/1/2017	ACI	387	390	386.2	388	386.7	385.5	603	25.792	66,644
1/2/2017	ACI	403	404	390	390	402.1	386.7	1,431	79.082	199,196
1/3/2017	ACI	415.2	415.8	402	402	414.7	402.1	1,685	87.066	212,297
1/4/2017	ACI	413.5	422.9	412.8	418.6	413.5	414.7	1,088	62.533	150,348
1/5/2017	ACI	408.5	416	404	416	406.4	413.5	941	49.441	120,674
1/8/2017	ACI	407	414.4	402.8	414.4	408.8	406.4	816	46.914	115,185
1/9/2017	ACI	414	418.9	405.3	410	415.6	408.8	1,238	73.354	177,217
1/10/2017	ACI	432	432	416.1	418.3	430.1	415.6	1,825	111.914	264,037
1/11/2017	ACI	433	437.6	430	434.6	431.6	430.1	1,348	76.027	175,308
1/12/2017	ACI	424.1	432	421	431.6	423.3	431.6	950	46.643	109,565
1/15/2017	ACI	430	431	424.1	424.1	427.6	423.3	903	48.854	114,000
/16/2017	ACI	428.7	437	426.2	429.8	430.2	427.6	1,040	65.912	152,235
1/17/2017	ACI	427	431	426.5	430.9	427.7	430.2	903	48.841	113,897
1/18/2017	ACI	423.5	430	423	427.1	424.4	427.7	864	52.773	123,756
1/19/2017	ACI	421	428	417.1	423	421.5	424.4	817	57.738	136,874

Table 3.3 Factors Data Set Used for DSE

Gold Price	Dollar Price	Bank Interest Rate	FDI	Inflation
2905	69.25	8.50	861736237.16	9
2995	69.25	8.50	861736237.16	9
3095	69.24	8.50	1184776059.05	8
3190	69.24	10	1184776059.05	8
3290	69.24	10	1474542605.46	7
3190	69.27	10	1474542605.46	7
3260	69.26	12	1474542605.46	9
3340	69.25	12	1501647072.05	9
3430	69.26	12.50	1501647072.05	8
3455	69.26	12.50	1501647072.05	11
3535	69.25	12.50	1501647072.05	11

The data set is preprocessed to prepare it for experimentation. As observed in the Table 3.2 and Table 3.3, some numbers are very big in range. So, min-max conversion is applied to scale the data to same range as per the Equation 3.18 where, N_i is normalized data, y_i is original data for *i* = 1,2,3,….., *n* and *n* is total number of observations.

$$N_i = \frac{y_i - min(y)}{max(y) - min(y)} \qquad 3.18$$

3.5.2 K-Means Clustering Algorithm

Data with similar pattern can be placed into same group through a process called clustering, which is an unsupervised learning algorithm. Clustering can partition unlabeled data into similar groups. Analysis of data and retrieval of information is the core task of cluster analysis (Xu et al., 2005). The application of clustering algorithm in finance includes market segmentation, prediction of bankruptcy and scoring of credit. Hence, the clustering process can be applied extensively for the splitting of a large database into multiple clusters to discover the interesting pattern.

K-means, a simple unsupervised algorithm, capable of addressing the clustering problem (Wu et al., 2014). K-means is a partition algorithm which can handle a large database with multiple objects and this algorithm can generate the optimal cluster quicker. The algorithm runs simply by way of dataset classification and split the data into a number of fixed clusters (eg, *k* cluster) in advance. Identifying the *k* centers is the basic operation of this algorithm which needs one for every cluster and the centers are located in a complicated way so that different result is produced by dissimilar location. So, the clusters need to be placed in a more distant location for better acceptable result. The next phase of k-means joins the nearest center of the dataset for every point. However, early grouping is formed to finish the first phase if no underlying point is available. Next, the re-calculation of *k* new centroids is required as barycenter for the available clusters from previous phase. At this stage, *k* new centroids are available and a new binding is required between nearest center and data points underlying same dataset to produce a loop. The loop facilitates the alteration of the location of *k* centers step by step so that no further alterations are possible.

The collected data set from the stock market consists multiple organizations that needs to select one organization and k-means clustering algorithm is applied to the preprocessed data set to select one organization. The observation of data set produced through k-means clustering categorizes the organizations into fast growing and slow growing.

3.5.3 Input Features with Stock Data

Supervised learning method is chosen in this research, where the model is trained through a target attribute known as output. The output or target attribute is chosen as Investment Decision for all stock market, High price for NYSE, NASDAQ and Bursa Malaysia and Average price for DSE. The attributes collected from stock market and other factors for stock prediction are described as below:

Stock Number: The number which is provided to a company during the enlistment with the stock market is known as Stock Number. For an instance, British American Tobacco (BAT) receives stock number 4162 in Bursa Malaysia.

Stock Name: The name by which investor can recognize a company and also the name provided during enlistment is known as Stock Name. For an instance, British American Tobacco Malaysia is recognized as "British American Tobacco" in Bursa Malaysia.

Date: The trading day when the stock market performs the trading operation is known as Date. For an instance, NYSE is open on January 2, 2018 at 9.30 am local time.

Open Price: A security is first traded on a price for a particular trading day immediately after the opening of the stock market, which is known as Open Price. For an instance, NYSE opens at 9.30 am local time and each security is traded at that time on an open price. Daily opening price is the first trade price for a listed stock.

Closing Price: The closing price represents the final trading price of a security for a particular trading day. It denotes the most recent valuation of a security till the commencement of next trading day for a stock market.

High Price: High Price represents the highest trading price of a security for a particular trading day. Usually, High Price is higher than the open or closing price of that security in a stock market.

Low Price: The lowest trading price of a security at a given trading day is identified as Low Price. Generally, Low Price is lower than the open, closing and high price of that particular security.

Average Price: The average price of Open, Close, High and Low price of a security is known as Average Price for a security.

Volume: The number of shares traded for a security or whole stock market during certain period of time is known as Volume. In stock trading, there is a seller for every buyer and total volume is calculated through number of transactions. If sellers and buyers agree for a transaction at an agreed price for certain number of shares, it makes one transaction and volume is determined through the number of transactions multiplied by number of shares.

Number of Trade: The number of trade or transaction took place for a security on whole stock market during certain period of time is known as Number of Trade.

Turn Over: The amount being traded for a security on a whole stock market during certain period of time is known as Turn Over.

Gold Price: The price of one gram of gold for a particular day is considered as Gold Price. In this research, gold price is measured through local currency based on same stock market.

Dollar Price: The exchange price of a local currency against US Dollar (USD) is known as Dollar Price. In international market, different foreign currencies are traded in terms of number of units per USD.

Bank Interest Rate: The amount paid to deposit holders by bank or financial institutions are known as Bank Interest Rate. The Bank Interest Rate is expressed as percentage of principal on annual basis.

FDI: The amount of investment made for establishing business or acquiring business assets by an individual or a company of one country to another country is known as Foreign Direct Investment (FDI).

Inflation: The rate at which the price of goods and services is rising and the purchasing power of currency is falling is known as Inflation. Usually, central bank keeps track of Inflation rate to limit inflation and avoid deflation so that the economy runs smoothly.

3.5.4 Target or Output in Stock Data

In building predictive model, selection of target variable is one of the preliminary steps which is a simple and straightforward process. In this research, the target or output variables are high price or average price and the prediction of price is made for a day ahead as per the recommendation made by Xing et al. (2017). The decision (buy, sell or hold) is provided to the investor as an additional information to determine whether a stock is suitable for him or her to make investment. In predicting the high price or average price, the closer the value availed through prediction is the better for investor. The decision whether to buy, sell or hold the stock, is calculated through the consultation with expert in stock field and also through the observation of the stock movement.

The output high price or average price and stock investment decision obtained through the experimentation are included in result section. The data is arranged in such a way that the actual output is separated for high price or average price prediction while investment decision is placed at last. The resulting output from the neural network model is the predicted output for this research.

3.5.5 Data Partitioning

The successful predictive models can be built through feed-forward neural network which is a powerful neural network structure proficient in modeling prediction class from a non-linear predictor attributes combination. The over-fitting problem needs to be tackled though the network is able to fit accurate model from normalized data. The over-fitting is a situation when the network does not have capability to generalize between input-output patterns (Haykin et al., 2009). Before the feed-forward neural network was trained, the data used for training, testing and validating the network was divided using the dividerand commands of MATLAB illustrated through the Function 3.19. The command codes cycle samples between the training set, validation set, and test set according to percentages. Where, training set is used to determine the optimal set of connection weights, test set is used to determine the appropriate network configuration and validation set is needed to measure the generalization capability of the model. Maier et al. (2000) investigated through the review of previous researches that data can be divided in any percentage without considering statistical properties. However, it is difficult to evaluate the optimum result. Hence, the data set is distributed initially 70% of the samples to the training set, 15% to the validation set and 15% to test set because the validation and test set requires same percentage (Maier et al., 2000).

$$[\mathit{trainInput\_Data}, \mathit{valInput\_Data}, \mathit{testInput\_Data}, \mathit{trainInd}, \mathit{valInd}, \mathit{testInd}] = \mathit{dividerand}(\mathit{Input\_Data}) \qquad 3.19$$

3.5.6 Evaluation of the Predicted Stock Price

The performance measurement of the proposed model is calculated applying Mean Absolute Percentage Error (MAPE), Mean Absolute Deviation (MAD) and Root Mean Squared Error (RMSE). The Equation 3.20, 3.21, and 3.22 will calculate the MAPE, MAD and RMSE. Where, the actual values of the stock are, (a_1,a_2,a_3,……..,a_n) and the predicted value of the stock are, (p_1,p_2,p_3,……..,p_n).

$$MAPE = 100 * \frac{1}{n} \sum_{i-1}^{n} \left| \frac{a_i - p_i}{a_i} \right| \qquad 3.20$$

$$MAD = \frac{\sum_{i=1}^{n} |a_i - p_i|}{n} \qquad 3.21$$

$$RMSE = \sqrt{mean(a_i - p_i)^2} \qquad 3.22$$

The *MAE* closest to zero indicates that *MAE* is the better for ensemble model or else the closest the predictions to the actual value.

3.5.7 Implementation of MGWO Algorithm for Feature Selection

All the attributes from the collected dataset may not be significant for prediction and hence attributes selection is crucial in stock prediction research. Stock price may be influenced by numerous factors and predictors may encounter difficulties in selecting the input for experimentation. Witten et al., (2016) have suggested to select the attributes based on deep understanding of the learning problems at hand and real meaning of the available attributes (Witten et al., 2016). Numerous methods are available for feature selection, whereas meta-heuristic algorithm can be a better option for feature selection which is also supported by Emary et al. (2016). In this research, the approach for MGWO are employed in the feature selection domain for finding feature subset maximizing the classification accuracy while minimizing the number of selected features. Wrapper's approach for feature subset selection suggests three main processes (Emary et al., 2016):

1. Classification method.

2. Feature evaluation criteria.

3. Search method.

In order to represent the population of each particular entry of MGWO inspected data, decision variables are defined to represent individuals, which are neural network parameters and inserted feature. This process is performed during each iteration of analysis. In other words, the algorithm inspects one feature in every iteration and provides an index value to rank that feature for further extraction process. Steps of MGWO implementation for feature selection are as indicated in Figure 3.5. The Equation 3.23 presents the cost value, cost of seed parameters of neural network used in each classification cycle ω as well as the number of inserted features (which is in our case 16 features). During inspection of each feature, the feature k will be checked against the classification rate F_k as a tagged value $\geq$ 0.5 which will determine how accurate is the classifier given the selected feature set, then the feature will be selected in indicating selected feature k, otherwise it will be excluded from the list of final features.

$$z\colon F_k = [Cost\ \omega\ f_1\ f_2\ \ldots\ f_n] \qquad 3.23$$

Each search agent calculates its fitness value upon selecting a sub-set of features F_k and gets compared against the index threshold value during training and testing processes. By applying this process during all iterations, eventually less competent features would be extracted as they have produced less impact on the obtained fitness value while the dominantly high indexed features would be kept in the final extracted list.

In this research, the selected features are confirmed through the consultation with three domain experts, who are the manager in securities division of bank related to DSE. Domain experts can guide well in selecting appropriate attributes for prediction (Suh, 2012) and the information collected through the expert views for selecting the attributes of stock prediction are really useful in this regard.

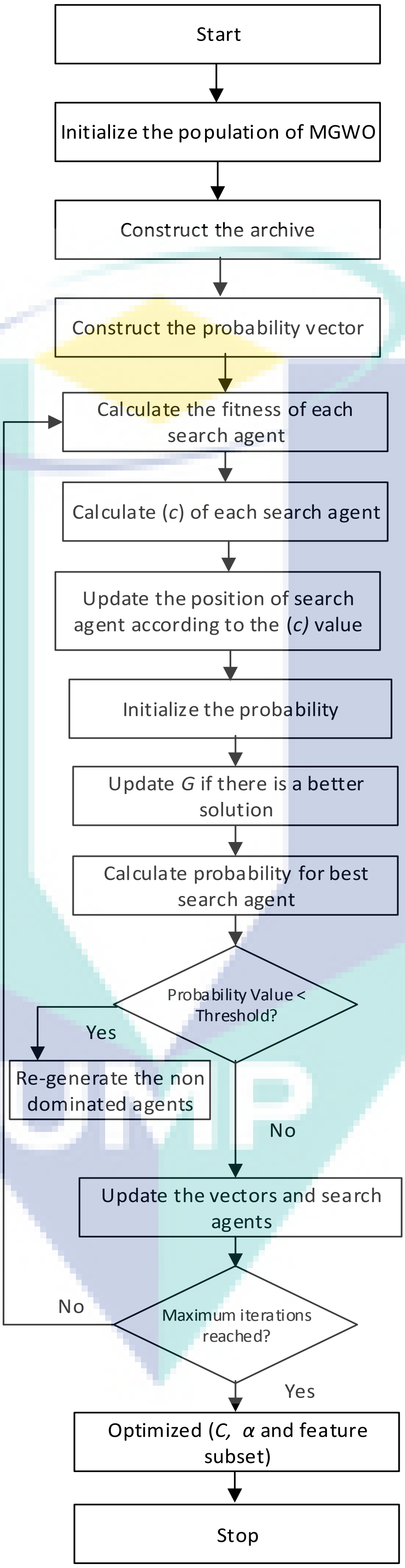

Start
Initialize the population of MGWO
Construct the archive
Construct the probability vector
Calculate the fitness of each search agent
Calculate (c) of each search agent
Update the position of search agent according to the (c) value
Initialize the probability
Update G if there is a better solution
Calculate probability for best search agent
Probability Value < Threshold?
Yes
Re-generate the non dominated agents
No
Update the vectors and search agents
No
Maximum iterations reached?
Yes
Optimized (C, α and feature subset)
Stop

Figure 3.5 Steps of MGWO Implementation for Feature Selection

3.5.8 Neural Network Model Design

In the current research, feed-forward multilayer perceptron architecture as illustrated in Figure 3.6 is used for designing the neural network model. The data is divided as training, validation and testing as explained in Section 3.5.4. MGWO algorithm is used for training neural network to optimize MLP parameters. The Mean Absolute Percentage Error (MAPE), Mean Absolute Deviation (MAD) and Root Mean Squared Error (RMSE) are used for performance evaluations which are objective function as well. Error is computed during each iteration of training so that error can be tracked easily and rising of error will terminate the iteration. Increasing of error indicates that the training process has converged.

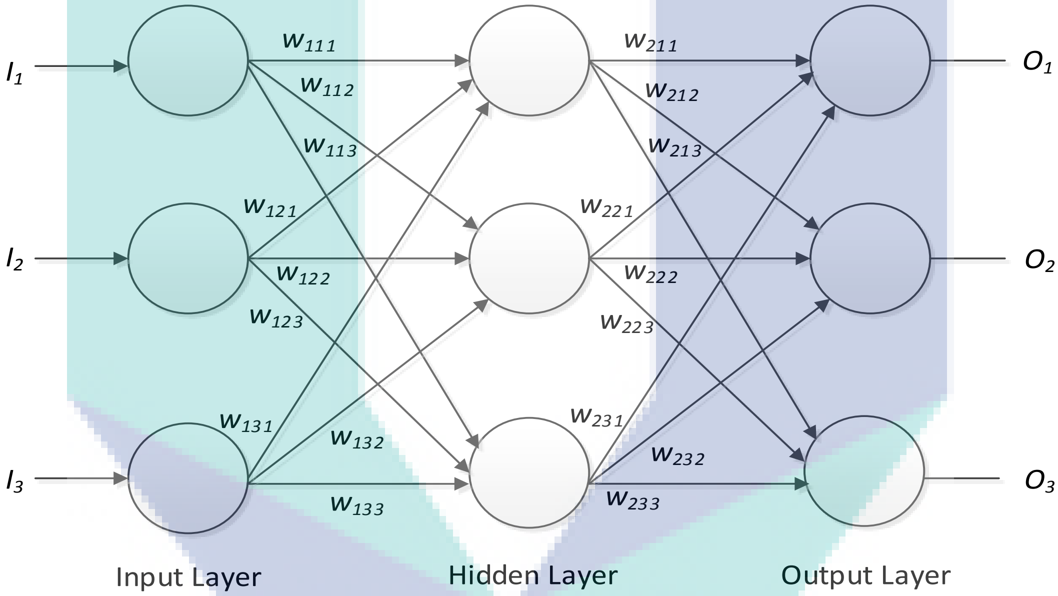


Figure 3.6 Multi-Layer Perceptron Architecture

In the Figure 3.6, the input layer is represented by I_1, I_2 and I_3 inputs, next the hidden layer which is not visible to users and it contains number of nodes, accepts value from preceding layer, executes mathematical operation on those values and pass it to the next layer which is the output layer that produces O_1, O_2 and O_3 outputs. In this architecture, each node calculates the sum of the value received from preceding nodes, performs the validation checking against a threshold value and produces the outputs by multiplying with layer weights W_{111}, W_{112}, W_{113}, W_{121}, W_{122}, W_{123}, W_{131}, W_{132}, W_{133}, W_{211}, W_{212}, W_{213}, W_{221}, W_{222}, W_{223}, W_{231}, W_{232}, W_{233} iteratively till the network converges and eventually the error starts rising at this point. The activation function or transfer function represents the output where a function that increases the values to balance linear and nonlinear nature called as sigmoid function which is used for the whole network. If Θ_m is the activation function for the output of a neuron in terms of the induced local field a and s is the slope of the function then the sigmoid function is represented by the following Equation 3.24.

$$\Theta_m = \frac{1}{1+exp(-sa)} \tag{3.24}$$

To avoid the over-fitting problem in this research, numbers of neurons are limited to 10 and the error is set to 0. Whereas, the maximum fails is fixed to 10 which facilitates the network to converge, if unable to meet up with other settings after a trial of 10 times. The iteration is set to 200, though the network may converge within a few numbers of epochs, if the configuration settings are fulfilled. The performance of neural network model is improved by training it using MGWO algorithm.

3.5.9 Ensemble of MGWO and Neural Network Algorithm

Figure 3.7 illustrates the ensemble of MGWO algorithm and neural network for feature selection and training Multi- Layer Perceptron neural network. The selected feature is forwarded to the MLP neural network that will be used to produce the best trading result through the learning of MGWO. Here, the predicted trend is evaluated to measure the performance of the prediction. In ensemble algorithm, the MGWO optimization process helps to determine the best feature combinations as indicated in section 3.5.7 and then facilitates the MLP neural network to select the features so that the network can determine the best set of features for prediction.

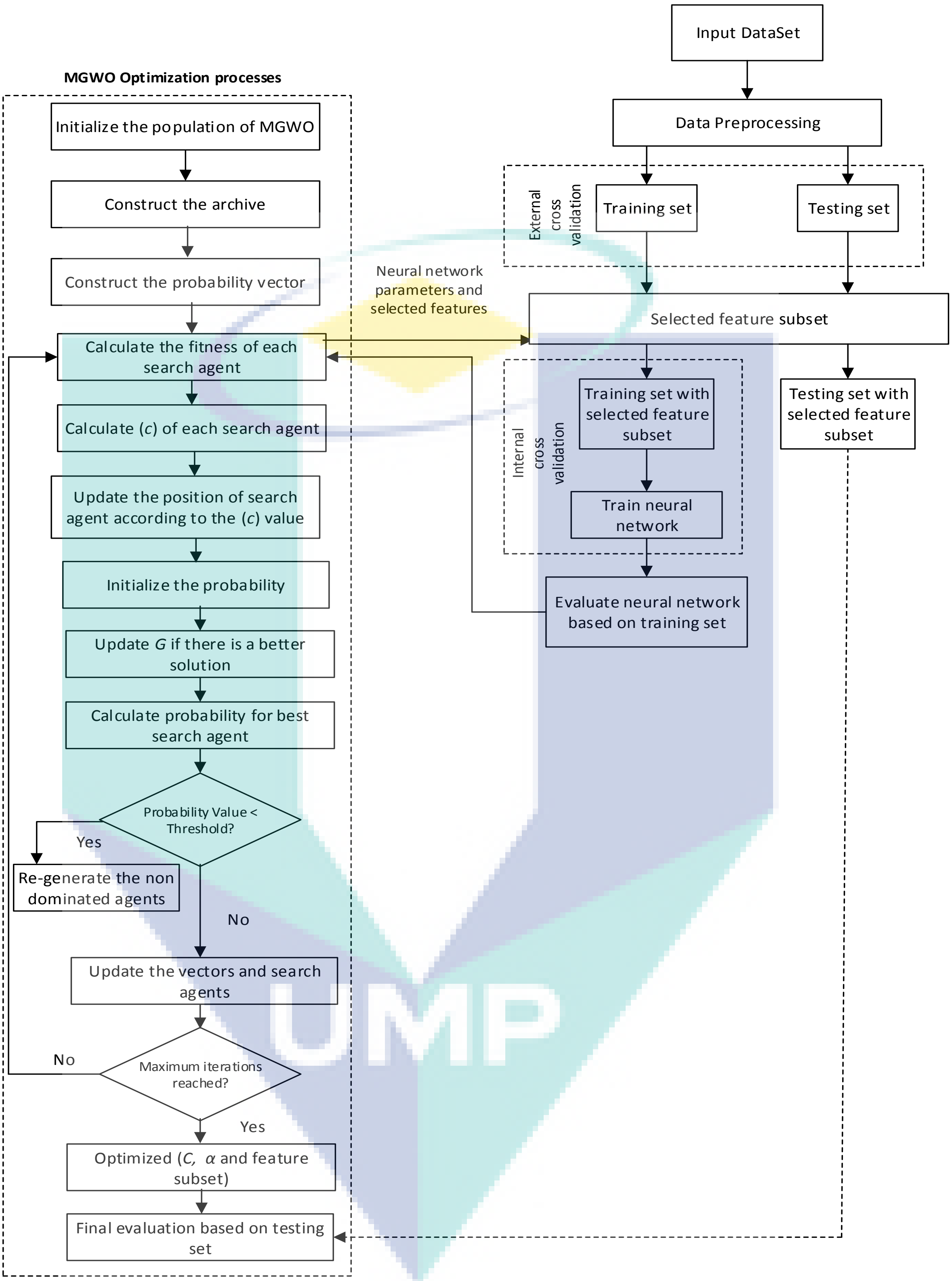

MGWO Optimization processes
Initialize the population of MGWO
Construct the archive
Construct the probability vector
Calculate the fitness of each search agent
Calculate (c) of each search agent
Update the position of search agent according to the (c) value
Initialize the probability
Update G if there is a better solution
Calculate probability for best search agent
Probability Value < Threshold?
Yes
Re-generate the non dominated agents
No
Update the vectors and search agents
No
Maximum iterations reached?
Yes
Optimized (C, α and feature subset)
Final evaluation based on testing set
Input DataSet
Data Preprocessing
External cross validation
Training set
Testing set
Neural network parameters and selected features
Selected feature subset
Internal cross validation
Training set with selected feature subset
Train neural network
Evaluate neural network based on training set
Testing set with selected feature subset

Figure 3.7 Classification through MGWO

3.5.10 Implementation of Ensemble Algorithm for Stock Prediction

In ensemble of neural network and MGWO, the first phase involves the selection and preprocessing of the datasets. It includes conversion of the stock data into continuous series and selection of the data for related factors such as bank interest rate, FDI, and gold rate. Besides, data from different information sources is converged into one substantial dataset, so all required data is accessible for performing simulation.

A k-means cluster data mining based decision support model is included so that the companies of the stock market can be categorized into two ways such as: high growth and low growth.

The classification of stock data is performed applying MGWO to verify how well they are compatible for prediction, select features and learning. The learning algorithm is determined at this stage.

The prediction of the stock price using Non-linear Autoregressive Exogenous neural network algorithm is performed that includes the creation of a neural network based decision support model so that the predicted stock price can be availed for both high growth and low growth organization. Finally, the predicted stock price is compared to evaluate the performance of the prediction.

After passing through the several steps mentioned above a conclusion regarding the proposed research questions would be possible to make. In particular, assessment of whether ensemble of neural network and MGWO model is suitable decision support model in the domain of stock prediction or not. Figure 3.8 illustrates the concept of ensemble of neural network and MGWO, where each technique complements one another to contribute for acceptable stock prediction model.

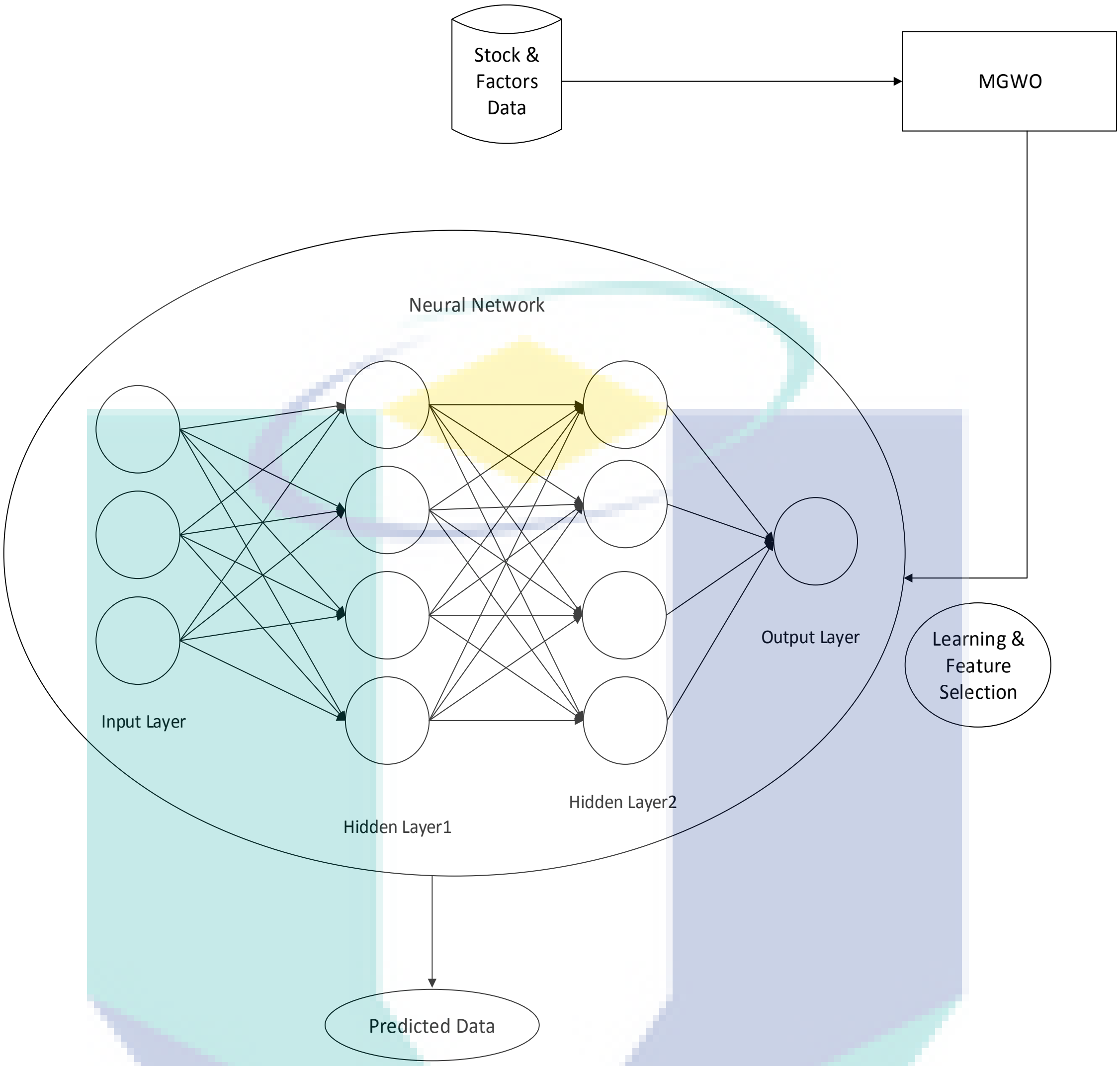


Figure 3.8 Ensemble of Neural Network and MGWO for Stock Prediction

The research applies the ensemble model consisting of neural network and MGWO to alleviate the limitation of each technique. The limitation of neural network includes difficulty in training (Srivastava et al., 2015), entrapment in local minima, result accuracy and convergence rate (Mirjalili et al., 2014a) while standard Grey Wolf Optimizer algorithm has drawbacks of low solving precision, slow convergence, and bad local searching ability (Yang et al., 2017).

The proposed ensemble algorithm is explained in Figure 3.9 which can be used for the stock prediction. The algorithm has 3 parts where Part (a) represents k-means clustering to determine the stock to be chosen for prediction and stock investment decision i.e. buy, sell or hold, Part (b) represents feature selection, learning, and classification, and Part (c) represents, stock prediction applying MLP algorithm and the evaluation of prediction result. The input of the algorithm needs the pre-processing of the data and output is the predicted stock price, Buy/Sell/Hold Decision, effect of various factors on Stock price and the evaluation result. The algorithm starts with applying k-means clustering to select the stock for prediction. Then, the algorithm will use the rule to make the decision of whether to Buy/Sell/Hold the stock. Next, MGWO algorithm will be applied for the training of neural network through the selection of the best features, determine the suitability of stock data for prediction and learning. Later, MLP neural network is applied to predict the stock price, various factors are integrated with stock data to determine the effect of various factors on stock price. Finally, the prediction result will be evaluated applying the Equation 3.20, Equation 3.21 and Equation 3.22.

Ensemble of neural network and MGWO for stock prediction – Part (a)

```
**********K-means clustering to determine the stock to be chosen for prediction**********
Begin
Compute the distance of each data-point di (1<=i<=n) to all the centroids
$c_j$ (1<=j<=k) as d ($d_i$, $c_j$)
 For each data-point $d_i$,
   Find the closest centroid $c_j$ and assign $d_i$ to cluster j
 End For
Set ClusterId[i] = j        // j:Id of the closest cluster
Set Nearest_Dist[i] = d($d_i$, $c_j$)
For each cluster j (1<=j<=k), recalculate the centroids
  Repeat
End For
For each data-point $d_i$,
   Compute its distance from the centroid of the present nearest cluster
   If this distance is less than or equal to the present nearest distance, the data-point stays in the cluster
   Else
      For every centroid $c_j$ (1<=j<=k)
          Compute the distance d($d_i$, $c_j$);
      End For
      Assign the data-point $d_i$ to the cluster with the nearest centroid $c_j$
      Set  ClusterId[i]=j
      Set Nearest_Dist[i]= d($d_i$, $c_j$)
End For
For each cluster j (1<=j<=k), recalculate the centroids
       Until the convergence criteria is met
End For
********** Determine the Buy/Sell/Hold Decision through the Stock Data **************
If Openprice2 < Openprice1, Highprice2<Highprice1, Lowprice2<Lowprice1,
     Closeprice2<Closeprice1 Then
        Decision = 2 (BUY)
Else If Openprice2 > Openprice1, Highprice2>Highprice1, Lowprice2>Lowprice1,
     Closeprice2>Closeprice1 Then
        Decision = 3 (SELL)
Else
        Decision = 1 (HOLD);
End If
Return (Decision)
```

Figure 3.9(a) K-means Clustering and Determining Buy/Sell/Hold Decision

Ensemble of neural network and MGWO for stock prediction – Part (b)

```
*********** Feature Selection, Learning and Classification of the stock data through MGWO ********
Initialize p_size, max_iter, n, pos, flag
Generate the init_pos of grey wolves randomly
Construct A archive of collected solutions in each iteration
Initialize $a$, $\vec{A}$ and $\vec{C}$
Initialize Controlling parameter of selecting/removing solutions from archive A (Selecting $G_\alpha{}^n P_1=1/N_i$
    Removing $G_\alpha{}^n P_2 = N_i$)
$G_\alpha$ = The grey wolf with the first highest fitness
$G_\beta$ = The grey wolf with the second highest fitness
$G_\delta$ = The grey wolf with the third highest fitness
Threshold=0.5
$P(G_\alpha{}^n) = 1$, the probability vector, $(\Pi = \{G_\alpha{}^1, G_\alpha{}^2, G_\alpha{}^3, \ldots\ldots\ldots, G_\alpha{}^n\})$
While $i$<max_iter
  Calculate the fitness of grey wolves
  If fitness $i_{th} < G_\alpha$
      Update $G_\alpha$ with new fitness $i_{th}$ value
      Update $G_\alpha$ Position
  Else If fitness $i_{th} > G_\alpha$ & fitness $i_{th} < G_\beta$
      Update $G_\beta$ with new fitness $i_{th}$ value
      Update $G_\beta$ Position
  Else If fitness $i_{th} > G_\alpha$ & fitness $i_{th} > G_\beta$ & $<G_\delta$
      Update $G_\delta$ with new fitness $i_{th}$ value
      Update $G_\delta$ Position
 End If
 For x = 1:p_size
   For y = 1:n
      If pos(x,y) > 0.5
         flag(y) = 1
      Else
         flag(y) = 0
      End If
   End For
 End For
Update the position of current grey wolf by, $\vec{G}(i+1) = (\vec{G}_1 + \vec{G}_2 + \vec{G}_3)/3$
Calculate the probability, $P(G_\alpha{}^n) = \frac{N(G_\alpha{}^n,\ A)}{\sum_{j=1}^{n} N(G_\alpha{}^j,\ A)}$
If $P(G_\alpha{}^n)$<Threshold
  v=abs(Max $(G_\alpha) - fitness\_value)/fitness\_value$   %Normalization
  For z = 1: p_size
     Find non-dominated $P_1(G_\alpha) < v$
    Update the archive A, remove $G_{\alpha_{th}}$
   Re-Generate the init_pos of $G_{\alpha_{th}}$ randomly
 End For
Else
  Update $a$, $\vec{A}$ and $\vec{C}$
  Calculate the fitness of grey wolves including selected features
  Update $G_\alpha$, $G_\beta$, and $G_\delta$
End If
i = i + 1
End While
Return $G_\alpha$, selected features
```

Figure 3.9(b)   Feature Selection, Clasification and Learning through MGWO

Ensemble of neural network and MGWO for stock prediction – Part (c)

```
*********  Predict the Stock Price applying MLP neural network algorithm *************
Partition the Stock Data,
    [A,P] = Stock_Data
    Anew = A(i : k)
    A = 1 : m
    P = 1 : m
Train the network using MGWO,
        net = narxnet(1:2,1:2,10)
[As,Ai,Pi,Ps] = preparets(net,A,{},P);
net = train(net,As,Ps,Ai,Pi);
view(net)
Calculate the network performance,
        [N,Af,Pf] = net(As,Ai,Pi)
perf = perform(net,As,N)
Run the prediction for necessary time steps ahead in closed loop mode,
        [netc,Aic,Pic] = closeloop(net,Af,Pf)
view(netc)
********** Evaluate the Effect of Factors on Stock Price through Regression **************
********* The equation used for measuring the effect of factors on stock price,

 $D = p + qI$
      Where, $D$ = dependent variable, $I$ = independent variable, $q$ is the slope of the line and
           $p$ is the M-intercept. *********************************

Calculate, $q = \frac{n*Sum(d*i)-Sum(i)*Sum(d)}{n*Sum(d^2)-(Sum(d))^2}$
Insert the values into the equation, $D = p + qI$

************************* Evaluate the prediction result ***************************

Calculate,   $MAPE = 100 * \frac{1}{n} * (Sum(ABS(\frac{a_i-p_i}{a_i}))$
             $MAD = \frac{Sum(ABS(a_i-p_i))}{n}$
             $RMSE = SQR(Mean((a_i - p_i)^2))$
End
```

Figure 3.9(c) Predict the Stock Price, Evaluate the Factors and Evaluate the Result

Figure 3.9 Proposed Ensemble Algorithm for Stock Prediction

3.6 Computational Complexity of MGWO

The computational complexity of the MGWO depends on the number of generation (g), the population number (n), and the parameters dimensions (d). Therefore the overall computational complexity is O(MGWO) = O(Initialization) + g(O(Calculate the fitness of wolves) + O(Calculate the population in the archive) + O (Sort the population and archive population) + O(Select n best grey wolves from the population and archive population) + O(Update the population)). The computational complexity of initialization is $O(nd)$, the computational complexity of calculating the archive population and fitness is $O(n)$, the computational complexity of sorting the population and archive population is $O(2n \log 2n)$, the computational complexity of selecting n best grey wolves from the population and opposition population is $O(n)$, the computational complexity of updating the population is $O(nd)$. Therefore, the final computational complexity is $O(MGWO) = O(nd) + g(2O(n) + O(2n \log 2n) + O(nd)))$.

3.7 Chapter Summary

This chapter describes the methodology engaged in this thesis related to the research questions. In particular, the flow and implementation of Modified Grey Wolf Optimization algorithm. The chapter also explains the overview of stock prediction, process of stock categorization using k-means clustering data mining algorithm, classification using MGWO and ensemble of MLP neural network. The analysis showed that neural network is trained applying the MGWO for feature selection. The feature selection and training the neural network through the selection of features is illustrated with appropriate flowchart in this chapter. The MGWO is explained with the algorithm. The investigation involving ensemble algorithm of neural network and MGWO for stock prediction with the evaluation of the model is illustrated also in this chapter. Finally, the computational complexity of MGWO is discussed.

The next chapter will demonstrate the performance of ensemble model applying neural network and MGWO. The performance of ensemble model will be evaluated against the existing approaches along with the statistical analysis.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

In the previous chapter, the ensemble model design and implementation were illustrated and elaborated. Specifically, the necessary adaptation of GWO to the stock prediction problem has been highlighted. Then, the tuning process for MGWO has been described, that is, by introducing the archive to evaluate which solution provides better probability to proceed further for training and regeneration. Finally, based on process design of MGWO, the implementation of MGWO has been elaborated.

This chapter explains and presents the results obtained at various stages of this research. The research is performed in three stages and hence the results obtained from all the stages are included here with detail explanation. This chapter precisely illustrates the clustering through the k-means, classification through the MGWO and comparison of classification using other methods i.e. PSO, GA, ACO, ES, PBIL, BBO and stock prediction through MLP neural network. The validation of the result and the error associated with each model are also discussed in this chapter. Finally, the comparative benchmarking experiments with well-known strategies that are presented along with the necessary statistical analysis. The stock prediction approaches reinforced through literature review and the key findings related to the approach are demonstrated in detail here.

4.2 Experimentations

The experiments contain three relevant goals. Firstly, k-means clustering algorithm is applied to produce two clusters and benchmarking of MGWO against other existing meta-heuristic algorithms are performed. Secondly, stock prediction through MLP neural network is made. Finally, findings are verified using statistical analysis.

The comparisons are made as per the well-known benchmarks applied by prior researches (Emary et al., 2018; Eswaramoorthy et al., 2016; Emary et al., 2016; Cai et al., 2015; Delnavaz, 2014; Mirjalili et al., 2014b).

The approaches described in this thesis has been implemented using the Weka Data Mining software and MATLAB. The results are presented in several tables and graphs. Tables 4.2 through 4.15 show the results obtained in the experiments. For fair comparison, Table 4.1 shows the parameters that are adopted for the existing meta-heuristic algorithms such as GA, ACO, PSO, PBIL, ES, BBO, GWO.

Table 4.1 Parameters for Existing Meta-Heuristic Algorithms

Algorithm	Parameter	Values
GA	Maximum number of generations	300
	Population size	1000
	Type	Real coded
	Selection	Roulette wheel
	Crossover	Single point (probability = 1)
	Mutation	Uniform (probability = 0.01)
ACO	Maximum number of iterations	300
	Population size	1000
	Initial pheromone (τ_0)	1e-06
	Pheromone update constant (Q)	20
	Pheromone constant (q_0)	1
	Global pheromone decay rate (p_g)	0.9
	Local pheromone decay rate (p_t)	0.5
	Pheromone sensitivity (α)	1
	Visibility sensitivity (β)	5
PSO	Maximum number of iterations	300
	Population size	1000
	Topology	Fully connected
	Cognitive constant (C_1)	1
	Social constant (C_2)	1
	Inertia constant (w)	0.3

Table 4.1 Continue

ES	Maximum number of iterations	300
	Population size	1000
	λ	10
	δ	1
PBIL	Maximum number of iterations	300
	Population size	1000
	Learning rate	0.05
	Good population member	1
	Bad population member	0
	Elitism parameter	1
	Mutational probability	0.1
BBO	Maximum number of iterations	300
	Population size	1000
	Habitat modification probability	1
	Immigration probability bounds per gene	[0,1]
	Step size for numerical integration of probabilities	1
	Max immigration (*I*) and Max emigration (*E*)	1
	Mutation probability	0.005
GWO	Maximum number of iterations	300
	Population size	1000
	$\vec{a}$	Linearly decreased from 2 to 0
	$\vec{A}$	Random values in -2a to 2a
	$\vec{C}$	Random values in 0 to 2

4.2.1 Result through K-means Clustering

In this research, k-means clustering algorithm is applied to produce two clusters. Based on Volume of Trades for a company, one cluster will contain fast growing companies and other will contain slow growing companies. The stock data used for experimentation is as per the sample shown in Table 3.2. Here, the tables 4.2, 4.3 and 4.4 demonstrate the k-means clustering outcome where, the stock with highest number of volume traded is placed in one cluster and the stock with average volume traded is placed in another cluster through k-means clustering algorithm.

Table 4.2 K-means Clustering Output for DSE

Attribute	Full Data (17092)	Cluster # 0 (12091)	Cluster # 1 (5001)
Company Name	ACI	ACI	SQURPHARMA
LOWPRC	1309.27	280.95	3795.47
HIPRC	1348.31	293.25	3899.15
AVGPRC	1330.63	288.11	3851.13
CLSPRC	1330.63	287.91	3851.63
TRDVOL	110681.94	147207.60	22373.25

Table 4.3 K-means Clustering Output for Bursa Malaysia (KLCI)

Attribute	Full Data (7375)	Cluster # 0 (1475)	Cluster # 1 (5900)
Company Name	BATM	BATM	Digi
LOWPRC	15.7035	57.4682	5.2624
HIPRC	15.843	58.0141	5.3002
OPNPRC	15.5668	56.9263	5.227
CLSPRC	15.7002	57.4475	5.2634
TRDVOL	6646949.6052	182207.7458	8263135.07

Table 4.4 K-means Clustering Output for NASDAQ

Attribute	**Full Data (7608)**	**Cluster # 0 (5732)**	**Cluster # 1 (1876)**
Company Name	Microsoft	SPARTAN	Microsoft
LOWPRC	14.9738	6.7926	39.9707
HIPRC	15.3503	7.0333	40.7623
OPNPRC	15.1611	6.9147	40.3574
CLSPRC	15.1708	6.914	40.3989
TRDVOL	15529463.775	8414224.1626	37269630.8635

It has been observed that ACI from DSE, Digi from Bursa Malaysia and Microsoft from NASDAQ have highest volume traded overall and hence those can be selected for stock prediction through Classification and MLP neural network as the investors have the higher possibility of gain through these stocks.

4.2.2 Feature Selection Applying MGWO

As explained in section 3.4.7, decreasing number of features to select key contributing features for better prediction is the main purpose of feature selection. In current research, MGWO with wrapper approach is applied for feature selection to provide better classification, faster convergence and avoid overfitting. Consequently, number of features are reduced to 6 for prediction from total number of 16 selected features that have been gathered originally. In the experimentation of feature selection, set of features have been provided for classification through MGWO initially and it produced different classification rate such as 80%, 70%, 60% and so on. As per the classification rate produced, a feature selection vector has been formed with the value 0 to 1 for each iteration. Finally, the set of features with best value has been chosen. This approach produced the best set of features as indicated in Table 4.5. Where, 1 represents “Hold”, 2 represents “Buy” and 3 represents “Sell”. The “Decision” column with stock data has been incorporated by forming the rules in stock market investment.

Table 4.5 Best Set of Features through MGWO

Instrument	Low Price	High Price	Open Price	Closing Price	Decision
IBM, NYSE	149.61	149.63	150.11	148.58	1 (Hold)
	150.26	149.25	151.95	149.22	1
	149.06	149.35	149.99	148.12	1
	149.07	148.25	149.6	148	1
	149.9	150.02	150.15	147.81	1
	151.43	150	151.6	149.65	1
	152.34	152.07	153.52	151.91	3 (Sell)
	150.51	152.52	152.96	150.25	1
	149.79	151.45	153.1	149.36	1
	149.95	148.41	150.41	148.32	1
	147.59	149.33	149.76	147.5	1
	147.75	148.4	148.65	147.23	1
	144.98	147.95	148.22	144.49	2 (Buy)
	149.61	149.63	150.11	148.58	1

4.2.3 Benchmarking the Result with GWO

This section demonstrates the comparison of GWO and MGWO with available results for stock data classification. Figure 4.1 demonstrates the convergence curve and classification accuracies alongwith comparison of classification with other algorithms i.e. PSO, GA, ACO, ES and PBIL. The classification rate for the stock data through MGWO is about 97% whereas, the classification rate through other algorithm is much lower. Moreover, the convergence graph shows that MGWO converge much faster than the compared meta-heuristic algorithms. The reason for such an improvement of convergence is: firstly, due to the strenghthen searching process by several random leaders in every iteration by MGWO whereas, GWO chooses the best alpha and beta, alpha follow. Secondly, due to the introducing of archive concept with the probabilistic model during the initialization phase that speeds up the convergence trends and enrich the quality of solution or accuracy. Thirdly, due to the re-generating random leaders in each iteration based on the statistical analysis performed on the collected fitness values in archive where, generation of wolf leaders will be highly randomized at the beginning of the hunt. This strategy can essentially improve the exploration power in the modified GWO from the early phase of iterations. Hence, the classification result through MGWO confirms that the selected stock data is suitable for applying neural network prediction algorithm to predict stock price.

Table 4.6 Experimental Result for the Stock Dataset

Algorithm	**Classification Rate**
GWO-MLP	95.3333%
PSO-MLP	80%
GA-MLP	67.6667%
ACO-MLP	66.6667%
ES-MLP	33.6667%
PBIL-MLP	33.3333%
MGWO-MLP	97%

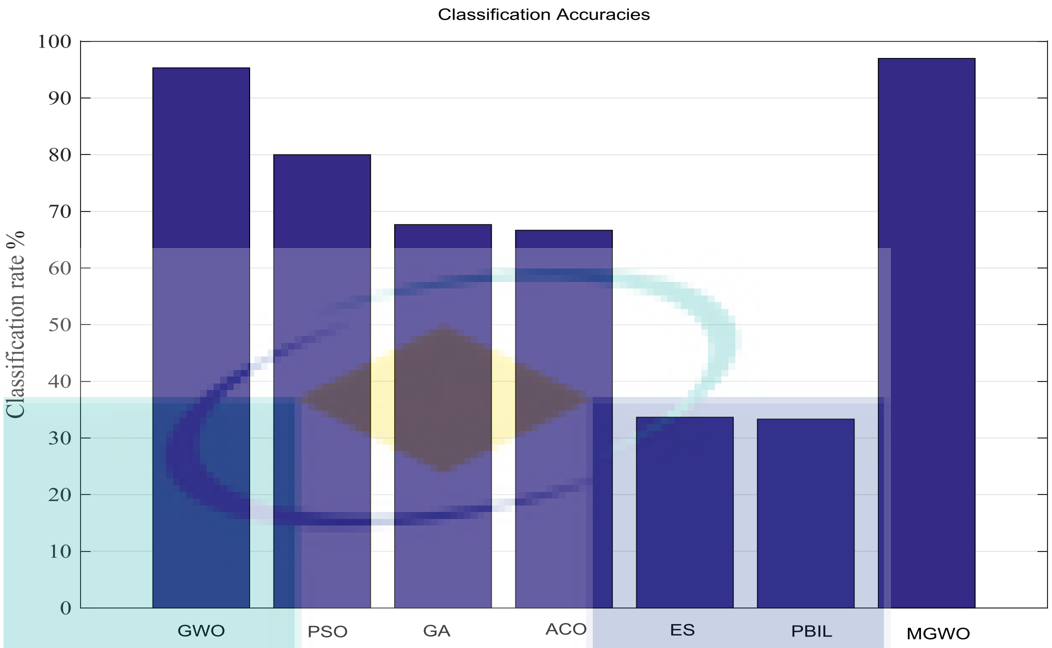


Figure 4.1(a) Classification Graph

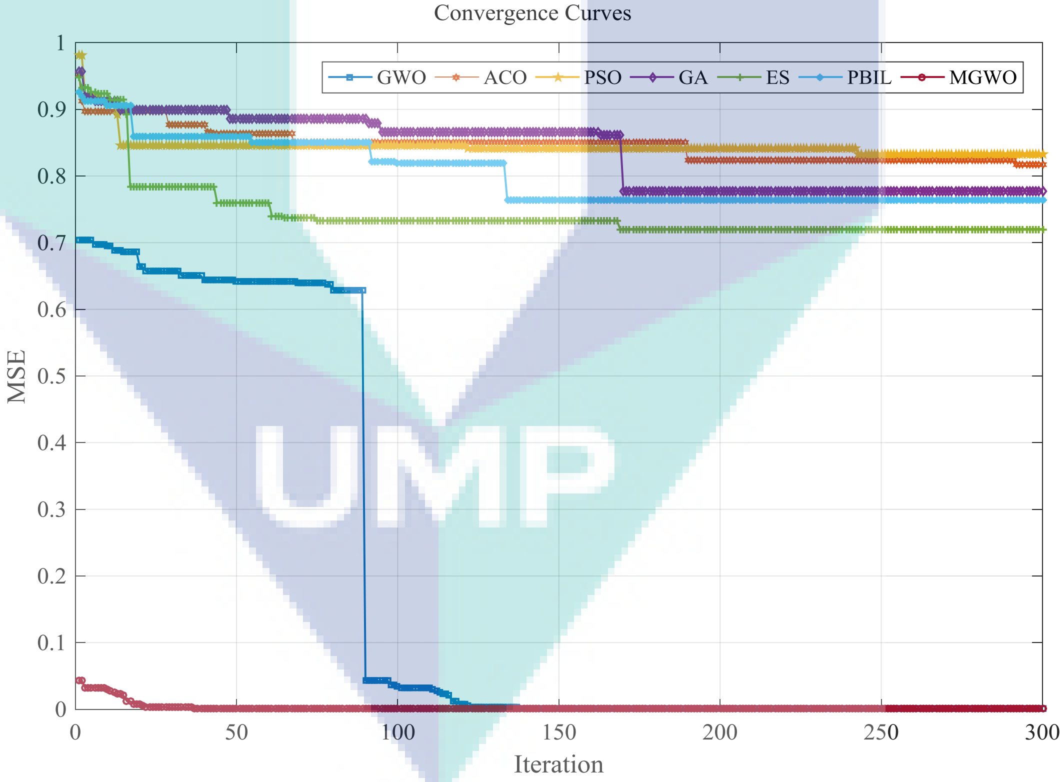


Figure 4.1(b) Classification Graph

4.2.4 Stock Prediction Results through MLP Neural Network

The six years of daily stock price for companies from NYSE, NASDAQ, Bursa Malaysia and DSE has been provided to predict the High Price through MLP neural network model which categorizes the data into Training, Validation and Test Set as indicated in Table 4.7. Where, Training Set consists 70% of total data, Validation Set contains 15% of data and Test Set holds 15% of data. The neural network model indicated in Figure 4.2 contains 10 Hidden Neurons and d is Number of Delays which is 2 in this architecture.

Table 4.7 Categorization of Dataset

Dataset Type	Amount
Training Set	70% of the target timesteps
Validation	15% of the target timesteps
Testing	15% of the target timesteps

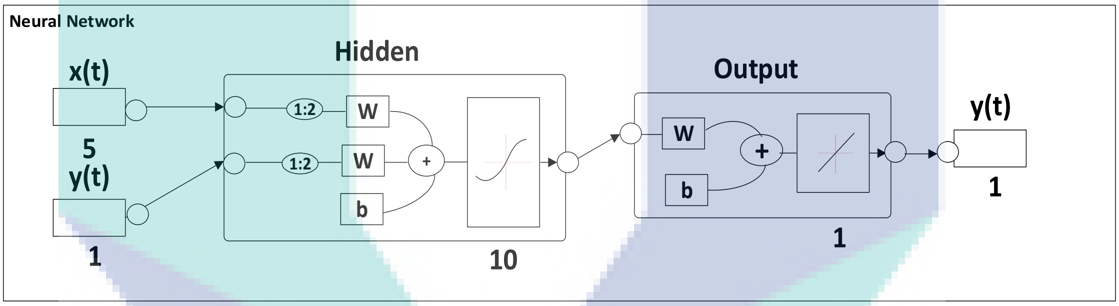


Figure 4.2 Architecture of the Neural Network Model

MGWO is applied to train the network to fit the inputs and targets. The training produces the model as indicated in Figure 4.3 which shows Training State (Plot train state), that Gradient= 0.39235, which is the calculation of weights used in network at epoch 28, which is one forward pass and one backward pass of all the training examples, Mu = 0.001 which is the control parameter for the algorithm used to train the neural network, at epoch 28. Validation checks= 6, at epoch 28.

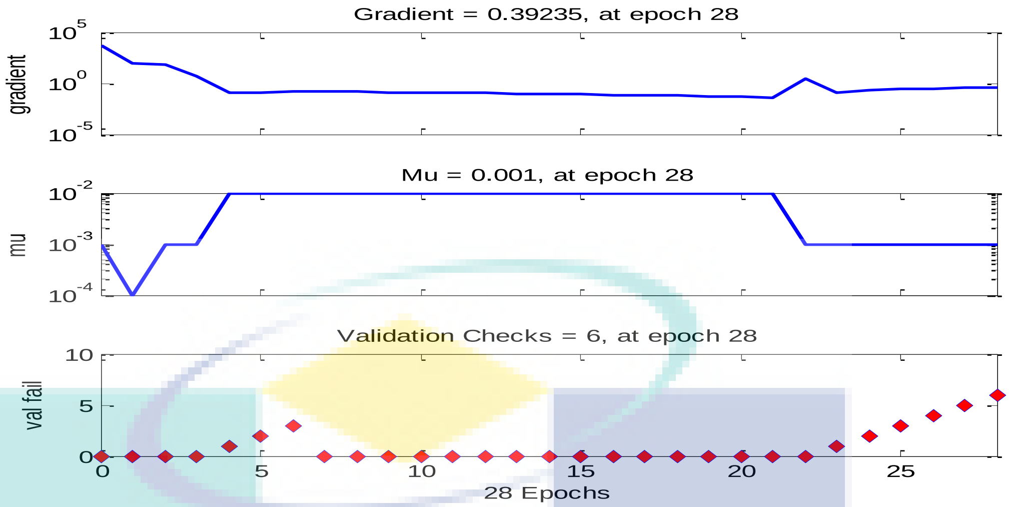


Figure 4.3 Training State, Gradient = 0.39235 at epoch 28. Mu = 0.001, at epoch 28. Validation Checks = 6, at epoch 28

The performance of the network is plotted in Figure 4.4. For different combinations of data and parameters, this performance curve varies. Training of the model stops when it reaches to mentioned number of epochs (shown in Figure 4.4) or alternatively, when Mean Squared Error (MSE) is almost never improving after certain epochs. The circle in the performance curve shows the best validation performance.

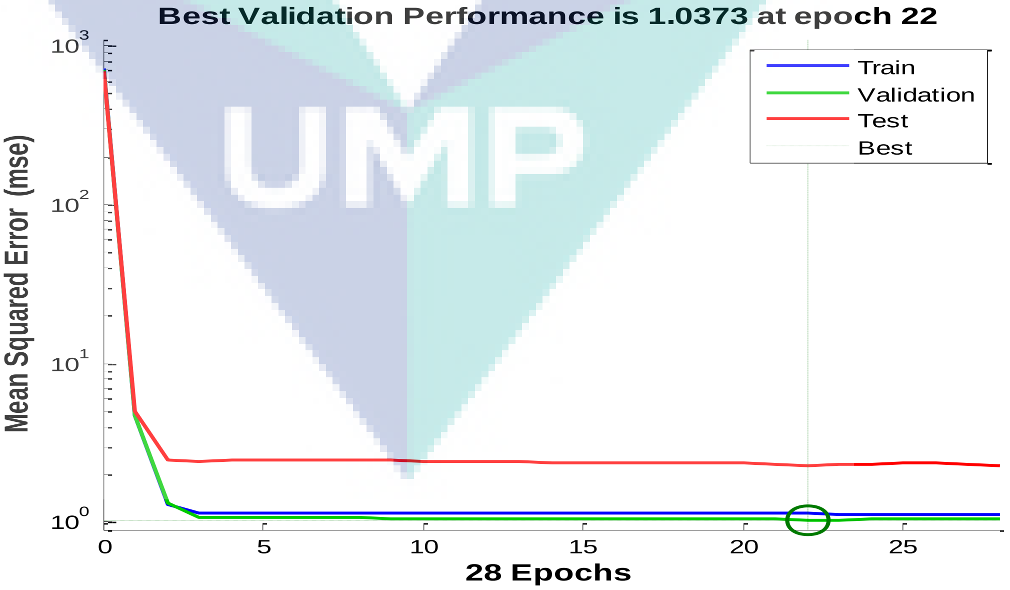


Figure 4.4 Validation Performance of the Network

Figure 4.5 represents the regression that facilitates defining nonlinear relationships in the experimental data and it can plot four regressions, demonstrating the network output with respect to actual data (target) for training, validation, test and all data sets. For a perfect fit, the data should fall along a 45 degree line, where the network outputs are equal to the targets (Mathworks, 2012a). Here, most data fall along 45 degree line and all the R values produced by each plot is more than 0.98 i.e. Regression by Training is 0.99273, by Validation 0.99356, by Test 0.98806 and by All 0.99194. Hence, it indicates that the fit by Regression is reasonably good for all data sets.

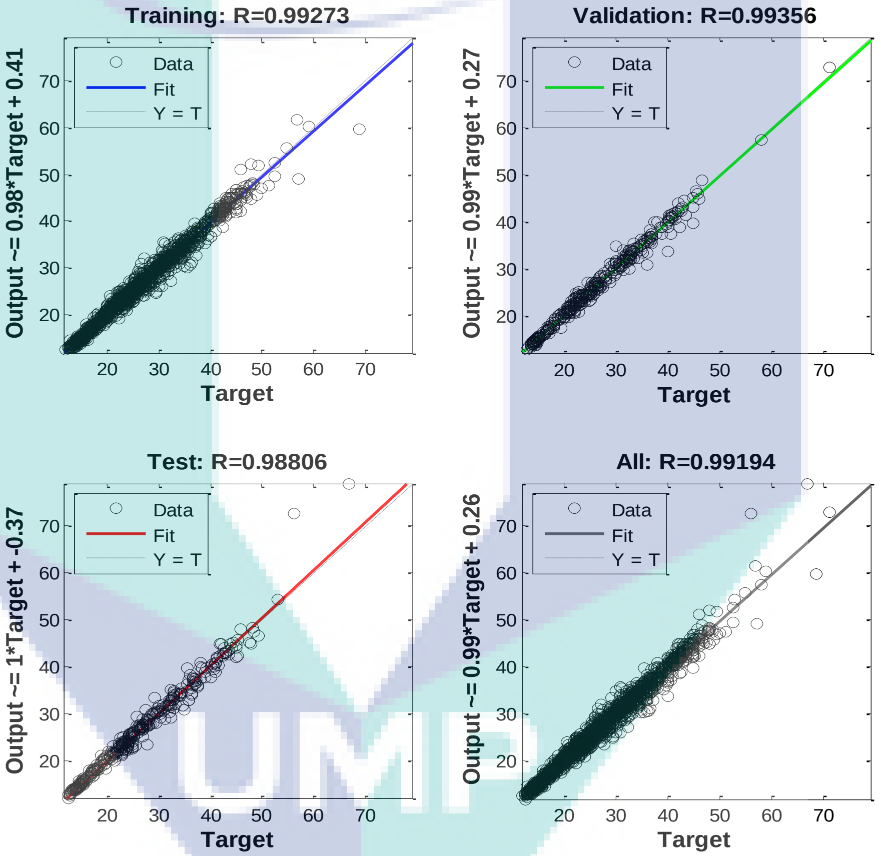


Figure 4.5 Neural Network Output with Respect to Target (Actual Data) through Regression

The time series response is plotted in Figure 4.6 which indicates that there is not much variation between training target, training outputs, validation targets, validation outputs, test targets, test outputs. We can observe that they tend to have similar patterns.

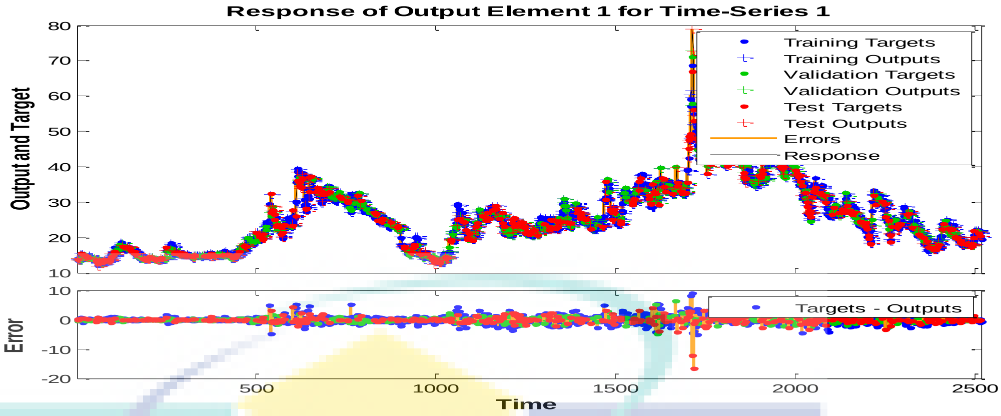


Figure 4.6 Response of Output Element for Time Series

The table 4.8 below shows Actual Price versus Predicted Price of Stock Prices through MLP neural network:

Table 4.8 Actual Price versus Predicted Price of INTECH

Company Name	**Actual**	**Predicted**	**Prediction Error (%)**
INTECH	15	13.725	8.5
INTECH	14	14.0309	0.2
INTECH	14	13.6375	2.6
INTECH	13.7	13.9958	2.2
INTECH	13.9	13.8036	0.7
INTECH	13.8	13.4366	2.6
INTECH	13.5	13.6769	1.3
INTECH	13.6	13.6421	0.3
INTECH	13.6	13.6421	0.3
INTECH	13.6	13.9009	2.2
INTECH	13.8	13.9647	1.2
INTECH	13.9	14.1863	2.1

The Time Lag of prediction is formally stated as: we find a function, $P: R^d \rightarrow R$ such as to obtain an estimate of $P(k)$ as indicated in Equation 3.23.

The final predicted closing price of ACI generated through MLP neural network is listed in the Table 4.9 which demonstrates Actual Price versus Predicted Price.

Table 4.9 Actual Price versus Predicted Price of ACI

Company Name	Date	Actual	Predicted	Prediction Error (%)
ACI	2015-10-01	583.30	583.74	0.1
ACI	2015-10-04	573.10	575.77	0.5
ACI	2015-10-05	577.40	575.91	0.3
ACI	2015-10-06	575.20	574.01	0.2
ACI	2015-10-07	570.80	571.10	0.1
ACI	2015-10-08	568.50	570.02	0.3
ACI	2015-10-11	560.40	562.39	0.4
ACI	2015-10-12	562.50	564.48	0.4
ACI	2015-10-13	556.70	559.11	0.4
ACI	2015-10-14	554.50	556.42	0.3
ACI	2015-10-15	553.70	557.81	0.7

Figure 4.7 demonstrates the price of ACI for 6 years (A day in October). The Table 4.10 illustrates Actual Price versus Predicted Price of stock price through MLP neural network for different stock market for a day, where the stock value for NYSE and NASDAQ in USD, stock value for Bursa Malaysia in Ringgit Malaysia (RM) and stock value for DSE in Bangladesh Taka (BDT).

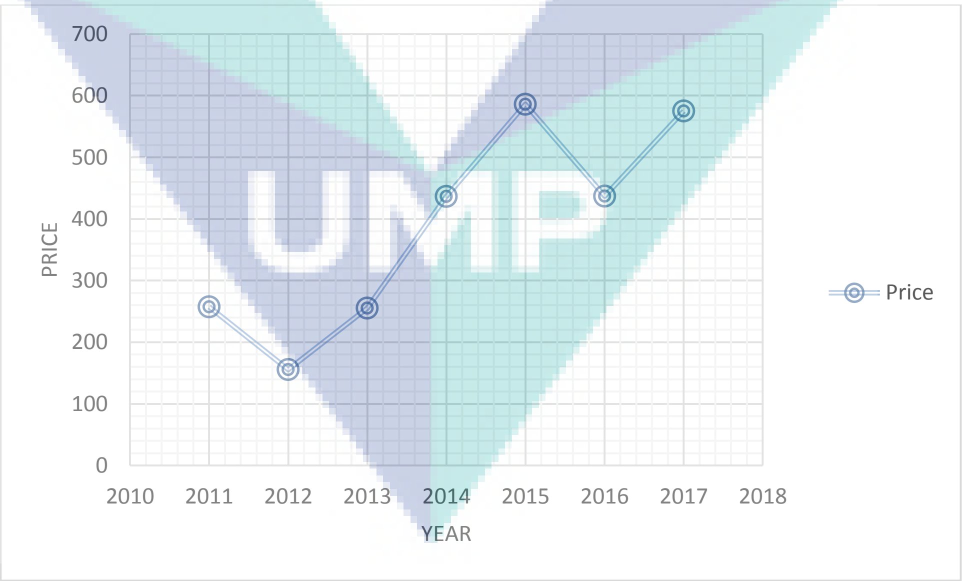


Figure 4.7 Share Price for ACI for 6 years (A day in October)

Table 4.10 Actual Price versus Predicted Price for Various Stock Markets Worldwide

Company Name	Stock Market	Date	Actual	Predicted	Prediction Error (%)
Microsoft	NASDAQ,USA	2016-12-01	60.15	59.71	0.72
IBM	NYSE,USA	2016-12-01	162.2	160.966	0.76
Digi	Bursa Malaysia	2016-12-01	4.96	4.986	0.536
ACI	DSE, Bangladesh	2016-12-01	414	415.37	0.33

4.2.5 Effect of Various Factors on Stock Price

The effect of stock price on various factors like Gold Price, Dollar Price, Bank Interest Rate, Foreign Direct Investment (FDI) and Inflation has been measured and it has produced the output indicated in Table 4.11. The P-value 0.5 or higher specifies that the effect is stronger, otherwise the effect is not very strong. Here, the P-value produced through the experiment indicates that there are some effect of those factors in fluctuating the stock price. However, the effect is not so strong and hence the factors may not heavily effect in changing (increasing or decreasing) the stock price.

Table 4.11 Effect of Various Factors on Stock Price

Regression Statistics	
Multiple R	0.0702883
R Square	0.0049405
Adjusted R Square	-0.0022389
Standard Error	0.5830351
Observations	699

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	5	1.16960749	0.23392	0.68815	0.63254
Residual	693	235.571451	0.33993		
Total	698	236.741059			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	1.480864	0.03804066	38.9284	1E-176	1.40618	1.55555	1.40618	1.55555
Gold Price	-0.0388271	0.06284375	-0.6178	0.53689	-0.1622	0.08456	-0.1622	0.08456
Dollar Price	0.0210943	0.03016187	0.69937	0.48456	-0.0381	0.08031	-0.0381	0.08031
Bank Interest Rate	-0.1545621	0.33800539	-0.4573	0.64762	-0.8182	0.50908	-0.8182	0.50908
FDI	-0.5084582	0.34658954	-1.467	0.14282	-1.1889	0.17203	-1.1889	0.17203
Inflation	0.0380297	0.10948312	0.34736	0.72843	-0.1769	0.25299	-0.1769	0.25299

4.2.6 Performance Measurement of Prediction

The performance measurement of the neural network model using Mean Absolute Percentage Error (MAPE), Mean Absolute Deviation (MAD) and Root Mean Squared Error (RMSE) is calculated. Suppose ($a_1,a_2,a_3,\ldots\ldots,a_n$) are actual values and ($p_1,p_2,p_3,\ldots\ldots,p_n$) are the predicted values then the MAPE, MAD and RMSE can be calculated using the Equation indicated in 3.20, 3.21 and 3.22.

The evaluation of the neural network performance through the above equation is shown in Table 4.12.

Table 4.12 Evaluation of Prediction for INTECH

Input Parameters (Previous 10 Average Prices)	**Neural Network Architecture**	**Forecasting Performance**	**INTECH**
P_{t-1}	MLP	MAPE	2.0
$P_{t-2},\ldots, P_{t-9}$	MLP	MAD	0.3
P_{t-10}	MLP	RMSE	0.4

In this work, INTECH Company's real price data is anticipated which was high level in Dhaka Stock Exchange. The stock data for INTECH is used for prediction through the neural network. After the network is created, the evaluation demonstrates a positive performance improvement, which is very encouraging for this research work and it will guide the investor towards investment in a particular security.

Table 4.13 demonstrates the performance evaluation for ACI that indicates positive performance improvement through the created network, which is encouraging for this research work as well for guiding the investor for investment into a particular stock.

Table 4.13 Performance Evaluation of Ensemble Model for ACI

Input Parameters (Previous 10 Closing Prices)	**Neural Network Architecture**	**Forecasting Performance**	**ACI**
P_{t-1}	MLP	MAPE	0.28
$P_{t-2},\ldots, P_{t-9}$	MLP	MAD	1.18
P_{t-10}	MLP	RMSE	1.75

The performance of the predicted price is evaluated through the Equation 3.22 for various instruments of different stock markets produced the Root Mean Square Error (RMSE) value as indicated in Table 4.14 and the Prediction error related to the prediction is indicated in Figure 4.8. The RMSE value close to 0 indicates no error and prediction is completely acceptable. The Prediction error remains lower and reasonable through the proposed model. However, the prediction error for NASDAQ is little higher than other stock market which is 0.7624 due to the effect of factors. Hence, it can be concluded that the proposed ensemble model can be applied for stock prediction that can also reduce the error.

Table 4.14 RMSE Value for Instrument of Various Stock Markets

Company Name	Stock Market	RMSE
Microsoft	NASDAQ,USA	0.7624
IBM	NYSE,USA	0.0199
Digi	Bursa Malaysia	0.0645
ACI	DSE, Bangladesh	0.2249

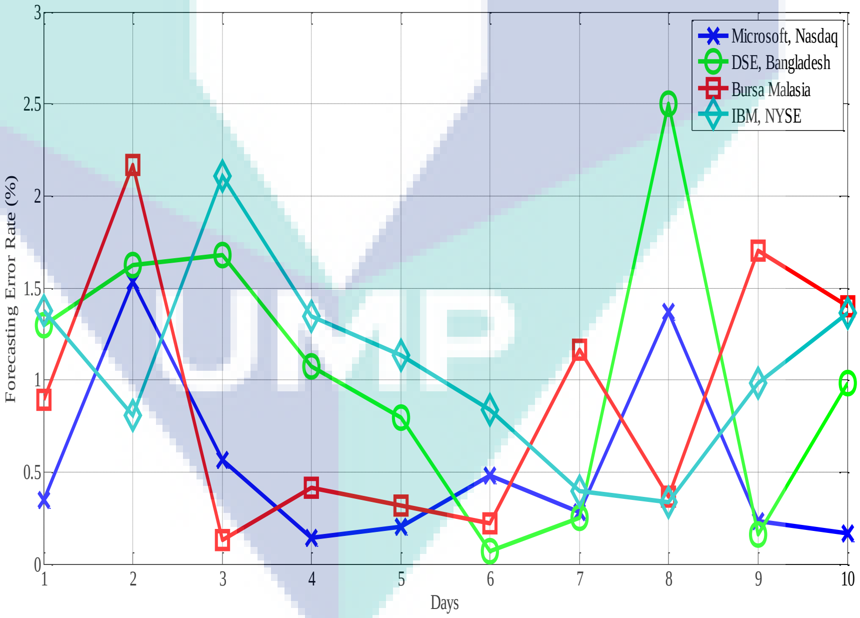


Figure 4.8 Comparison of Prediction Error for Various Stock Markets

The performance of a neural network can be also evaluated through generalization capability (Kaastra & Boyd, 1996). The proposed prediction model can perform well for various stock markets and hence it has good generalization capacity.

4.3 Comparison of Proposed Model with Existing Works

The existing models tried to predict from the markets perspective by applying models such as technical analysis, fundamental analysis and linear regression; none of these however has proved to be consistent in making correct prediction with less error (Adebiyi et al., 2012 and Lawrence, 1997). These methods are based on base level standard. Technical analysis phase is very subjective and contradict the efficient market hypothesis (Lawrence, 1997). It is difficult to time the market through fundamental analysis and optimizing the approach is time consuming and hard to implement (Adebiyi et al., 2012 and Lawrence, 1997). In contrast, finding the global network minimum is not guaranteed through Back Propagation. Although error is not minimized through this, there is possibility of weight modification to meet local minimum in the error landscape, but the network may not be optimized (Lawrence, 1997 and Rojas, 1996). Convergence through Back Propagation is very slow and not guaranteed. Moreover, learning requires input scaling and normalization (Budhani et al., 2012). Neural network approach is having a drawback of only "learn" through past patterns which requires skilled tuning of the parameters. The basis of the stock price movements is very difficult to capture. Over fitting is a serious problem (Haykin, 1994) that occurs when the network has too many free parameters. These parameters allow the network to fit well with the training data but typically lead to poor generalization. There are two main reasons for this, first is due to having too many nodes to the networks and the second is due to the network being trained more. If the input data has high dimensions then NN is restrained in learning the patterns (Rojas, 1996). Multi-layer perceptron has problems of getting stuck in a local minimum and it is very slow in learning (Mirjalili et al., 2014a).

In the proposed model, an ensemble model consisting of neural network and MGWO has been implemented. Here, k-means clustering is used to categorize the organization, classification to determine the suitability of data for prediction, feature selection, and learning. Then, MLP neural network algorithm is applied to predict the stock price. For stock prediction, neural network has been taken as preferred network instead hybrid model because it takes less time and performs well (Jia et al., 2013 and Fasanghari et al., 2015). After processing the result the error is calculated in percentage and then evaluated the performance. Fewer errors have been found in the case of the ensemble model that consists of neural network and MGWO when compared to other meta-heuristic models. Finally, the performance analysis phase demonstrates the accuracy of the model of the stock market.

According to the analysis, it has been realized that, more accurate matches amongst the data has been experienced through this research. Figure 4.9 shows the performance of the prediction through the proposed model, where the value of Actual, Predicted and Prediction Error have compared. It indicates that the prediction error is slightly high for first observation. However, the prediction errors are trivial for remaining observations and most of the Actual and Predicted values are much closer. A predictive model is just a guess of what will happen in future. Hence, experimental findings through a model will be acceptable if the error is less.

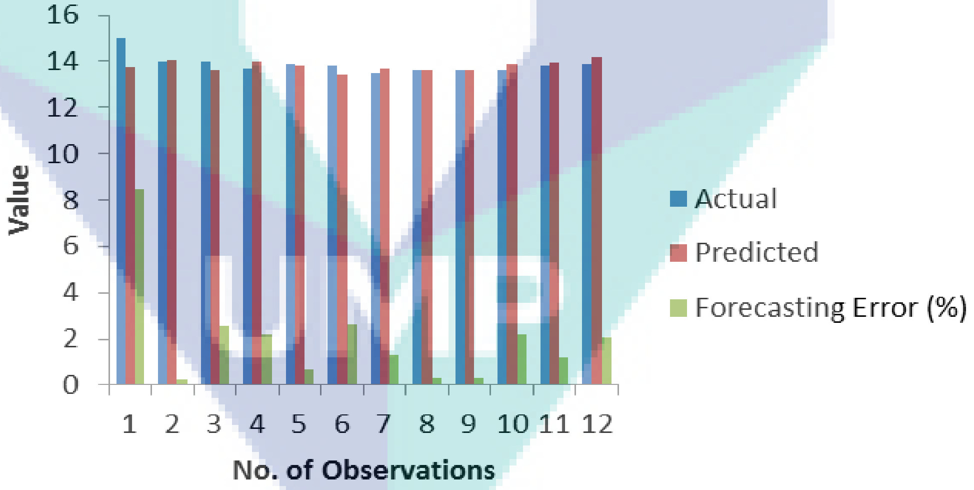


Figure 4.9 Prediction Performance

The Mean Squared Error (MSE) value is also quite reasonable in proposed work i.e. 0.4. Provided that the lower MSE value is always better. MSE value 0 indicates that there is no error and the prediction is utterly acceptable (Yetis et al., 2014).

MGWO has produced good classification rate 97 which is better than other algorithms. Figure 4.10 demonstrates below the performance comparison between Proposed and Existing work in terms of prediction price. It indicates that the prediction is closer to actual price in proposed model applying MLP compared with existing models applying GWO, GA, ACO, PSO, PBIL and ES (Navale et al., 2016; Hafezi et al., 2015; Billah et al., 2015; Khan et al., 2011). So it can be concluded that, the ensemble of neural network and MGWO can reduce the errors and the prediction can be more accurate as well. Table 4.15 presents the comparison between existing and proposed research findings.

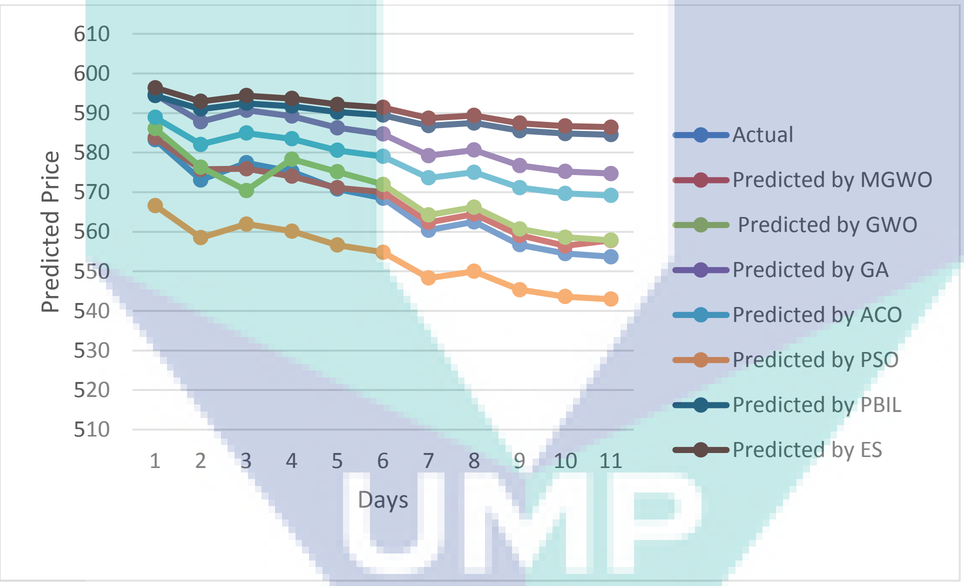


Figure 4.10 Comparison of Prediction Performance

Table 4.15 Comparison between Existing and Current Research Findings

Research Model	Year of Research	Existing Research Finding	Current Research Finding
General Stock Prediction Model	Xiong et al. (2014)	Statistical model i.e. multi-output support vector regression is used for stock index prediction (Xiong et al., 2014) which can deal with linear data but stock data is non-linear.	Ensemble model can deal well with non-linear data.
Stock Prediction using Data Mining	Patel et al. (2015)	Data pre-processing and use of discrete data is emphasized for the improvement of prediction accuracy (Patel et al., 2015) where the algorithm gained about 50% accuracy whereas some algorithms remained silent for few situations i.e. when to sell the stock.	Ensemble model consisting of Neural Network and MGWO is applied for stock prediction which can make about 95% accurate prediction.
Stock Prediction using Neural Network	Hafezi et al. (2015)	Bat Neural Network Multi Agent System (BNNMAS) is used to predict the stock price for DAX and gained significant result with good accuracy for long term period. However, the Mean Absolute Percentage Error (MAPE) remained 2.84 for such model (Hafezi et al., 2015).	Ensemble model can reduce the errors, avoids the problem of over-fitting or under-fitting and MAPE remained 2.0 or lower.
Stock Prediction using Classification	Ballings et al. (2015)	Some studies have compared their result with buy-and-hold strategy and found the combination of technical analysis and classification produced more profits. Eventually, their proposed model also reduced the risk. However, they have suggested using of better classification algorithm such as SVM, and k-NN. to achieve more generalization (Ballings et al., 2015).	Ensemble model is a combination of neural network and MGWO which gains more accuracy in predicting and error is reduced as well. The model is applied to Asian market and compared with prominent market

Table 4.15 Continued

Research Model	**Year of Research**	**Existing Research Finding**	**Current Research Finding**
Stock Prediction using Ensemble Algorithms	Navale et al. (2016)	Some studies have availed about 77% accuracy in prediction applying combinatorial algorithms which is higher than the single use of ANN or DT. Researchers recommended to perform further research by applying other models such as SVM, ensemble of artificial intelligence and other similar algorithm to reveal the weaknesses of other researches to reveal the uncertainty of stock market (Navale et al., 2016).	The proposed Ensemble model gained more accuracy which is about 95% through the combination of neural network and MGWO that provides more accuracy with reduced error rate
Stock Prediction covering various factors	Bonde et al. (2012), Negnevitsky (2005) and Hafezi et al. (2015)	Few studies have investigated the effect of some factors and indicated that various internal and external factors play important role in moving the stock price that is why stock prediction is so complicated.	Current study combines the factors with stock data and investigates the effect of the factors on stock data to provide better guideline to investors.
Generalization of stock prediction for various markets	Lertyingyod et al. (2016) and Hafezi et al. (2015)	Most of the researches are concentrated on a single stock market. Lertyingyod et al. (2016) investigated Thailand Stock Exchange and Hafezi et al. (2015) concentrated on DAX.	The proposed research provides more generalization by predicting for various stock markets i.e. promising and emerging stock markets and making a comparison.

4.4 Statistical Analysis for the Experimental Findings

To validate the findings through the experimentation, this section demonstrates the statistical analysis and comparisons for the obtained outcomes. The statistical difference between the existing and proposed approach is determined through two tests specifically, Friedman test and Wilcoxon signed-rank test. Precisely, Friedman test is implemented to distinguish contrasts between the findings through all approaches and Wilcoxon signed-rank test is applied based on the result of Friedman test for the analysis of each approach's significance.

Friedman test is conducted through two hyphotheses namely, null hypothesis (H_0) and alternative hypothesis (H_1) as per the Equation indicated in 4.1, 4.2 and 4.3.

$$H_0: \mu_1 = \mu_2 = \mu_3 .... = \mu_n \quad 4.1$$

$$H_1: \mu_1 \neq \mu_2 \neq \mu_3 .... \neq \mu_n \quad 4.2$$

$$\square^2 = \frac{12c}{n(n+1)} \sum_{i=1}^{n} \left(R_i - \frac{(n+1)}{2} \right)^2 \quad 4.3$$

Where, n is the number of approaches, μ_i is the median of results for ith approach, c is the number of comparisons, and R_i are the ranks approaches' results. In Equation 4.3, if the Friedman test statistic $\square^2$ > critical value, then H_0 will be rejected. Here, the critical value is calculated based on a probability threshold called Alpha (α) and Degree of freedom (df).

Friedman test has been conducted for the stock prediction samples in Table 4.9 for all the approaches namely, MGWO-based-ensemble, GWO-based-model, GA-based-model, ACO-based-model, PSO-based-model, PBIL-based-model and ES-based-model. The test produces the results presented in Table 4.16.

Table 4.16 Friedman Test

Test Statistics		Decision
Number of Samples	11	Reject H_0 and there are differences between findings of each approaches
Degree of freedom (df)	7	
Critical Value	14.06	
Chi-Square($\square^2$)	269.0000048	
Asymp. Sig.	0.000	

Table 4.16 illustrates the statistical analysis through Friedman test for stock prediction data. The analysis specifies that there are significant differences between exiting approaches and MGWO. For the experiment, null hypothesis (*H_0*) is rejected. Moreover, MGWO performs significantly better than GWO, GA, ACO, PSO, ES and PBIL.

Wilcoxon Signed Rank Test analyses the significance of each approach. The test is conducted through two hyphotheses namely, null hypothesis (*H_0*) and alternative hypothesis (*H_1*) as per Equation indicated in 4.4 and 4.5.

$$H_0: \mu 1 - \mu 2 = 0 \tag{4.4}$$

$$H_1: \mu 1 - \mu 2 \neq 0 \tag{4.5}$$

where μ1, μ2 are the median for proposed approach and median for other approach respectively. H_0 in Equation 4.4 implies that there is no significant difference between the two approaches', while H_1 in Equation 4.5 specifies that there is a difference between the two approaches'. Here, the decision is made based on α or significance level.

Table 4.17 illustrates the statistical analysis through Wilcoxon signed-rank test for stock prediction data. For the experiment, null hypothesis (*H_0*) is rejected. The analysis specifies that there are significant differences between MGWO and existing approaches.

Table 4.17 Wilcoxon Signed Rank Test

Pairs	**Ranks**				**Asymp. Sig. (2-tailed)**	**Conclusion**
	Negative	**Positive**	**Ties**	**Total**		
GWO-MGWO	66	0	0	66	0.9987	Reject H_0
PSO-MGWO	66	0	0	66	0.9987	Reject H_0
GA-MGWO	66	0	0	66	0.9987	Reject H_0
ACO-MGWO	66	0	0	66	0.9987	Reject H_0
ES-MGWO	66	0	0	66	0.9987	Reject H_0
PBIL-MGWO	66	0	0	66	0.9987	Reject H_0

4.5 Discussion

Developing a general model for stock prediction is very complex because of non-linear nature of stock data. There is no single predominant approach to perform stock prediction. Parameter tuning and feature selection can play vital role to achieve fair outcome. Application of meta-heuristic can be an effective approach in this field, however meta-heuristic algorithm needs to exploit the search operator to enhance the performance.

The choice of Grey Wolf Optimizer (GWO) as a basis of classification is worth mentioning here. The encouraging phase of GWO and MGWO are that the algorithms are formed based on simple concept consists of hunting preys by grey wolves in nature. Due to this, the application of the algorithm is straightforward. On the other hand, all meta-heuristic algorithms are not so easy to implement as an algorithm suits well with one may not perform well for other problems.

Moreover, choice of appropriate value for parameter is very challenging with meta-heuristic approach because the performance of the algorithm vigorously depend on parameter modification. Inappropriate parameter value may result expensive computational efforts with poor outcome. In addition, difficult approaches with numerous parameters require noteworthy endeavors for adjustment. For instance, GA needs adjustment of mutation rate, crossover rate, population size, but GWO and MGWO needs to adjust two parameters $\vec{A}$ and $\vec{C}.$ Feature selection is another issue that needs attention as additional features may deteriorate the outcome. So, reduced and appropriate feature produce better prediction which is facilitated through MGWO.

Regarding the overall performance of MGWO and complete ensemble model for stock prediction, ensemble model with MGWO produces better result in comparison with other models. Additionally. MGWO can produce better result in comparison with GWO due to its exploration and exploitation capacity with archive, strengthen searching process by several random leaders and re-generating random leaders in each iteration. Indeed, MGWO can convergence well in compared to GWO. However, computationally MGWO may take longer time than GWO because of implementing archive.

Finally, the statistical test confirms that there are significant differences between MGWO and existing approaches. Besides, MGWO performs significantly better than GWO, GA, ACO, PSO, ES and PBIL.

4.6 Validity Threats

The research may accompany several validity threats with the experimental studies. Few threats have been detected in this study that may effect the results obtained through current research.

Firstly, the choice of benchmark is a crucial threat. The study implements the experimental bechmarks for various renowned researches undertaken earlier in literature. Although, the employed benchmarks are chosen from Bangladesh, Malaysia, NYSE and NASDAQ stock markets, the same technique proposed in this research can be applicable to other stock markets also.

Secondly, all the approaches GA, ACO, PSO, ES, PBIL, GWO, MGWO select 300 maximum number of generations and 1000 population size which can be a major threat to the experimentations due to unfair number of comparisons. Because, some of the approaches may complete the search earlier. We can elliminate this threat by not limiting the generations and population size, instead we can set the same maximum number of fitness function evaluation as a stopping criteria.

Thirdly, the meta-heuristic approaches implement random search operator which can be a threat as well. The optimum result can be determined just for once by chance. Hence, the comparison of the optimum result may not indicate the actual performance of an approach. The solution to this threat may be avoided by selecting the mean result instead of optimum result.

Another threat is the comparison with other approaches. Many approaches are adopted meta-heuristic algorithm for stock prediction. However, accommodating all the approaches is beyong the implementation of current study as MGWO is not benchmarked with all available approaches in literature. To overcome this threat, the current research selects the recently published journal to choose the approaches related to GWO and stock prediction.

Lastly, performance evaluation strategies can also indicate another threat. The internal structure of performance measurement algorithm differs for different approaches. However, the current study selects the recognized performance measurement in literature (i.e., for stock prediction).

4.7 Chapter Summary

This chapter demonstrates the experimental setup and evaluation of MGWO. Initially, MGWO is compared against meta-heuristic models such as PSO, GA, ACO, ES, PBIL and GWO. Then, the performance of ensemble model is demonstrated through stock dataset. The stock prediction result through application of k-means clustering data mining algorithm, classification using meta-heuristic algorithm are also displayed. The result through the ensemble of MLP neural network and MGWO is also demonstrated. The comparison of proposed research work with existing models and performance evaluation of the prediction are also illustrated in this chapter. Finally, the results from experiments are evaluated statistically applying statistical test.

CHAPTER 5

CONCLUSION

5.1 Introduction

The previous chapter presented the experiments related to application of ensemble model consisting neural network and Modified Grey Wolf Optimizer (MGWO) for stock prediction. The conclusion of the thesis is included in this chapter that comprises some of the key findings of this research work in light of the research objectives. The contribution of this research towards knowledge is also restated in this chapter, specifically in the area of neural network and MGWO. Limitations of the research particularly the strict designing and incapability of the research are discussed as well in this chapter. The brief answers to research questions listed in chapter 1 are also attempted in this chapter. Finally, the obstacles faced in performing the research and recommendations of future directions of this research are outlined in this chapter.

5.2 Objectives Revisited

The research was aimed at enhancing the GWO algorithm and address its limitation as far as exploration and exploitation capabilities.
The objectives of the research were as below:

- To develop a modified GWO algorithm with random selection of leaders
- To adopt the modified GWO algorithm for training of neural network as ensemble model with stock market prediction analysis as case study
- To evaluate the performance of ensemble model against existing strategies in terms of the other developed optimization model in literature

The first objective has been addressed in Chapter 3 that proposes the improvement of GWO. The GWO has been modified by strengthen the searching process via several random leaders in each iteration, re-generating the random leaders in each iteration and introducing archive to verify the solution with better probability to proceed further for training and re-generation. The improvement of exploration is achieved through the proposed approach.

The second objective has been addressed in Chapter 3 that demonstrates the design and implementation of the proposed ensemble of neural network and MGWO approach. The basis of the research is GWO algorithm that has been modified to design MGWO. The MGWO has been implemented with the wrapper's approach to select the best set of features for stock prediction. The objective has been fulfilled through the successful implementation of MGWO and neural network to form ensemble approach for stock prediction.

The third objective of the study has been achieved through the evaluation of the proposed approach, presented in Chapter 4, demonstrating the comparison of MGWO against GWO and other existing meta-heuristic approaches. The performance of ensemble approach is better than the existing strategies. For stock prediction, the performance of MGWO and ensemble approach outperforms in comparison with other existing approach as established by the statistical analysis.

Regarding the performance of MGWO and GWO, the experimental results demonstrated that MGWO outperforms GWO. Moreover, the convergence rate of MGWO is better than GWO.

Placing everything altogether, the objectives of the research has been achieved through the design, implementation and evaluation of MGWO. The research has been made significant contributions to fulfill the objectives.

5.3 Contributions of the Research

To add up the earlier discussion, the research contributes an ensemble approach in relation to stock prediction. The contribution of the research toward knowledge can be listed as:

- Modified Grey Wolf Optimizer (MGWO) modified GWO for improving the exploration of GWO via several random leaders in each iteration, re-generating the random leaders in each iteration and introducing archive to verify the solution to check the solution that provides better probability to proceed further for training and re-generation

- Ensemble of neural network and MGWO improves stock prediction with good accuracy and reduced error rate

- Ensemble approach contributes to apply it as an alternative approach for building better prediction model compared to existing approaches

5.4 Future Directions of the Research

The research has demonstrated the modification of GWO and training neural network to form the ensemble model for stock market prediction. It has also shown the designing of algorithm applying numerous pre-defined steps that can be used for exploring stock data. The research has also illustrated the ways to reveal the useful pattern incorporated with stock data to facilitate the investor to gain through stock investment. The future research directions can be indicated as:

- Currently, the research is an attempt to modify GWO and training neural network to form ensemble model for stock market prediction. To improve and strengthen the result of the research, it is possible to carry out better ensemble or hybrid approach.

- The study investigated to implement ensemble approach applying MLP neural network and MGWO. Further investigations can be made to combine other methods and various parameters can be fine-tuned for better improvement.

- The improvement of computational cost for MGWO algorithm may be investigated by implementing the algorithm for parallel approach. The complexity of the MGWO algorithm may be addressed by fine tuning the algorithm.

- This study uses daily stock data for six years of stock data. However, attempts can be made to consider the stock data with different time period. Additionally, different ratio of data may be considered for training, validation and testing.

- Positive and negative news can be incorporated with stock data to measure their effect on stock movement.

- Moreover, the research attempted to measure the effect of five different factors on stock price of DSE, which may be extended for more factors. In addition, attempt can be made further to collect the factors data for other stock markets.

- Finally, the performance of MGWO algorithm may be fine-tuned through hybrid it with other meta-heuristic algorithm so that the algorithm may gain better search ability.

REFERENCES

Adebiyi, A., Ayo, C., Adebiyi, M. O., & Otokiti, S. (2012). Stock Price Prediction Using Neural Network with Hybridized Market Indicators. *Journal of Emerging Trends in Computing and Information Sciences*, *3*(1), 1-9.

Alba, E., Talbi, E., Luque, G., & Melab, N. (2005). *Parallel Metaheuristics: A New Class of Algorithms*. Wiley, 79-104.

Ali, B. A. S., Ahmed, A. B. E. D., & Muhammad, A. E. D. (2014) . Incremental Learning Approach for Enhancing the Performance of Multi-Layer Perceptron for Determining the Stock Trend. International Journal of Sciences: Basic and Applied Research (IJSBAR), Volume 16, No 1, 15-23.

Ammu, P. K., Sivakumar, K. C., & Rejimoan, R. (2013). Biogeography-Based Optimization-A Survey. International Journal of Electronics and Computer Science Engineering, 2(1), 154-160.

Anstreicher, K. M. (2012). On Convex Relaxations for Quadratically Constrained Quadratic Programming. Mathematical Programming, 136(2), 233-251.

Atsalakis, G. S., Dimitrakakis, E. M., & Zopounidis, C. D. (2011). Elliott Wave Theory and Neuro-Fuzzy Systems, in Stock Market Prediction: The WASP System. Expert Systems with Applications, 38(8), 9196-9206.

Ballings, M., Van den Poel, D., Hespeels, N., & Gryp, R. (2015). Evaluating Multiple Classifiers for Stock Price Direction Prediction. Expert Systems with Applications, 42(20), 7046-7056.

Baluja, S. (1994). Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning. Population Based Incremental Learning, CMU-CS-94-163, 1-41.

Bao, Y., Hu, Z., & Xiong, T. (2013). A PSO and Pattern Search Based Memetic Algorithm for SVMs Parameters Optimization. Neurocomputing, 117, 98-106.

Bazaara, M. S., Jarvis, J. J., & Sherali, H.D. (2011). Linear Programming and Network Flows: John Wiley & Sons.

Behesti, Z., & Shamsuddin, S. M. H. (2013). A Review of Population-Based Meta-Heuristic Algorithms. International Journal of Advance Soft Computing Applications, 5(1), 1-35.

Berry, M. J., & Linoff, G. (1997). Data Mining Techniques: for Marketing, Sales, and Customer Support: John Wiley & Sons, Inc.

Bertekas, D.P. (2014). Constrained Optimization and Lagrange Multiplier Methods: Academic Press.

Bilbao, M., & Alba, E. (2009). Simulated Annealing for Optimization of Wind Farm Annual Profit. Paper Presented at the Logistics and Industrial Informatics.

Billah, M., Waheed, S., & Hanifa, A. (2015). Predicting Closing Stock Price using Artificial Neural Network and Adaptive Neuro Fuzzy Inference System ANFIS: The Case of the Dhaka Stock Exchange. International Journal of Computer Applications, 129(11), 1-5.

Binitha, S., & Sathya, S. S. (2012). A Survey of Bio Inspired Optimization Algorithms. International Journal of Soft Computing and Engineering, 2(2), 137-151.

Birge, J. R., & Louveaux, F. (2011). Introduction to Stochastic Programming: Springer Science & Business Media.

Bisoi, R., & Dash, P. K. (2014). A Hybrid Evolutionary Dynamic Neural Network for Stock Market Trend Analysis and Prediction Using Unscented Kalman Filter. Applied Soft Computing, 19, 41-56.

Bliss, C. A., Frank, M. R., Danforth, C. M., & Dodds, P. S. (2014). An Evolutionary Algorithm Approach to Link Prediction in Dynamic Social Networks. Journal of Computational Science, 5(5), 750-764.

Blondin, J., & Saad, A. (2010). Metaheuristic Techniques For Support Vector Machine Model Selection. Paper Presented at the 10th International Conference on Hybrid Intelligent Systems (HIS).

Blum, C., & Roli, A. (2003). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. ACM Computing Surveys (CSUR), 35(3), 268-308.

Bonde, G., & Khaled, R. (2012). Stock Price Prediction Using Genetic Algorithms and Evolution Strategies. Paper Presented at the Proceedings of the International Conference on Genetic and Evolutionary Methods (GEM), 2012.

Budhani, N., Jha, C., & Budhani, S. K. (2012). Application of Neural Network in Analysis of Stock Market Prediction. IJCSET, 3, 61-68.

Burke, E., Bykov, Y., & Hirst, J. (2007). Great Deluge Algorithm for Protein Structure Prediction.

Burke, E. K., & Bykov, Y. (2017). The Late Acceptance Hill-Climbing Heuristic. European Journal of Operations Research, 258(1), 70-78.

Burke, E., Bykov, Y., Newall, J., & Petrovic, S. (2003). A Time-Predefined Approach to Course Timetabling. Yugoslav Journal of Operations Research, 13(2), 19-51.

Cai, Q., Zhang, D., Zheng, W., & Leung, S. C. (2015). A New Fuzzy Time Series Forecasting Model Combined with Ant Colony Optimization and Auto-Regression. Knowledge-Based Systems, 74, 61-68.

Chan, R. W., Yuen, J. K., Lee, E. W., & Arashpour, M. (2015). Application of Nonlinear-Autoregressive-Exogenous Model to Predict the Hysteretic Behaviour of Passive Control Systems. Engineering Structures, 85, 1-10.

Chandrashekar, G., & Sahin, F. (2014). A Survey on Feature Selection Methods. Computers & Electrical Engineering, 40(1), 16-28.

Codd, E. F. (2014). Cellular Automata: Academic Press.

Conti, S., Held, H., Pach, M., Rumpf, M., & Schultz, R. (2009). Shape Optimization under Uncertainty – A Stochastic Programming Perspective. SIAM Journal on Optimization, 19(4), 1610-1632.

Cura, T. (2009). Particle Swarm Optimization Approach to Portfolio Optimization. Nonlinear Analysis: Real World Applications, 10(4), 2396-2406.

Crandall, D., Owens, A., Snavely, N., & Huttenlocher, D. (2011). Discrete-Continous Optimization for Large-Scale Structure from Motion. Paper Presented at the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Cuomo, C. A. Desjardins, C. A. Bakowski, M. A., Goldberg, J., Ma, A. T., Becnel, J. J., Levin, J. Z. (2012). Microsporidian Genome Analysis Reveals Evolutionary Strategies for Obligate Intracellular Growth. Genome Research, 22(12), 2478-2488.

Davenport, T. H. (2013). Process Innovation: Reengineering Work through Information Technology: Harvard Business Press.

Davoodi, E., Hagh, M. T., & Zadeh, S. G. (2014). A Hybrid Improved Quantum-Behaved Particle Swarm Optimization-Simplex Method (IQPSOS) to Solve Power System Load Flow Problems. Applied Soft Computing, 21, 171-179.

De Castro, L. N. (2007). Fundamentals of Natural Computing: An Overview. Physics of Life Reviews, 4(1), 1-36.

Deb, K. (2014). Multi-Objective Optimization Search Methodologies (pp. 403-449): Springer.

Delnavaz, B. (2014). Forecasting of the Stock Price Index by using Fuzzy-Neural Network And Genetic Algorithms. Journal of Applied Science and Agriculture, 9, 109-117.

Dewangan, S., Naik, A., & Agrawal, A. (2014). Study of Nature Inspired Computing.

Dhouib, S. (2010). A Multi Start Great Deluge Metaheuristic for Engineering Design Problems. Paper Presented at the International Conference on Computer Systems and Applications (AICCSA), 2010 IEEE/ACS.

Di Caro, G., & Dorigo, M. (1998). AntNet: Distributed Stigmergetic Control for Communications Networks. Journal of Artificial Intelligence Research, 317-365.

Dodig-Crnkovic, G. (2012). Info-Computationalism and Morphological Computing of Informational Structure Integral Biomathics 97-104: Springer.

Dorigo, M. (1992). Optimization, Learning and Natural Algorithms. Ph. D. Thesis, Politecnico di Milano, Italy.

Dowsland, K. A., & Thompson, J. M. (2012). Simulated Annealing Handbook of Natural Computing 1623-1655: Springer.

Ducatelle, F., Di Caro, G. A., & Gambardella, L. M. (2010). Principles and Applications of Swarm Intelligence for Adaptive Routing in Telecommunications Networks. Swarm Intelligence, 4(3), 173-198.

Eberhart, R. C., & Kennedy, J. (1995). A New Optimizer using Particle Swarm Theory. Paper Presented at the Proceedings of the 6th International Symposium on Micro Machine and Human Science.

Emary, E., Zawbaa, H. M., & Grosan, C. (2018). Experienced Gray Wolf Optimization Through Reinforcement Learning and Neural Networks. IEEE Transactions on Neural Networks and Learning Systems, 29(3), 681-694.

Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016). Binary Grey Wolf Optimization Approaches for Feature Selection. Neurocomputing, 172, 371-381.

Eswaramoorthy, S., Sivakumaran, N., & Sekaran, S. (2016). Grey Wolf Optimization Based Parameter Selection for Support Vector Machines. COMPEL-The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, 35(5), 1513-1523.

Faludi, A. (2013). A Reader in Planning Theory: Elsevier.

Faris, H., Aljarah, I., Al-Betar, M. A., & Mirjalili, S. (2018). Grey Wolf Optimizer: A Review of Recent Variants and Applications. Neural Computing and Applications, 30(2), 413-435.

Farmer, J. D., Packard, N. H., & Perelson, A. S. (1986). The Immune System, Adaptation, And Machine Learning. Physica D: Nonlinear Phenomena, 22(1), 187-204.

Fasanghari, M., Iranmanesh, S. H., & Amalnick, M. S. (2015). Predicting the Success of Projects Using Evolutionary Hybrid Fuzzy Neural Network Method in Early Stages. Journal of Multiple-Valued Logic & Soft Computing, 25(2-3), 291-321.

Feist, A. M., & Palsson, B.O. (2010). The Biomass Objective Function. Current Opinion in Microbiology, 13(3), 344-349.

Fisher, D. H. (1987). Knowledge Acquisition via Incremental Conceptual Clustering. Machine Learning, 2(2), 139-172.

Fister Jr, I., Yang, X.-S., Fister, I., Brest, J., & Fister, D. (2013). A Brief Review of Nature-Inspired Algorithms for Optimization. arXiv preprint arXiv:1307.4186.

Galbally, J., Fierrez, J., & Ortega-Garcia, J. (2007). Bayesian Hill-Climbing Attack and Its Application to Signature Verification. Advances in Biometrics (pp. 386-395): Springer.

Gamil, A. A., Elfouly, R. S., & Darwish, N. M. (2007). Stock Technical Analysis using Multi Agent and Fuzzy Logic. Paper presented at the World Congress on Engineering, 2007.

Garakani, A. R., & Branch, S. T. (2018). Stock Price Prediction Using Multilayer Perceptron Neural Network by Monitoring Frog Leaping Algorithm. Journal of Intelligent Computing Volume, 9(1), 15-23.

Gentle, J. E. (2013). Random Number Generation and Monte Carlo Methods: Springer Science & Business Media.

Gigenrenzer, G., & Gaissmaier, W. (2011). Heuristic Decision Making. Annual Review of Psychology, 62, 451-482.

Göçken, M., Özçalıcı, M., Boru, A., & Dosdoğru, A. T. (2016). Integrating Metaheuristics and Artificial Neural Networks for Improved Stock Price Prediction. Expert Systems with Applications, 44, 320-331.

Goyal, S., & Patterh, M. S. (2016). Modified Bat Algorithm for Localization of Wireless Sensor Network. Wireless Personal Communications, 86(2), 657-670.

Gupta, S., & Deep, K. (2017). Performance of Grey Wolf Optimizer on Large Scale Problems. AIP Conference Proceedings, 1802(1), 020005.

Gratton, S., Lawless, A. S., & Nicholas, N. K. (2007). Approximate Guass-Newton Methods for Nonlinear Least Squares Problems. SIAM Journal on Optimization, 18(1), 106-132.

Hafezi, R., Shahrabi, J., & Hadavandi, E. (2015). A Bat-Neural Network Multi-Agent System (BNNMAS) for Stock Price Prediction: Case Study of DAX Stock Price. Applied Soft Computing, 29, 196-210.

Haftka, R. T., & Gürdal, Z. (2012). Elements of Structural Optimization (Vol. 11): Springer Science & Business Media.

Hassan, M. R., & Nath, B. (2005). Stock Market Forecasting Using Hidden Markov Model: A New Approach. Paper Presented at the Proceedings of the 5th International Conference on Intelligent Systems Design and Applications.

Haykin, S. (1994). Neural Networks, a Comprehensive Foundation (0023527617). Macmilan, USA.

Haykin, S. S., Haykin, S. S., Haykin, S. S., & Haykin, S. S. (2009). Neural Networks and Learning Machines (Vol. 3): Pearson Upper Saddle River, NJ, USA.

Hemamalini, S., & Simon, S.P. (2011). Dynamic Economic Dispatch using Artificial Immune System for Units with Valve-Point Effect. International Journal of Electrical Power & Energy Systems, 33(4), 868-874.

Higle, J. L., & Sen, S. (2013). Stochastic Decomposition: A Statistical Method for Large Scale Stochastic Linear Programming (Vol. 8): Springer Science & Business Media.

Hirvensalo, M. (2013). Quantum Computing: Springer.

Hoffman, J. (2010). A Heuristic for Domain Independent Planning and Its Use in an Enforced Hill-Climbing Algorithm. Foundations of Intelligent Systems (pp. 216-227): Springer.

Horst, R., & Tuy, H. (2013). Global Optimization: Deterministic Approaches: Springer Science & Business Media.

Hu, Y., Liu, K., Zhang, X., Su, L., Ngai, E., & Liu, M. (2015). Application of Evolutionary Computation for Rule Discovery in Stock Algorithmic Trading: A Literature Review. Applied Soft Computing, 36, 534-551.

Huang, Z-H., & Ni, T. (2010). Smoothing Algorithms for Complementarity Problems over Symmetric Cones. Computational Optimization and Applications, 45(3), 557-579.

Hughes, T. J. (2012). The Finite Element Method: Linear Static and Dynamic Finite Element Analysis: Courier Corporation.

Jia, L., Yang, B., Park, D. H., Tan, F., & Park, M. (2013). Software Maintainability Prediction Model Based on Fuzzy Neural Network. Journal of Multiple-Valued Logic & Soft Computing, 20(1-2), 39-53.

Jothimani, D., Shankar, R., & Yadav, S. S. (2016). A Comparative Study of Ensemble-Based Forecasting Models for Stock Index Prediction. Proceedings of Proceedings of the Eleventh Midwest Association for Information Systems Conference (MWAIS), Milwaukee, Wisconsin, May 19-20, 1-7.

Kaastra, I., & Boyd, M. (1996). Designing a Neural Network for Forecasting Financial and Economic Time Series. Neurocomputing, 10(3), 215-236.

Karaboga, D., & Bastruk, B. (2007). A Powerful and Efficient Algorithm for Numeric Function Optimization: Artificial Bee Colony (ABC) Algorithm. Journal of Global Optimization, 39(3), 459-471.

Kavousi-Fard, A., Samet, H., & Marzbani, F. (2014). A New Hybrid Modified FireFly Algorithm and Support Vector Regression Model for Accurate Short Term Load Forecasting. Expert Systems with Applications, 41(13), 6047-6056.

Kefi, S., Rokbani, N., Krömer, P., & Alimi, A.M. (2015). A New Ant Supervised-PSO Variant Applied to Travelling Salesman Problem. Paper Presented at the Hybrid Intelligent Systems: 15th International Conference HIS 2015 on Hybrid Intelligent Systems, Seoul, South Korea, November 16-18, 2015.

Kennedy, J. (2010). Particle Swarm Optimization Encyclopedia of Machine Learning (pp. 760-766): Springer.

Kennedy, J., Kennedy, J. F., Eberhart, R. C., & Shi, Y. (2001). Swarm Intelligence: Morgan Kaufmann.

Khajavi, S., & Amiri, F. S. (2017). Prediction of Stock Price Using Particle Swarm Optimization Algorithm and Box-Jenkins Time Series. International Journal of Finance & Managerial Accounting, 2(7), 25-31.

Khan, Z. H., Alin, T. S., & Hussain, M. A. (2011). Price Prediction of Share Market Using Artificial Neural Network (ANN). International Journal of Computer Applications, 22(2), 42-47.

Kifah, S. & Abdullah, S. (2015). An Adaptive Non-Linear Great Deluge Algorithm for the Patient-Admission Problem. Information Sciences, 295, 573-585.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. Science, 220(4598), 671-680.

Kothari, D. P. (2012). Power System Optimization. Paper Presented at the 2nd International Conference on Computational Intelligence and Signal Processing (CISP), 2012.

Kouvelis, P., & Yu, G. (2013). Robust Discrete Optimization and its Applications, Vol. 14: Springer Science & Business Media.

Kozdraj, T. (2009). Using Artificial Neural Networks to Predict Stock Prices. Acta Universitatis Lodziensis. Folia Oeconomica (Acta Universitatis Lodziensis). Folia Oeconomica, 225, 295-308

Kuhn, H. W. (2014). Nonlinear Programming: A Historical View Traces and Emergence of Nonlinear Programming (pp. 393-414): Springer.

Kumar, T. S. (2014). Introduction to Data Mining: Pearson Education India.

Kumbharana, N., & Pandey, G. M. (2013). A Comparative Study of ACO, GA and SA for Solving Travelling Salesman Problem. International Journal of Societal Applications of Computer Science, 2(2), 224-228.

Kuo, R. J., Chen, C., & Hwang, Y. (2001). An Intelligent Stock Trading Decision Support System Through Integration of Genetic Algorithm Based Fuzzy Neural Network and Artificial Neural Network. Fuzzy Sets and Systems, 118(1), 21-45.

Lahmiri, S., & Boukadoum, M. (2015). An Ensemble System Based on Hybrid EGARCH-ANN with Different Distributional Assumptions to Predict S&P 500 Intraday Volatility. Fluctuation and Noise Letters, 14(01), 1-10.

Lawrence, R. (1997). Using Neural Networks to Forecast Stock Market Prices. University of Manitoba, 333, 1-21.

Le, C. V., Gilbert, M., & Askes, H. (2009), Limit Analysis of Plates using the EFG Method and Second Order Cone Programming. International Journal for Numerical Methods in Engineering, 78(13), 1532-1552.

Ledesma, S., Aviña, G., & Sanchez, R. (2008). Practical Considerations for Simulated Annealing Implementation. Simulated Annealing, 20, 401-420.

Lee, J., & Leyffer, S. (2011). Mixed Integer Nonlinear Programming (Vol. 154): Springer Science & Business Media.

Lertyingyod, W., & Benjamas, N. (2016). Stock Price Trend Prediction Using Artificial Neural Network Techniques Case Study: Thailand Stock Exchange. Paper Presented at the 2016 Computer Science and Engineering Conference (ICSEC).

Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010). A Contextual-Bandit Approach to Personalized News Article Recommendation. Paper Presented at the Proceedings of the 19th International Conference on World Wide Web.

Li, M., Du, W., & Nian, F. (2014). An Adaptive Particle Swarm Optimization Algorithm Based on Directed Weighted Complex Network. Mathematical Problems in Engineering, 2014(2014), 1-7.

Li, Q., Chen, H., Huang, H., Zhao, X., Cai, Z., Tong, C. & Tian, X. (2017). An Enhanced Grey Wolf Optimization Based Feature Selection Wrapped Kernel Extreme Learning Machine for Medical Diagnosis. Computational and Mathematical Methods in Medicine, 2017, 1-15.

Long, W., Liang, X., Cai, S., Jiao, J., & Zhang, W. (2017). A Modified Augmented Lagrangian with Improved Grey Wolf Optimization to Constrained Optimization Problems. Neural Computing and Applications, 28(1), 421-438.

Lopez, M. I., Luna, J., Romero, C., & Ventura, S. (2012). Classification via Clustering for Predicting Final Marks Based on Student Participation in Forums. International Educational Data Mining Society, Paper presented at the 5th International Conference on Educational Data Mining (EDM), Chania, Greece.

Luo, Z.-Q., Pang, J.-S., & Ralph, D. (1996). Mathematical Programs with Equilibrium Constraints: Cambridge University Press.

Machairas, V., Tsangrassoulis, A., & Axarli, K. (2014). Algorithms for Optimization of Building Design: A Review. Renewable and Sutainable Energy Reviews, 31, 101-112.

Mahadavi, M., Fesanghary, M., & Damangir, E. (2007). An Improved Harmony Search Algorithm for Solving Optimization Problems. Applied Mathematics and Computation, 188(2), 1567-1579.

Mahajan, K. S., Jamsandekar, S. S., & Kulkarni, R. (2015). Portfolio Investment Model Using Neuro Fuzzy System. International Journal of Computer Science and Information Technology (IJCSIT), 6(2), 1819-1823.

Mahale, R. A., & Chavan, S. (2012). A Survey: Evolutionary and Swarm Based Bio-Inspired Optimization Algorithms. International Journal of Scientific and Research Publications, 2(12), 1-6.

Maier, H. R., & Dandy, G. C. (2000). Neural Networks for the Prediction and Forecasting of Water Resources Variables: A Review of Modelling Issues and Applications. Environmental Modelling & Software, 15(1), 101-124.

Mäkisara, K., Simula, O., Kangas, J., & Kohonen, T. (2014). Artificial Neural Networks (Vol. 2): Elsevier.

Malek, M., Guruswamy, M., Pandya, M., & Owens, H. (1989). Serial and Parallel Simulated Annealing and Tabu Search Algorithms for the Traveling Salesman Problem. Annals of Operations Research, 21(1), 59-84.

Manjarres, D., Landa-Torres, I., Gil-Lopez, S., Del Ser, J., Bilbao, M. N., Salcedo-Sanz, S., & Geem, Z. W. (2013). A Survey on Applications of the Harmony Search Algorithm. Engineering Applications of Artificial Intelligence, 26(8), 1818-1831.

Mathworks. (2012a). Neural Network Toolbox User’s Guide. Matlab, Mathworks, R2012a.

Mcmullan, P. (2007). An Extended Implementation of the Great Deluge Algorithm for Course Timetabling. Computational Science-ICCS 2007 (pp. 538-545): Springer.

Merh, N., Saxena, V., & Pardasani, K. R. (2011). Next Day Stock market Forecasting: An Application of ANN and ARIMA. IUP Journal of Applied Finance, 17(1), 70.

Miller, B. M., & Rubinovich, E. Y. (2012). Impulsive Control in Continuous and Discrete-Continuous Systems: Springer Science & Business Media.

Minton, S., Johnston, M.D., Philips, A. B., & Laird, P. (1992). Minimizing Conflicts: A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems. Artificial Intelligence, 58(1-3), 161-205.

Mirjalili, S. (2015). How Effective is the Grey Wolf Optimizer in Training Multi-Layer Perceptrons. Applied Intelligence, 43(1), 150-161.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014a). . Let a Biogeography-Based Optimizer Train Your Multi-Layer Perceptron. Information Sciences, 269, 188-209.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014b). Grey Wolf Optimizer. Advances in Engineering Software, 69, 46-61.

Mittal, N., Singh, U., & Sohi, B. S. (2016). Modified Grey Wolf Optimizer for Global Engineering Optimization. Applied Computational Intelligence and Soft Computing, 8(1), 1-16.

Mohamed, A. A. A., El-Gaafary, A. A., Mohamed, Y. S., & Hemeida, A. M. (2015). Design Static VAR Compensator Controller Using Artificial Neural Network Optimized by Modify Grey Wolf Optimization. Paper Presented on 2015 International Joint Conference on Neural Networks.

Mohapatra, P., & Das, S. (2013). Stock Market Prediction using Bio-Inspired Computing: A Survey. International Journal of Engineering Science and Technology, 5(4), 739-746.

Monteiro, C., Ramirez-Rosado, I. J., Fernandez-Jimenez, L. A., & Ribeiro, M. (2018). New Probabilistic Price Forecasting Models: Application to the Iberian Electricity Market. International Journal of Electrical Power & Energy Systems, 103, 483-496.

Morais, H., Kádár, P., Faria, P., Vale, Z. A., & Khodr, H. M. (2010). Optimal Scheduling of a Renewable Micro-Grid in an Isolated Load Area using Mixed-Integer Linear Programming. Renewable Energy, 35(1), 151-156.

Morales, J. L., & Nocedal, J. (2011). Remark on “Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-Scale Bound Constrained Optimization”. ACM Transactions on Mathematical Software (TOMS), 38(1), 7.

Mosavi, M. R., Khishe, M., & Ghamgosar, A. (2016). Classification of Sonar Data Set Using Neural Network Trained by Gray Wolf Optimization. Neural Network World, 26(4), 393-415.

Motwani, R., & Raghavan, P. (2010). Randomized Algorithms: Chapman & Hall/CRC.

Nabeel, R. (2010). Hybrid Genetic Algorithms with Great Deluge for Course Timetabling. International Journal of Computer Science and Network Security, 10, 283-288.

Nahas, N., Kadi, D. A., & El Fath, M. N. (2010). Iterated Great Deluge for the Dynamic Facility Layout Problem: CIRRELT.

Narayanan, B., & Govindarajan, M. (2015). Prediction of Stock Market Using Ensemble Model. International Journal of Computer Applications, 128(1), 18-21.

Navale, G., Dudhwala, N., Jadhav, K., Gabda, P., & Vihangam, B. K. (2016). Prediction of Stock Market Using Data Mining and Artificial Intelligence. International Journal of Engineering Science, 6(6), 6539-6544.

Negnevitsky, M. (2005). Artificial intelligence: A Guide to Intelligent Systems: Pearson Education, USA.

Nemhauser, G., & Bienstock, D. (2005). Integer Programming and Combinatorial Optimization: Springer.

Nguyen, T. H., Shirai, K., & Velcin, J. (2015). Sentiment Analysis on Social Media for Stock Movement Prediction. Expert Systems with Applications, 42(24), 9603-9611.

Niu, M., Wang, Y., Sun, S., & Li, Y. (2016). A Novel Hybrid Decomposition-and-Ensemble Model Based on CEEMD and GWO for Short-Term PM2. 5 Concentration Forecasting. Atmospheric Environment, 134, 168-180.

Nozohour-leilabady, B., & Fazelabdolabadi, B. (2016). On the Application of Artificial Bee Colony (ABC) Algorithm for Optimization of Well Placements in Fractured Reservoirs; Efficiency Comparison with the Particle Swarm Optimization (PSO) Methodology. Petoleum, 2(1), 79-89.

Nur, I. M., & Ülker, E. (2018). A Hybrid Cloud-Based Intrusion Detection and Response System (IDRS) Based on Grey Wolf Optimizer (GWO) and Neural Network (NN). Paper Presented on 2018 International Conference on Cyber Security and Computer Science (ICONCS'18).

Ojha, V. K., Abraham, A., & Snášel, V. (2017). Metaheuristic Design of Feed-Forward Neural Networks: A Review of Two Decades of Research. Engineering Applications of Artificial Intelligence, 60, 97-116.

Osaba, E., Yang, X.-S., Diaz, F., Lopez-Garcia, P., & Carballedo, R. (2016). An Improved Discrete Bat Algorithm for Symmetric and Asymmetric Traveling Salesman Problems. Engineering Applications of Artificial Intelligence, 48, 59-71.

Osman, I. H., & Kelly, J. P. (2012). Meta-Heuristics: Theory and Applications: Springer Science & Business Media.

Özcan, E., Misir, M., Ochoa, G., & Burke, E. K. (2012). A Reinforcement Learning: Great-Deluge Hyper-Heuristic. Modeling, Analysis, and Applications in Metaheuristic Computing: Advancements and Trends, 34.

Pandiri, V., & Singh, A. (2015). Swarm Intelligence Approaches for Multidepot Salesmen Problems with Load Balancing. Applied Intelligence, 1-13.

Park, C., Pan, J., & Manocha, D. (2013). Real-Time Optimization-Based Planning in Dynamic Environments using GPUs. Paper Presented at the IEEE International Conference on Robotics and Automation (ICRA), 2013.

Parpinelli, R. S., & Lopes, H. S. (2011). New Inspirations in Swarm Intelligence: A Survey. International Journal of Bio-Inspired Computation, 3(1), 1-16.

Parsian, A., Ramezani, M., & Ghadimi, N. (2017). A Hybrid Neural Network-Gray Wolf Optimization Algorithm for Melanoma Detection. Biomedical Research, 28(8), 3408-3411.

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting Stock and Stock Price Index Movement Using Trend Deterministic Data Preparation and Machine Learning Techniques. Expert Systems with Applications, 42(1), 259-268.

Păun, G. (2012). Membrane Computing. Handbook of Natural Computing, 1355-1377.

Pereira, G. (2011). Particle Swarm Optimization. INESCID and Instiuto Superior Tecnico, Porto. Available at: https://pdfs.semanticscholar.org/0033/9e89348fdda5b019591 de93f0c6bf229bf4b.pdf. 93f0c6bf229bf4b.pdf. Accessed on 13/12/2018

Pierre, D. M., Zakaria, N., & Pal, A. J. (2011). Master-Slave Parallel Vector-Evaluated Genetic Algorithm for Unmanned Aerial Vehicle's Path Planning. Paper Presented at the 11th International Conference on Hybrid Intelligent Systems (HIS), 2011.

Poli, R., (2007). An Analysis of Publications on Particle Swarm Optimization Applications. Essex, UK: Department of Computer Science, University of Essex.

Prakasam, A., & Savarimuthu, N. (2015). Metaheuristic Algorithms and Probabilistic Behaviour: A Comprehensive Analysis of Ant Colony Optimization and its Variants. Artificial Intelligence Review, 1-34.

Puntura, A., Theera-Umpon, N., & Auephanwiriyakul, S. (2016). Optimizing Support Vector Machine Parameters Using Cuckoo Search Algorithm via Cross Validation. Paper Presented at the 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE).

Rani, D., Jain, S. K., Srivastava, D. K., & Perumal, M. (2012). 3 Genetic Algorithms and Their Applications to Water Resources Systems. Metaheuristics in Water, Geotechnical and Transport Engineering, 43.

Rao, R. V., Savsani, V. J., & Vakharia, D. (2012). Teaching-Learning-Based Optimization: An Optimization Method for Continuous Non-Linear Large Scale Problems. Information Sciences, 183(1), 1-15.

Rather, A. M., Agarwal, A., & Sastry, V. (2015). Recurrent Neural Network and A Hybrid Model for Prediction of Stock Returns. Expert Systems with Applications, 42(6), 3234-3241.

Razali, N. M., & Geraghty, J. (2011). Genetic Algorithm Performance with Different Selection Strategies in Solving TSP. Paper Presented at the Proceedings of the World Congress on Engineering.

Rios, L. M., & Sahinidis, N. V. (2013). Derivative-Free Optimization: A Review of Algorithms and Comparison of Software Implementations. Journal of Global Optimization, 56(3), 1247-1293.

Rodriguez-Lujan, I., Huerta, R., Elkan, C., & Cruz, C. S. (2010). Quadratic Programming Feature Selection. The Journal of Machine Learning Research, 11, 1491-1516.

Rojas, R. (1996). The Back-propagation Algorithm. Neural Networks, Springer, 149-182.

Rozenberg, G., Bck, T., & Kok, J. N. (2011). Handbook of Natural Computing: Springer Publishing Company, Incorporated.

Safari, S. A. (2015). Wildlife and Big 5. http://sasafari.com/wildlife-and-the-big-five/, Accesed on 12/12/2018.

Sagar S. Nika. (2015). A Comparative Study of Classification Techniques in Data Mining Algorithms. Oriental Journal of Computer Science & Technology, 8(1), 13-19.

Said, Y., & Wegman, E. (2009). Roadmap for Optimization. Wiley Interdisciplinary Reviews: Computational Statistics, 1(1), 3-17.

Samadzadegan, F., Soleymani, A., & Abbaspour, R. A. (2010). Evaluation of Genetic Algorithms for Tuning SVM Parameters in Multi-Class Problems. Paper Presented at the 11th International Symposium on Computational Intelligence and Informatics (CINTI).

Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. Neural Networks, 61, 85-117.

Seidy, E. E. (2016). A New Particle Swarm Optimization Based Stock Market Prediction Technique. International Journal of Advanced Computer Science and Applications, 7(4), 322-327.

Senthilnath, J., Das, V., Omkar, S., & Mani, V. (2012). Clustering Using Lévy Flight Cuckoo Search. Paper Presented at the BIC-TA (2).

Sivaramakrishnan, K. K. (2002). Linear Programming Approaches to Semidefinite Programming Problems. Renselaer Polytechnic Institute.

Sharma, A, Sharma, A., Panigrahi, B., Kiran, D., & Kumar, R. (2016). Ageist Spider Monkey Optimization Algorithm. Swarm and Evolutionary Computation.

Sniedovich, M. (2010). Dynamic Programming Foundations and Principles: CRC Press.

Song, X., Tang, L., Zhao, S., Zhang, X., Li, L., Huang, J., & Cai, W. (2015). Grey Wolf Optimizer for Parameter Estimation in Surface Waves. Soil Dynamics and Earthquake Engineering, 75, 147-157.

Srivastava, R. K., Greff, K., & Schmidhuber, J. (2015). Training Very Deep Networks. Advances in Neural Information Processing Systems, 2377-2385.

Streiner, D. L., Norman, G. R., & Cairney, J. (2014). Health Measurement Scales: A Practical Guide to Their Development and Use: Oxford University Press.

Stützle, T., López-Ibáñez, M., & Dorigo, M. (2011). A Concise Overview of Applications of Ant Colony Optimization. Wiley Encyclopedia of Operations Research and Management Science.

Suh, S. (2012). Practical Applications of Data Mining: Jones & Bartlett Publishers, USA.

Sujatha, K., Punithavathani, DS. (2018). Optimized Ensemble Decision-Based Multi-Focus Imagefusion Using Binary Genetic Grey-Wolf Optimizer in Camera Sensor Networks. Multimedia Tools Application, 77(2), 1735-1759.

Tambouratzis, T. (1997). A Simulated Annealing Artificial Neural Network Implementation of the N-Queens Problem. International Journal of Intelligent Systems, 12(10), 739-751.

Tanweer, M., Suresh, S., & Sundarajan, N. (2015). Improved SRPSO Algorithm for Solving CEC 2015 Computationally Expensive Numerical Optimization Problems. Paper Presented at the 2015 IEEE Congress on Evolutionary Computation (CEC).

Thiele, L., Miettinen, K., Korhonen, P. J., & Molina, J. (2009). A Preference-Based Evolutionary Algorithm for Multi-Objective Optimization. Evolutionary Computation, 17(3), 411-436.

Tilakaratne, C. D., Mammadov, M. A., & Morris, S. A. (2009). Modified Neural Network Algorithms for Predicting Trading Signals of Stock Market Indices. Advances in Decision Sciences, 2009(125308), 1-22.

Toga, A. W., Clark, K. A., Thompson, P. M., Shattuch, D. W., & Van Horn, J. D. (2012). Mapping the Human Connectome. Neurosurgery, 71(1), 1.

Tuba, M. Subotic, M., & Stanarevic, N. (2011). Modified Cuckoo Search Algorithm for Unconstrained Optimization Problems. Paper Presented at the Proceedings of the 5th European Conference on European Computing Conference.

Tuba, E., Tuba, M., & Simian, D. (2016a). Adjusted Bat Algorithm for Tuning of Support Vector Machine Parameters. Paper Presented at the IEEE Congress on Evolutionary Computation (CEC).

Tuba, E., Mrkela, L., & Tuba, M. (2016b). Support Vector Machine Parameter Tuning using Firefly Algorithm. Paper Presented at the 26th International Conference on Radio-Electronics.

Turabieh, H. (2016). A Hybrid ANN-GWO Algorithm for Prediction of Heart Disease. American Journal of Operations Research, 6(02), 136-146.

Van den Berg, J., Kaymak, U., & Van den Bergh, W.-M. (2004). Financial Markets Analysis by Using a Probabilistic Fuzzy Modelling Approach. International Journal of Approximate Reasoning, 35(3), 291-305.

Van Laarhoven, P. J., Aarts, E.H., & Lenstra, J. K. (1992). Job Shop Scheduling by Simulated Annealing. Operations Research, 40(1), 113-125.

Vazan, P., & Tanuska, P. (2012). A Short Reflection on the Strenghts and Weaknesses of Simulation Optimization. Paper Presented at the Proceedings of World Academy of Science, Engineering and Technology.

Venter, G. (2010). Review of Optimization Techniques. Encyclopedia of Aerospace Engineering.

Vincent, O., & Bamiro, K. (2013). Fluctuations in Stock Market Prices: What Went Wrong, Its Implications to Nigerian Economy? International Journal of Computer Applications, 63(19), 13-20.

Wang, G. G., Gandomi, A. H., Zhao, X., & Chu, H. C. E. (2016). Hybridizing Harmony Search Algorithm with Cuckoo Search for Global Numerical Optimization. Soft Computing, 20(1), 273-285.

Wang, J.-J., Wang, J.-Z., Zhang, Z.-G., & Guo, S.-P. (2012). Stock Index Forecasting Based on a Hybrid Model. Omega, 40(6), 758-766.

Wei, L.-Y. (2013). A Hybrid Model Based on ANFIS and Adaptive Expectation Genetic Algorithm to Forecast TAIEX. Economic Modelling, 33, 893-899.

Wei, Y., Ni, N., Liu, D., Chen, H., Wang, M., Li, Q. & Ye, H. (2017). An Improved Grey Wolf Optimization Strategy Enhanced SVM and Its Application in Predicting the Second Major. Mathematical Problems in Engineering, 2017(1), 1-12.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, USA.

Wolkowicz, H., Saigal, R., & Vandenberghe, L. (2012). Handbook of Semidefinite Programming: Theory, Algorithms, and Applications (Vol: 27): Springer Science & Business Media.

Wolpert, D. H., & McReady, W. G. (1997). No Free Lunch Theorems for Optimization. IEEE Transaction on Evolutionary Computation, 1, 67-82.

Wolsey, L.A., & Nemhauser, G. L. (2014). Integer and Combinatorial Optimization: John Wiley & Sons.

Wong, L., & Moin, N. H. (2015). Enhanced Ant Colony Optimization for Inventory Routing Problem. Paper Presented at the 22nd National Symposium on Mathematical Sciences (SKSM22): Strengthening Research and Collaboration of Mathematical Sciences in Malaysia.

Wong, W.-K., Manzur, M., & Chew, B.-K. (2012). How Rewarding is Technical Analysis? Evidence from Singapore Stock Market. Applied Financial Economics, 13(7), 543-551.

Wooldridge, J. M. (2010). Econometric Analysis of Cross Section and Panel Data: MIT Press.

Wu, J., & Coggeshall, S. (2012). Foundations of Predictive Analytics: CRC Press, USA.

Wu, K.-P., Wu, Y.-P., & Lee, H.-M. (2014). Stock Trend Prediction by Using K-Means and AprioriAll Algorithm for Sequential Chart Pattern Mining. Journal of Information Science & Engineering, 30(3), 669-686.

Li, X.-G., & Wei, X. (2008). An Improved Genetic Algorithm-Simulated Annealing Hybrid Algorithm for the Optimization of Multiple Reservoirs. Water Resources Management, 22(8), 1031-1049.

Xi, B., Liu, Z., Raghavachari, M., Xia, C. H., & Zhang, L. (2004). A Smart Hill-Climbingh Algorithm for Application Server Configuration. Paper Presented at the Proceedings of the 13th International Conference on World Wide Web.

Xie, C., Lin, D.-Y., & Waller, S.T. (2010). A Dynamic Evacuation Network Optimization Problem with Lane Reversal and Crossing Elimination Strategies. Transportation Research Part E: Logistics and Transportation Review, 46(3), 295-316.

Xing, F. Z., Cambria, E., & Welsch, R. E. (2017). Natural Language Based Financial Forecasting: A Survey. Artificial Intelligence Review, 1-25.

Xiong, T., Bao, Y., & Hu, Z. (2014). Multiple-Output Support Vector Regression with a Firefly Algorithm for Interval-Valued Stock Price Index Forecasting. Knowledge-Based Systems, 55, 87-100.

Xu, H., Caramanis, C., & Mannor, S. (2012). Sparse Algorithms Are Not Stable: A No-Free-Lunch Theorem. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(1), 187-193.

Xu, R., & Wunsch, D. (2005). Survey of Clustering Algorithms. IEEE Transactions on Neural Networks, 16(3), 645-678.

Yang, B., Zhang, X., Yu, T., Shu, H., & Fang, Z. (2017). Grouped Grey Wolf Optimizer for Maximum Power Point Tracking of Doubly-Fed Induction Generator Based Wind Turbine. Energy Conversion and Management, 133, 427-443.

Yang, C.-T., Sung, T.-C., & Weng, W.-C. (2006). An Improved Tabu Search Approach with Mixed Objective Function for One-Dimensional Cutting Stock Problems. Advances in Engineering Software, 37(8), 502-513.

Yang, X.-S. (2010). A New Metaheuristic Bat-Inspired Algorithm Nature Inspired Cooperative Strategies for Optimization (NICSO 2010) 65-74: Springer.

Yang, X.-S. (2011). Review of Meta-Heuristics and Generalised Evolutionary Walk Algorithm. International Journal of Bio-Inspired Computation, 3(2), 77-84.

Yang, X.-S. (2012). Swarm-Based Metaheuristic Algorithms and No-Free-Lunch Theorems: Theory and New Applications of Swarm Intelligence.

Yang, X.-S., Deb, S., & Fong, S. (2014). Metaheuristic algorithms: Optimal Balance of Intensification and Diversification. Applied Mathematics & Information Sciences, 8(3), 977.

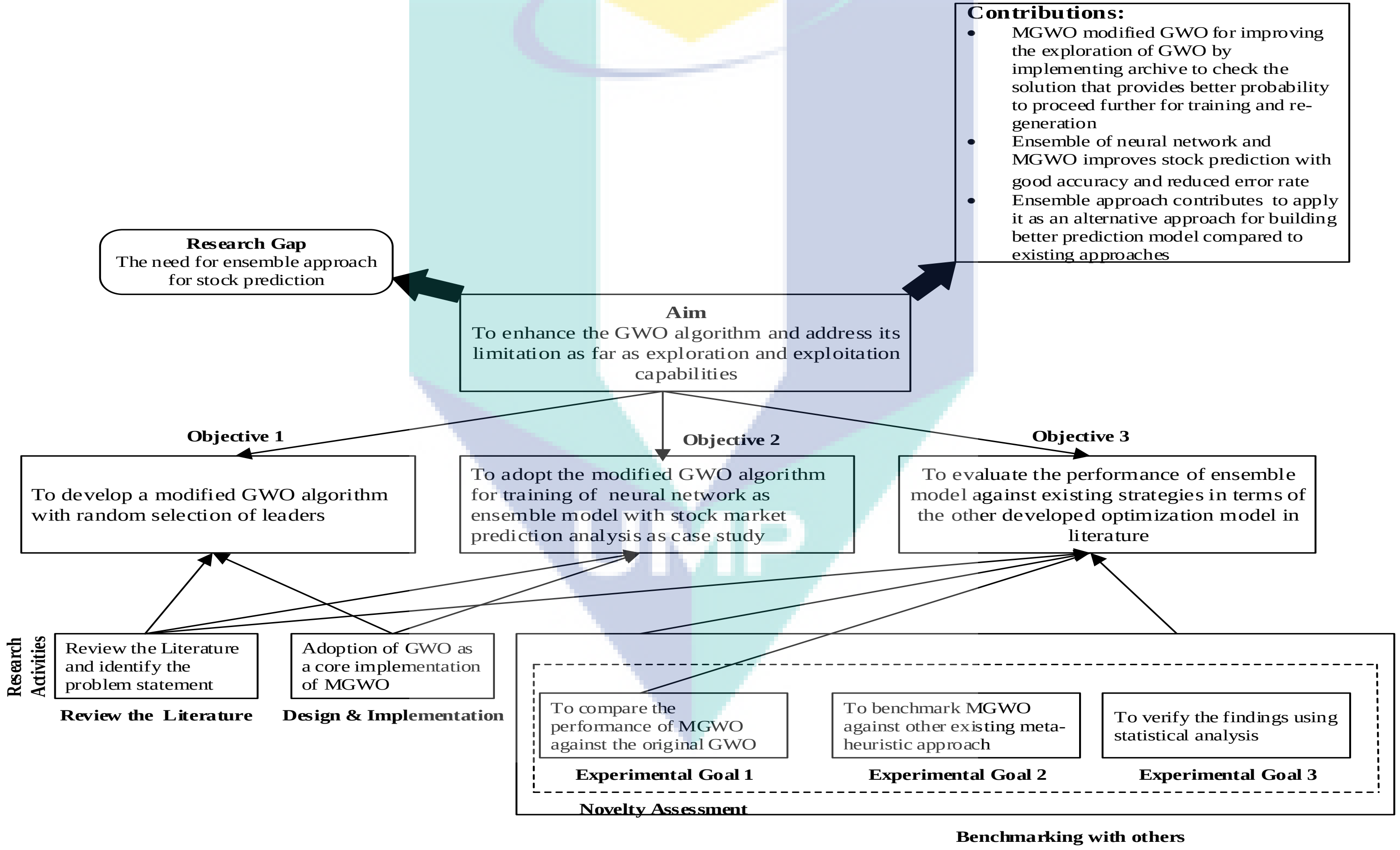
Yetis, Y., Kaplan, H., & Jamshidi, M. (2014). Stock Market Prediction by Using Artificial Neural Network. Paper Presented at the 2014 World Automation Congress (WAC).

Zang, H., Zhang, S., & Hapeshi, K. (2010). A Review of Nature-Inspired Algorithms. Journal of Bionic Engineering, 7(4), S232-S237.

Zhang, Y., Phillips, P., Wang, S., Ji, G., Yang, J., & Wu, J. (2016). Fruit Classification by Biogeography- Based Optimization and Feed-Forward Neural Network. Expert Systems, 33(3), 239-253.

APPENDIX A

RESEARCH MAPPING



Contributions:

- MGWO modified GWO for improving the exploration of GWO by implementing archive to check the solution that provides better probability to proceed further for training and re-generation
- Ensemble of neural network and MGWO improves stock prediction with good accuracy and reduced error rate
- Ensemble approach contributes to apply it as an alternative approach for building better prediction model compared to existing approaches

Research Gap
The need for ensemble approach for stock prediction

Aim
To enhance the GWO algorithm and address its limitation as far as exploration and exploitation capabilities

Objective 1
To develop a modified GWO algorithm with random selection of leaders

Objective 2
To adopt the modified GWO algorithm for training of neural network as ensemble model with stock market prediction analysis as case study

Objective 3
To evaluate the performance of ensemble model against existing strategies in terms of the other developed optimization model in literature

Research Activities

Review the Literature and identify the problem statement

Review the Literature

Adoption of GWO as a core implementation of MGWO

Design & Implementation

To compare the performance of MGWO against the original GWO

Experimental Goal 1

To benchmark MGWO against other existing meta-heuristic approach

Experimental Goal 2

To verify the findings using statistical analysis

Experimental Goal 3

Novelty Assessment

Benchmarking with others

APPENDIX B

LIST OF PUBLICATIONS ON THE PHD WORK

The publications made out of the thesis are listed as follows:

Journals

1. Debashish Das, Ali Safa Sadiq, A. Noraziah (2016), An Efficient Time Series Analysis for Pharmaceutical Sector Stock Prediction by Applying Hybridization of Data Mining and Neural Network Technique; *Indian Journal of Science and Technology* (SCOPUS Indexed), Vol. 9(21), pp. 1-7, ISSN(Print):0974-6846, ISSN (Online) : 0974-5645. DOI: 10.17485/ijst/2016/v9i21/95152

2. Debashish Das, Ali Safa Sadiq, A. Noraziah, Jaime Lloret (2017), Stock Market Prediction with Big Data Through Hybridization of Data Mining and Optimized Neural Network Techniques; *Journal of Multi-Valued Logic and Soft Computing (MVLSC)*, USA, ISI Indexed Journal, IF 0.667, Vol. 29, pp. 157-181.

3. Debashish Das, Ali Safa Sadiq, Seyedali Mirjalili, A. Noraziah (2017), Hybrid Clustering-GWO-NARX neural network technique in predicting stock price; *Journal of Physics: Conference Series (JPCS)* (SCOPUS Indexed), Vol. 892, No. 012018.

Conference Papers Presented

1. Debashish Das, Ali Safa Sadiq, A. Noraziah (2015), An Efficient Time Series Analysis for Pharmaceutical Sector Stock Prediction by Applying Hybridization of Data Mining and Neural Network Technique; *Global Congress on Computing and Media Technologies (GCMT - 2015)*, Sathyabama University, Chennai and Asia Pacific University of Technology and Innovation (APU), Malaysia, 25-27 November. 2015.

2. Debashish Das, Ali Safa Sadiq, Seyedali Mirjalili, A. Noraziah (2017), Stock Prediction By Applying Hybrid Clustering-MGWO-NARX Neural Network Technique", *6th International Conference on Computer Science and Computational Mathematics (ICCSCM 2017)*, Langkawi, Malaysia, 4-5th May 2017.

APPENDIX C
STOCK DATASET

UNPROCESSED STOCK DATA FROM BURSA MALAYSIA (PARTIAL)

Stock Number	Stock Name	Date	Open Price	High Price	Low Price	Closing Price	Volume
6947	Digi Com	20050103	0.5525	0.5525	0.5347	0.5436	728000
		20050104	0.5392	0.5436	0.5392	0.5392	1981000
		20050105	0.5392	0.5436	0.5392	0.5436	2178000
		20050106	0.5436	0.5436	0.5392	0.5436	1880000
		20050107	0.5436	0.5481	0.5392	0.5392	5528000
		20050110	0.5392	0.5525	0.5392	0.5392	7693000
		20050111	0.5392	0.5481	0.5347	0.5436	4614000
		20050112	0.5436	0.5481	0.5436	0.5481	3167000
		20050113	0.5481	0.5525	0.5436	0.5436	4632000
		20050114	0.5525	0.5525	0.5436	0.5436	3392000
		20050117	0.5481	0.557	0.5436	0.5481	6807000
		20050118	0.5481	0.5481	0.5436	0.5436	7070000
		20050119	0.5525	0.5525	0.5392	0.5392	1384000
		20050120	0.5436	0.5436	0.5392	0.5392	4064000
		20050124	0.5392	0.5392	0.5347	0.5347	2525000
		20050125	0.5347	0.5392	0.5258	0.5302	4994000
		20050126	0.5302	0.5302	0.5124	0.5258	7393000
		20050127	0.5213	0.5258	0.5213	0.5213	1591000
		20050128	0.5213	0.5213	0.5169	0.5213	1458000
		20050131	0.5213	0.5347	0.508	0.508	7347000
		20050202	0.5124	0.5169	0.508	0.5124	3057000
		20050203	0.5124	0.5347	0.5124	0.5258	15602000
		20050204	0.5213	0.5302	0.5213	0.5258	4731000
		20050207	0.5302	0.5347	0.5302	0.5302	8840000
		20050208	0.5347	0.5392	0.5302	0.5347	8257000
		20050214	0.5392	0.5481	0.5347	0.5436	27897000
		20050215	0.5481	0.5525	0.5436	0.5436	17165000
		20050216	0.5436	0.5481	0.5436	0.5481	12852000
		20050217	0.5436	0.5481	0.5347	0.5347	4589000
		20050218	0.5347	0.5347	0.5347	0.5347	3629000
		20050221	0.5302	0.5347	0.5302	0.5302	4545000
		20050222	0.5302	0.5347	0.5258	0.5347	5380000
		20050223	0.5258	0.5258	0.5169	0.5213	2268000
		20050224	0.5258	0.5258	0.5169	0.5213	1489000

UNPROCESSED STOCK DATA FROM DSE, BANGLADESH (PARTIAL)

DATE	Company Code	LOW	HIGH	AVERAGE	CLOSE	TRD.VOL.	TURN OVER	COMPANY NAME
2011-01-02	18455	366.00	380.00	373.02	372.82	3467205.00	19404000.00	ACI
2011-01-03	18455	360.00	372.80	369.89	370.17	2456430.00	19404000.00	
2011-01-04	18455	365.00	373.00	367.10	366.62	3941125.00	19404000.00	
2011-01-05	18455	350.00	371.00	366.09	367.10	3046935.00	19404000.00	
2011-01-06	18455	345.00	365.10	361.24	361.82	3618185.00	19404000.00	
2011-01-09	18455	333.20	361.00	343.41	344.03	3478150.00	19404000.00	
2011-01-10	18455	303.10	335.00	316.37	316.48	1155140.00	19404000.00	
2011-01-11	18455	325.00	364.80	359.10	359.29	10228915.00	19404000.00	
2011-01-12	18455	336.60	357.10	351.20	351.76	4783995.00	19404000.00	
2011-01-13	18455	345.00	353.80	348.07	348.14	3376940.00	19404000.00	
2011-01-16	18455	331.00	365.00	342.16	345.23	6645730.00	19404000.00	
2011-01-17	18455	340.00	350.00	345.37	345.40	6139820.00	19404000.00	
2011-01-18	18455	305.00	344.00	338.08	338.92	2018931.10	19404000.00	
2011-01-19	18455	320.00	336.10	327.80	327.65	2080605.00	19404000.00	
2011-01-20	18455	310.00	323.00	316.67	319.67	383600.00	19404000.00	
2011-01-25	18455	320.10	347.50	343.49	343.38	7708890.00	19404000.00	
2011-01-26	18455	340.00	355.00	346.70	347.07	7845280.00	19404000.00	
2011-01-27	18455	335.00	350.00	343.49	343.87	5147720.00	19404000.00	
2011-01-30	18455	325.00	343.90	340.64	340.59	1631440.00	19404000.00	
2011-01-31	18455	315.00	340.00	337.95	338.32	4976750.00	19404000.00	
2011-02-01	18455	316.00	340.00	334.77	334.50	2953655.00	19404000.00	
2011-02-02	18455	320.00	333.40	329.87	329.83	2558795.00	19404000.00	
2011-02-03	18455	311.00	327.60	323.61	323.55	2482265.00	19404000.00	
2011-02-06	18455	302.00	320.00	309.91	309.95	9546580.00	19404000.00	
2011-02-07	18455	285.00	310.00	303.59	303.87	7226035.00	19404000.00	
2011-02-08	18455	298.00	315.00	304.81	303.54	7876930.00	19404000.00	
2011-02-09	18455	308.00	330.00	314.72	314.45	3179125.00	19404000.00	
2011-02-10	18455	295.00	307.90	301.11	300.98	2531230.00	19404000.00	
2011-02-13	18455	271.30	290.20	275.79	275.59	5635885.00	19404000.00	
2011-02-14	18455	248.00	270.00	248.79	248.67	3817105.00	19404000.00	
2011-02-15	18455	227.00	260.00	243.96	242.72	5162555.00	19404000.00	
2011-02-20	18455	259.00	265.00	264.48	264.67	1805040.00	19404000.00	

UNPROCESSED STOCK DATA FROM NYSE, USA (PARTIAL)

Company Name	Date	Open	High	Low	Close	Volume
IBM, NYSE	3-Jan-11	147.21	148.2	147.14	147.48	4603800
	4-Jan-11	147.56	148.22	146.64	147.64	5060100
	5-Jan-11	147.34	147.48	146.73	147.05	4657400
	6-Jan-11	147.13	148.79	146.82	148.66	5029200
	7-Jan-11	148.79	148.86	146.94	147.93	4135700
	10-Jan-11	147.58	148.06	147.23	147.64	3633400
	11-Jan-11	148.2	148.35	146.75	147.28	4163600
	12-Jan-11	147.99	149.29	147.67	149.1	4091500
	13-Jan-11	149.24	149.29	148.25	148.82	3445800
	14-Jan-11	148.89	150	148.47	150	4544200
	18-Jan-11	149.82	151.46	149.38	150.65	9176900
	19-Jan-11	153.26	156.13	152.83	155.69	12141000
	20-Jan-11	154.53	155.96	154.45	155.8	7439900
	21-Jan-11	156.4	156.78	154.96	155.5	7009000
	24-Jan-11	155.42	159.79	155.33	159.63	7285100
	25-Jan-11	159.21	164.35	159	161.44	8260800
	26-Jan-11	161.67	161.9	160.42	161.04	5353100
	27-Jan-11	161.43	162.18	160.86	161.07	4878300
	28-Jan-11	161.05	161.92	158.67	159.21	6725600
	31-Jan-11	159.18	162	158.68	162	7197200
	1-Feb-11	162.11	163.94	162	163.56	5831300
	2-Feb-11	163.4	163.6	162.61	163.3	3904000
	3-Feb-11	163.16	164.2	162.81	163.53	4683400
	4-Feb-11	163.48	164.14	163.22	164	3755200
	7-Feb-11	164.08	164.99	164.02	164.82	4928100
	8-Feb-11	164.82	166.25	164.32	166.05	5612600
	9-Feb-11	165.62	165.97	164.1	164.65	4633600
	10-Feb-11	163.9	165	163.18	164.09	5737800
	11-Feb-11	163.98	165.01	163.31	163.85	5185200
	14-Feb-11	164.18	164.38	162.85	163.22	4129800
	15-Feb-11	162.89	163.57	162.52	162.84	3768700
	16-Feb-11	163.33	163.6	162.75	163.4	3216000
	17-Feb-11	163.3	164.67	162.85	164.24	3230500
	18-Feb-11	164.46	164.84	164.1	164.84	4245000
	22-Feb-11	163.57	164.26	161.78	161.95	5209300
	23-Feb-11	161.81	162.68	160.14	160.18	5998100

UNPROCESSED STOCK DATA FROM NASDAQ, USA (PARTIAL)

Company Name	Date	Open	High	Low	Close	Volume
Microsoft, NASDAQ	3-Jan-11	28.05	28.18	27.92	27.98	53443800
	4-Jan-11	27.94	28.17	27.85	28.09	54405600
	5-Jan-11	27.9	28.01	27.77	28	58998700
	6-Jan-11	28.04	28.85	27.86	28.82	88026300
	7-Jan-11	28.64	28.74	28.25	28.6	73762000
	10-Jan-11	28.26	28.4	28.04	28.22	57573600
	11-Jan-11	28.2	28.25	28.05	28.11	50298900
	12-Jan-11	28.12	28.59	28.07	28.55	52631100
	13-Jan-11	28.33	28.39	28.01	28.19	67077600
	14-Jan-11	28.08	28.38	27.91	28.3	62688400
	18-Jan-11	28.16	28.74	28.14	28.66	53322700
	19-Jan-11	28.46	28.68	28.27	28.47	50005900
	20-Jan-11	28.5	28.55	28.13	28.35	58613600
	21-Jan-11	28.4	28.43	28.02	28.02	58080300
	24-Jan-11	28.02	28.56	27.99	28.38	52047800
	25-Jan-11	28.14	28.45	28.12	28.45	42436600
	26-Jan-11	28.51	28.99	28.5	28.78	74628800
	27-Jan-11	28.75	29.46	28.49	28.87	146938600
	28-Jan-11	28.9	28.93	27.45	27.75	141249400
	31-Jan-11	27.77	27.9	27.42	27.73	65029000
	1-Feb-11	27.8	28.06	27.61	27.99	62810700
	2-Feb-11	27.93	28.11	27.88	27.94	45824000
	3-Feb-11	27.97	27.97	27.54	27.65	60340100
	4-Feb-11	27.7	27.84	27.51	27.77	40412200
	7-Feb-11	27.8	28.34	27.79	28.2	68980900
	8-Feb-11	28.1	28.34	28.05	28.28	34904200
	9-Feb-11	28.19	28.26	27.91	27.97	52905100
	10-Feb-11	27.93	27.94	27.29	27.5	76672400
	11-Feb-11	27.76	27.81	27.07	27.25	83939700
	14-Feb-11	27.21	27.27	26.95	27.23	56766200
	15-Feb-11	27.04	27.33	26.95	26.96	44116500
	16-Feb-11	27.05	27.07	26.6	27.02	70817900
	17-Feb-11	26.97	27.37	26.91	27.21	57207300
	18-Feb-11	27.13	27.21	26.99	27.06	68667800
	22-Feb-11	26.78	27.1	26.52	26.59	60889000
	23-Feb-11	26.53	26.86	26.43	26.59	60234100

FORMATTED STOCK DATA FOR PHARMACEUTICAL SECTOR (PARTIAL) USED FOR K-MEANS CLUSTERING

COMPANY NAME	DATE	LOW	HIGH	AVERAGE	CLOSE	TRD.VOL.	TURN OVER

```
ACI,2011-01-02,366,380,373.02,372.9,979300,3467205
ACI,2011-01-03,360,372.8,369.89,370.7,706636,2456430
ACI,2011-01-04,365,373,367.1,366.6,9310750,3941125
ACI,2011-01-05,350,371,366.09,369,878300,3046935
ACI,2011-01-06,345,365.1,361.24,357.7,10110000,3618185
ACI,2011-01-09,333.2,361,343.41,340.1,10810110,3478150
ACI,2011-01-10,303.1,335,316.37,311.3,493650,1155140

BXPHARMA,2011-01-02,110,145,139.66,141.7,3020,1025331,143251680.2
BXPHARMA,2011-01-03,114,148,140.74,139.8,1718,569133,80339041.5
BXPHARMA,2011-01-04,112,153,138.1,137.3,1360,412050,56988008.2
BXPHARMA,2011-01-05,110,140,135.61,135.6,1149,383854,52136569
BXPHARMA,2011-01-06,109.1,142,135.82,133.3,1767,667923,90841559.9
BXPHARMA,2011-01-09,115,136,131.35,126.7,1948,792768,104646626.6
BXPHARMA,2011-01-10,110,130,122.46,119.6,457,186421,22919545

GLAXOSMITH,2011-01-02,1121.1,1152,1142.24,1141.6,38,2700,3084655
GLAXOSMITH,2011-01-03,1140,1168,1148.42,1143.7,53,3600,4131570
GLAXOSMITH,2011-01-04,1110,1147,1123.2,1117.7,25,2200,2464695
GLAXOSMITH,2011-01-05,1105,1129,1118.3,1111.1,51,2800,3132765
GLAXOSMITH,2011-01-06,1080,1120,1105.91,1104.8,35,2350,2599745
GLAXOSMITH,2011-01-09,970,1101,1018.07,998.4,49,3550,3591410
GLAXOSMITH,2011-01-10,932,990,954.33,953.5,6,400,381400

RECKITBEN,2011-01-02,1207.1,1290,1238.42,1238.9,10,550,681410
RECKITBEN,2011-01-03,1214,1244,1222.65,1222.4,14,750,916855
RECKITBEN,2011-01-04,1200,1215,1209.01,1209,8,400,483605
RECKITBEN,2011-01-05,1195,1257,1235.99,1235.9,20,1000,1235990
RECKITBEN,2011-01-06,1201,1230,1209.26,1208.6,12,650,785605
RECKITBEN,2011-01-09,1058.1,1150,1098.2,1098.2,10,500,549100

SQURPHARMA,2011-01-02,3521,3565,3553.74,3559.75,2349,33997,120842594
SQURPHARMA,2011-01-03,3525.25,3580,3553.84,3534.5,2609,26008,92456295
SQURPHARMA,2011-01-04,3500.25,3549,3515.99,3507.25,2215,24329,85563216.5
SQURPHARMA,2011-01-05,3464,3534.75,3482.11,3478.5,2900,20678,72001794.75
SQURPHARMA,2011-01-06,3403,3505,3440.84,3415.75,2652,20058,69006406.5
SQURPHARMA,2011-01-09,3145,3450,3281.79,3176.25,4904,73526,238928996.5
SQURPHARMA,2011-01-10,3010,3269,3063.89,3030,1309,28409,86424439.75
```

FACTORS DATA (PARTIAL)

Gold Price	Dollar Price	Bank Interest Rate	FDI	Inflation
2905.00	69.25	8.50	861736237.16	9.00
2905.00	69.25	8.50	861736237.16	9.00
2905.00	69.24	8.50	861736237.16	9.00
2905.00	69.24	8.50	861736237.16	9.00
2905.00	69.24	8.50	861736237.16	9.00
2905.00	69.27	8.50	861736237.16	9.00
2905.00	69.26	8.50	861736237.16	9.00
2905.00	69.25	8.50	861736237.16	9.00
2905.00	69.26	8.50	861736237.16	9.00
2905.00	69.25	8.50	861736237.16	9.00
2905.00	69.25	8.50	861736237.16	9.00
2905.00	69.25	8.50	861736237.16	9.00
2905.00	69.25	8.50	861736237.16	9.00
2905.00	69.25	8.50	861736237.16	9.00
2905.00	69.25	8.50	861736237.16	9.00
2905.00	69.25	8.50	861736237.16	9.00
2905.00	69.25	8.50	861736237.16	9.00
2905.00	69.25	8.50	861736237.16	9.00
2905.00	69.25	8.50	861736237.16	9.00
2995.00	69.25	8.50	861736237.16	9.00
2995.00	69.26	8.50	861736237.16	9.00
2995.00	69.31	8.50	861736237.16	9.00
2995.00	69.30	8.50	861736237.16	9.00
2995.00	69.30	8.50	861736237.16	9.00
2995.00	69.31	8.50	861736237.16	9.00
2995.00	69.31	8.50	861736237.16	9.00
2995.00	69.29	8.50	861736237.16	9.00
2995.00	69.29	8.50	861736237.16	9.00
2995.00	69.29	8.50	861736237.16	9.00
2995.00	69.30	8.50	861736237.16	9.00
3095.00	69.27	8.50	861736237.16	9.00
3095.00	69.27	8.50	861736237.16	9.00
3095.00	69.27	8.50	861736237.16	9.00
3095.00	69.27	8.50	861736237.16	9.00
3095.00	69.29	8.50	861736237.16	9.00
3095.00	69.31	8.50	861736237.16	9.00
3095.00	69.33	8.50	861736237.16	9.00
3095.00	69.38	8.50	861736237.16	9.00
3095.00	69.39	8.50	861736237.16	9.00

FORMATTED STOCK DATA FOR MICROSOFT, NASDAQ (PARTIAL) USED FOR CLASSIFICATION

```
Open High Low Close Volume Decision
37.35,37.4,37.1,37.16,30632200,1
37.2,37.22,36.6,36.91,31134800,2
36.85,36.89,36.11,36.13,43603700,2
36.33,36.49,36.21,36.41,35802800,1
36,36.14,35.58,35.76,59971700,2
35.88,35.91,35.4,35.53,36516300,2
35.9,36.15,35.75,36.04,40548800,3
35.99,36.02,34.83,34.98,45901900,1
34.73,35.88,34.63,35.78,41623300,1
35.9,36.79,35.85,36.76,44812600,3
36.69,37,36.31,36.89,38018700,3
36.83,36.83,36.15,36.38,46267500,1
36.82,36.82,36.06,36.17,31567300,2
36.26,36.32,35.75,35.93,21904300,2
36.09,36.13,35.52,36.06,43954000,1
37.45,37.55,36.53,36.81,76395500,3
36.87,36.89,35.98,36.03,44420800,2
36.12,36.39,35.75,36.27,36205500,1
35.98,36.88,35.9,36.66,52745900,1
36.79,36.88,36.23,36.86,35036300,1
36.95,37.89,36.56,37.84,93162300,3
37.74,37.99,36.43,36.48,64063100,1
36.97,37.19,36.25,36.35,54697900,2
36.29,36.47,35.8,35.82,55814400,2
35.8,36.25,35.69,36.18,35351800,1
36.32,36.59,36.01,36.56,33260500,3
36.63,36.8,36.29,36.8,26767000,3
36.88,37.26,36.86,37.17,32141400,3
37.35,37.6,37.3,37.47,27051800,3
37.33,37.86,37.33,37.61,37635500,1
37.39,37.78,37.33,37.62,31407500,1
37.63,37.78,37.41,37.42,32834000,1
37.22,37.75,37.21,37.51,29750400,1
37.57,37.87,37.4,37.75,27526100,3
37.94,38.35,37.86,37.98,38021300,3
37.69,37.98,37.54,37.69,32085100,2
37.61,37.85,37.35,37.54,30736500,2
```

NORMALIZED STOCK DATA WITH FACTORS FOR DSE (PARTIAL) USED FOR CLASSIFICATION AND PREDICTION

```
LOW HIGH AVERAGE CLOSE TRD.VOL. TURN OVER Gold Dollar Bank Interest FDI Inflation Decision
0.64,0.65,0.64,0.64,0.03,0.31,0.11,0,0.5,0.6,1
0.63,0.63,0.63,0.63,0.02,0.31,0.11,0,0.5,0.6,2
0.64,0.63,0.62,0.62,0.03,0.31,0.11,0.38,0.5,0.6,1
0.6,0.63,0.62,0.62,0.03,0.31,0.11,0.38,0.5,0.6,1
0.59,0.61,0.61,0.61,0.03,0.31,0.11,0.38,0.5,0.6,1
0.56,0.6,0.56,0.56,0.03,0.31,0.12,0.38,0.5,0.6,1
0.48,0.53,0.49,0.49,0.01,0.31,0.12,0.38,0.5,0.6,2
0.53,0.61,0.6,0.6,0.09,0.31,0.12,0.38,0.5,0.6,3
0.57,0.59,0.58,0.58,0.04,0.31,0.12,0.38,0.5,0.6,1
0.59,0.58,0.57,0.57,0.03,0.31,0.12,0.38,0.5,0.6,1
0.55,0.61,0.56,0.56,0.06,0.31,0.12,0.38,0.5,0.6,1
0.57,0.57,0.57,0.56,0.06,0.31,0.12,0.38,0.5,0.6,1
0.48,0.55,0.55,0.55,0.02,0.31,0.12,0.38,0.5,0.6,2
0.52,0.53,0.52,0.52,0.02,0.31,0.12,0.38,0.5,0.6,1
0.49,0.5,0.49,0.5,0,0.31,0.12,0.38,0.5,0.6,2
0.52,0.56,0.56,0.56,0.07,0.31,0.13,0.88,0.5,0.6,1
0.57,0.58,0.57,0.57,0.07,0.31,0.13,0.88,0.5,0.6,3
0.56,0.57,0.56,0.56,0.05,0.31,0.13,0.88,0.5,0.6,2
0.53,0.55,0.55,0.55,0.01,0.31,0.13,0.88,0.5,0.6,2
0.51,0.54,0.55,0.55,0.05,0.27,0.13,0.88,0.5,0.6,1
0.51,0.54,0.54,0.54,0.03,0.27,0.13,0.88,0.5,0.6,1
0.52,0.53,0.52,0.52,0.02,0.27,0.13,0.88,0.5,0.6,1
0.5,0.51,0.51,0.51,0.02,0.27,0.13,0.88,0.5,0.6,2
0.47,0.49,0.47,0.47,0.1,0.27,0.13,0.88,0.5,0.6,1
0.43,0.46,0.45,0.45,0.08,0.27,0.13,0.88,0.5,0.6,2
0.46,0.48,0.46,0.45,0.08,0.27,0.13,0.88,0.5,0.6,1
0.49,0.52,0.48,0.48,0.03,0.27,0.13,0.88,0.5,0.6,1
0.46,0.46,0.45,0.44,0.03,0.27,0.13,0.88,0.5,0.6,2
0.39,0.41,0.38,0.38,0.07,0.27,0.13,0.88,0.5,0.6,1
0.33,0.36,0.31,0.3,0.05,0.27,0.13,0.88,0.5,0.6,2
0.28,0.33,0.29,0.29,0.07,0.27,0.13,0.88,0.5,0.6,1
0.36,0.34,0.35,0.35,0.02,0.27,0.13,0.88,0.5,0.6,1
0.31,0.41,0.4,0.4,0.14,0.27,0.13,0.88,0.5,0.6,1
0.39,0.38,0.38,0.37,0.03,0.31,0.13,0.88,0.5,0.6,1
0.34,0.39,0.37,0.37,0.04,0.31,0.13,0.88,0.5,0.6,1
0.32,0.34,0.31,0.31,0.06,0.31,0.13,0.88,0.5,0.6,1
0.28,0.3,0.27,0.27,0.05,0.31,0.14,0.88,0.5,0.6,2
0.26,0.31,0.3,0.3,0.04,0.31,0.14,0.88,0.5,0.6,1
0.28,0.32,0.28,0.28,0.05,0.35,0.14,0.88,0.5,0.6,1
```

PREDICTED STOCK PRICE FOR DSE (PARTIAL)

Date	Actual High Price	Predicted High Price
2016-02-14	575	567.104408
2016-02-15	570	565.41389
2016-02-16	570	557.9748441
2016-02-17	569.8	562.1146667
2016-02-18	567	560.5571734
2016-02-22	560.3	564.9972015
2016-02-23	562.7	560.4792054
2016-02-24	559	560.8838751
2016-02-25	560	565.5088182
2016-02-28	569.3	561.5189172
2016-02-29	566	564.5064042
2016-03-01	562	560.901061
2016-03-02	569.7	572.1518561
2016-03-03	570	554.4631048
2016-03-06	568.2	566.5079885
2016-03-07	566.8	563.109268
2016-03-08	564	555.9084887
2016-03-09	564.7	560.4712885
2016-03-10	563	562.3853059
2016-03-13	561.9	564.7234263
2016-03-14	560.2	561.5902217
2016-03-15	560.2	553.1495136
2016-03-16	559.8	563.2032919
2016-03-20	558.9	542.9006256
2016-03-21	558.2	548.4974759
2016-03-22	557.9	566.8374045
2016-03-23	560	561.1573945
2016-03-24	551.9	557.6789548

PREDICTED STOCK PRICE FOR IBM, NYSE (PARTIAL)

Date	High Price	Predicted Price
1-Nov-16	153.91	153.5217954
2-Nov-16	153.35	154.5623053
3-Nov-16	153.74	154.163205
4-Nov-16	153.64	156.0988626
7-Nov-16	156.11	156.6354739
8-Nov-16	155.93	155.7437583
9-Nov-16	155.56	160.7995985
10-Nov-16	161.16	161.1298414
11-Nov-16	161.34	161.9060894
14-Nov-16	161.86	158.6418793
15-Nov-16	159.15	159.62198
16-Nov-16	159.55	159.93521
17-Nov-16	159.93	160.5806973
18-Nov-16	160.72	162.819883
21-Nov-16	163	162.936417
22-Nov-16	163	161.9651804
23-Nov-16	162.38	162.8870071
25-Nov-16	163.19	164.9447125
28-Nov-16	164.66	164.2288648
29-Nov-16	164.41	163.3701992
30-Nov-16	163.8	162.4664055
1-Dec-16	162.2	160.9661825
2-Dec-16	160.29	161.8363297
5-Dec-16	161.15	161.47892
6-Dec-16	160.79	165.339742
7-Dec-16	165.18	166.5287985
8-Dec-16	166	167.2625337
9-Dec-16	166.72	167.8186317
12-Dec-16	166.79	171.961337
13-Dec-16	169.95	172.1560214
14-Dec-16	169.89	169.4907888
15-Dec-16	169.85	169.9466108
16-Dec-16	169.11	167.7368666
19-Dec-16	167.26	167.3167567
20-Dec-16	168.25	167.3502629
21-Dec-16	167.94	167.4708727

PREDICTED STOCK PRICE FOR MICROSOFT, NASDAQ (PARTIAL)

Date	High Price	Predicted Price
1-Nov-16	60.02	59.85341687
2-Nov-16	59.93	59.48643484
3-Nov-16	59.64	59.8107413
4-Nov-16	59.28	60.78194607
7-Nov-16	60.52	60.79773199
8-Nov-16	60.78	60.29013033
9-Nov-16	60.59	60.17522061
10-Nov-16	60.49	58.92571291
11-Nov-16	59.12	58.89426991
14-Nov-16	59.08	59.4716785
15-Nov-16	59.49	59.62661375
16-Nov-16	59.66	60.9383013
17-Nov-16	60.95	61.08024971
18-Nov-16	61.14	60.72042479
21-Nov-16	60.97	60.95745316
22-Nov-16	61.26	60.8435135
23-Nov-16	61.1	60.27370776
25-Nov-16	60.53	60.82941335
28-Nov-16	61.02	61.17455982
29-Nov-16	61.41	61.06014329
30-Nov-16	61.18	60.15983265
1-Dec-16	60.15	59.71492073
2-Dec-16	59.47	61.15964863
5-Dec-16	60.59	61.19285246
6-Dec-16	60.46	61.31860902
7-Dec-16	61.38	61.42687908
8-Dec-16	61.58	61.91230784
9-Dec-16	61.99	62.23494712
12-Dec-16	62.3	63.37615115
13-Dec-16	63.42	63.62880811
14-Dec-16	63.45	63.14978196
15-Dec-16	63.15	63.07456504
16-Dec-16	62.95	63.81526137
19-Dec-16	63.77	64.0740938
20-Dec-16	63.8	63.76901177
21-Dec-16	63.7	64.17915854
22-Dec-16	64.1	63.49788455

PREDICTED STOCK PRICE FOR DIGI, BURSA MALAYSIA (PARTIAL)

Date	High Price	Predicted Price
20161117	5.05	5.026104957
20161118	5.02	4.99638383
20161121	4.99	4.997618784
20161122	4.99	5.027148684
20161123	5.02	5.030042152
20161124	5.01	5.008830135
20161125	5	5.012364769
20161128	5	5.018324306
20161129	5.01	5.002249116
20161130	5	4.995711232
20161201	4.99	4.985371102
20161202	4.98	4.996754893
20161205	4.98	4.961938902
20161206	4.97	4.9853446
20161207	4.99	4.979235465
20161208	4.96	4.986594763
20161209	4.98	4.982747736
20161213	4.99	4.988881997
20161214	4.99	5.000623984
20161215	4.99	5.023841968
20161216	5	5.009126594
20161219	5	4.996199664
20161220	5	5.020539845
20161221	4.99	5.008051237
20161222	4.99	4.971299992
20161223	4.97	5.022726837
20161227	5	5.013838595
20161228	5	5.037868758
20161229	5	5.037548222
20161230	5	5.022618618

STOCK INVESTMENT DECISION DATA FOR IBM, NYSE (PARTIAL)

DATE	OPEN	CLOSE	HIGH	LOW	DECISION
12-APR-16	149.61	149.63	150.11	148.58	Hold
11-APR-16	150.26	149.25	151.95	149.22	Hold
8-APR-16	149.06	149.35	149.99	148.12	Hold
7-APR-16	149.07	148.25	149.6	148	Hold
6-APR-16	149.9	150.02	150.15	147.81	Hold
5-APR-16	151.43	150	151.6	149.65	Hold
4-APR-16	152.34	152.07	153.52	151.91	Sell
1-APR-16	150.51	152.52	152.96	150.25	Hold
31-MAR-16	149.79	151.45	153.1	149.36	Hold
30-MAR-16	149.95	148.41	150.41	148.32	Hold
29-MAR-16	147.59	149.33	149.76	147.5	Hold
28-MAR-16	147.75	148.4	148.65	147.23	Hold
24-MAR-16	144.98	147.95	148.22	144.49	Buy
23-MAR-16	148	145.4	148.03	145.13	Hold
22-MAR-16	148.06	148.1	149.28	147.84	Sell
21-MAR-16	147.3	148.63	148.71	146.72	Hold
18-MAR-16	147.4	147.09	147.51	145.51	Hold
17-MAR-16	144.78	147.04	147.32	144.45	Buy
16-MAR-16	142.62	144.79	144.88	142.11	Buy
15-MAR-16	141.74	142.96	143.33	141.54	Buy
14-MAR-16	142.01	142.78	143.19	141.04	Hold
11-MAR-16	141.73	142.36	142.92	140.51	Buy
10-MAR-16	141.24	140.19	141.47	138.09	Buy
9-MAR-16	139.31	140.41	142.17	139.23	Hold
8-MAR-16	139.71	139.07	140.35	137.42	Hold
7-MAR-16	137.28	140.15	140.51	136.87	Hold
4-MAR-16	137.54	137.8	139.42	137.02	Hold
3-MAR-16	137.22	137.8	137.96	136.07	Hold
2-MAR-16	133.7	136.3	137.44	133.22	Buy
1-MAR-16	132.24	134.37	134.64	132.03	Buy

STOCK INVESTMENT DECISION DATA FOR MICROSOFT, NASDAQ (PARTIAL)

Date	Open	High	Low	Close	Decision
13-Apr-16	55.12	55.35	55.44	54.89	Hold
12-Apr-16	54.37	54.65	54.78	53.76	Buy
11-Apr-16	54.49	54.31	55.15	54.3	Hold
8-Apr-16	54.67	54.42	55.28	54.32	Sell
7-Apr-16	54.87	54.46	54.91	54.23	Hold
6-Apr-16	54.36	55.12	55.2	54.21	Hold
5-Apr-16	55.19	54.56	55.3	54.46	Hold
4-Apr-16	55.43	55.43	55.66	55	Sell
1-Apr-16	55.05	55.57	55.61	54.57	Hold
31-Mar-16	54.95	55.23	55.59	54.86	Hold
30-Mar-16	54.93	55.05	55.64	54.9	Hold
29-Mar-16	53.66	54.71	54.86	53.45	Buy
28-Mar-16	54.21	53.54	54.29	53.33	Hold
24-Mar-16	53.84	54.21	54.33	53.73	Hold
23-Mar-16	54.11	53.97	54.24	53.74	Hold
22-Mar-16	53.61	54.07	54.25	53.46	Hold
21-Mar-16	53.25	53.86	53.93	52.93	Buy
18-Mar-16	54.92	53.49	54.97	53.45	Hold
17-Mar-16	54.21	54.66	55	54	Hold
16-Mar-16	53.45	54.35	54.6	53.4	Buy
15-Mar-16	52.75	53.59	53.59	52.74	Buy
14-Mar-16	52.71	53.17	53.59	52.63	Hold
11-Mar-16	53	53.07	53.07	52.38	Hold
10-Mar-16	52.93	52.05	52.94	51.16	Buy
9-Mar-16	51.89	52.84	52.85	51.86	Hold
8-Mar-16	50.8	51.65	52.13	50.6	Buy
7-Mar-16	51.56	51.03	51.8	50.58	Hold
4-Mar-16	52.4	52.03	52.45	51.71	Sell
3-Mar-16	52.97	52.35	52.97	51.78	Sell
2-Mar-16	52.41	52.95	52.96	52.16	Hold
1-Mar-16	50.97	52.58	52.59	50.92	Buy
29-Feb-16	51.35	50.88	51.65	50.66	Hold
26-Feb-16	52.6	51.3	52.68	51.1	Sell
25-Feb-16	51.73	52.1	52.1	50.61	Hold
24-Feb-16	50.69	51.36	51.5	50.2	Buy
23-Feb-16	52.34	51.18	52.37	50.98	Hold
22-Feb-16	52.28	52.65	53	52.28	Hold
19-Feb-16	51.97	51.82	52.28	51.53	Buy

STOCK INVESTMENT DECISION DATA FOR DIGI, BURSA MALAYSIA (PARTIAL)

Date	Open	High	Low	Close	Decision
20161111	4.97	5.02	4.91	4.99	Buy
20161114	4.99	4.99	4.88	4.92	Hold
20161115	4.92	4.99	4.92	4.98	Hold
20161116	5	5.02	4.98	4.99	Sell
20161117	5	5.01	4.93	5	Hold
20161118	4.99	5	4.95	4.99	Hold
20161121	5	5	4.97	4.99	Hold
20161122	5	5.01	4.94	4.99	Hold
20161123	5	5	4.96	4.99	Hold
20161124	4.99	4.99	4.95	4.97	Buy
20161125	4.95	4.98	4.95	4.97	Hold
20161128	4.97	4.98	4.96	4.97	Hold
20161129	4.93	4.97	4.9	4.95	Buy
20161130	4.97	4.99	4.87	4.87	Hold
20161201	4.9	4.96	4.88	4.95	Hold
20161202	4.95	4.98	4.94	4.95	Hold
20161205	4.91	4.99	4.91	4.96	Hold
20161206	4.97	4.99	4.95	4.98	Hold
20161207	4.97	4.99	4.96	4.98	Hold
20161208	4.99	5	4.97	4.99	Sell
20161209	4.95	5	4.95	4.99	Hold
20161213	5	5	4.96	4.98	Hold
20161214	4.96	4.99	4.96	4.98	Hold
20161215	4.94	4.99	4.93	4.97	Hold
20161216	4.94	4.97	4.94	4.96	Hold
20161219	4.94	5	4.94	4.98	Hold
20161220	5	5	4.97	4.99	Hold
20161221	5	5	4.97	4.98	Hold
20161222	4.99	5	4.95	4.97	Hold
20161223	4.97	4.99	4.95	4.95	Hold
20161227	4.93	4.97	4.93	4.93	Buy
20161228	4.93	4.99	4.93	4.94	Hold
20161229	4.94	4.98	4.92	4.96	Hold
20161230	4.95	4.97	4.83	4.83	Hold

STOCK PRICE PREDICTION VALIDATION DATA FOR ACI, DSE (PARTIAL)

Date	High Price	Predicted High Price	Forecasting Error (%)	MAPE	MAD	RMSE
2016-02-14	575	567.104408	1.373146443	3.337062276	7.895592045	62.34037374
2016-02-15	570	565.41389	0.804580696		4.586109968	21.03240464
2016-02-16	570	557.9748441	2.109676467		12.02515586	144.6043735
2016-02-17	569.8	562.1146667	1.348777343		7.685333298	59.0643479
2016-02-18	567	560.5571734	1.13630099		6.442826612	41.51001475
2016-02-22	560.3	564.9972015	0.838336878		4.697201526	22.06370217
2016-02-23	562.7	560.4792054	0.394667599		2.220794579	4.931928563
2016-02-24	559	560.8838751	0.337008075		1.883875138	3.548985536
2016-02-25	560	565.5088182	0.983717529		5.508818162	30.34707755
2016-02-28	569.3	561.5189172	1.36678074		7.781082754	60.54524883
2016-02-29	566	564.5064042	0.263886176		1.493595757	2.230828286
2016-03-01	562	560.901061	0.195540742		1.098938971	1.207666862
2016-03-02	569.7	572.1518561	0.430376713		2.451856133	6.011598498
2016-03-03	570	554.4631048	2.725771083		15.53689517	241.3951116
2016-03-06	568.2	566.5079885	0.297784488		1.692011462	2.862902787
2016-03-07	566.8	563.109268	0.651152437		3.690732013	13.62150279
2016-03-08	564	555.9084887	1.434665126		8.091511309	65.47255526
2016-03-09	564.7	560.4712885	0.748842122		4.228711463	17.88200064
2016-03-10	563	562.3853059	0.109181907		0.614694136	0.37784888
2016-03-13	561.9	564.7234263	0.502478433		2.823426317	7.971736168
2016-03-14	560.2	561.5902217	0.24816525		1.39022173	1.932716459
2016-03-15	560.2	553.1495136	1.258565939		7.050486388	49.7093583
2016-03-16	559.8	563.2032919	0.607947817		3.403291879	11.58239562
2016-03-20	558.9	542.9006256	2.862654209		15.99937438	255.9799804
2016-03-21	558.2	548.4974759	1.738180598		9.702524096	94.13897384
2016-03-22	557.9	566.8374045	1.601972489		8.937404516	79.87719948
2016-03-23	560	561.1573945	0.206677593		1.157394519	1.339562074
2016-03-24	551.9	557.6789548	1.047101787		5.778954763	33.39631815
2016-03-27	555	552.4391825	0.461408566		2.560817539	6.557786466
2016-03-28	542.6	536.8608295	1.057716636		5.739170466	32.93807764
2016-03-29	535.2	548.5876587	2.501430995		13.38765869	179.2294051
2016-03-30	542	551.7744488	1.80340384		9.774448813	95.5398496
2016-03-31	549	547.443365	0.283540078		1.55663503	2.423112615
2016-04-03	546.5	550.4843579	0.729068237		3.984357917	15.87510801
2016-04-04	558.7	568.4842289	1.75124913		9.784228891	95.73113499
2016-04-05	560.8	574.8960249	2.513556505		14.09602488	198.6979174
2016-04-06	556	573.449396	3.138380572		17.44939598	304.48142
2016-04-07	574	581.9066801	1.377470396		7.906680075	62.51558982

STOCK PRICE PREDICTION VALIDATION DATA FOR BURSA MALAYSIA (PARTIAL)

Date	High Price	Predicted High Price	Forecasting Error (%)	MAPE	MAD	RMSE
20161117	5.05	5.026104957	0.473169174	0.796920921	0.023895	0.00057097
20161118	5.02	4.99638383	0.470441628		0.023616	0.00055772
20161121	4.99	4.997618784	0.152681041		0.007619	5.8046E-05
20161122	4.99	5.027148684	0.744462599		0.037149	0.00138002
20161123	5.02	5.030042152	0.200042871		0.010042	0.00010084
20161124	5.01	5.008830135	0.023350599		0.00117	1.3686E-06
20161125	5	5.012364769	0.247295374		0.012365	0.00015289
20161128	5	5.018324306	0.366486116		0.018324	0.00033578
20161129	5.01	5.002249116	0.154708272		0.007751	6.0076E-05
20161130	5	4.995711232	0.085775369		0.004289	1.8394E-05
20161201	4.99	4.985371102	0.092763485		0.004629	2.1427E-05
20161202	4.98	4.996754893	0.336443628		0.016755	0.00028073
20161205	4.98	4.961938902	0.362672655		0.018061	0.0003262
20161206	4.97	4.9853446	0.308744464		0.015345	0.00023546
20161207	4.99	4.979235465	0.215722151		0.010765	0.00011588
20161208	4.96	4.986594763	0.536184732		0.026595	0.00070728
20161209	4.98	4.982747736	0.055175424		0.002748	7.5501E-06
20161213	4.99	4.988881997	0.022404862		0.001118	1.2499E-06
20161214	4.99	5.000623984	0.212905486		0.010624	0.00011287
20161215	4.99	5.023841968	0.678195753		0.033842	0.00114528
20161216	5	5.009126594	0.182531888		0.009127	8.3295E-05
20161219	5	4.996199664	0.076006718		0.0038	1.4443E-05
20161220	5	5.020539845	0.410796892		0.02054	0.00042189
20161221	4.99	5.008051237	0.361748227		0.018051	0.00032585
20161222	4.99	4.971299992	0.374749661		0.0187	0.00034969
20161223	4.97	5.022726837	1.060902145		0.052727	0.00278012
20161227	5	5.013838595	0.27677191		0.013839	0.00019151
20161228	5	5.037868758	0.757375157		0.037869	0.00143404
20161229	5	5.037548222	0.750964442		0.037548	0.00140987
20161230	5	5.022618618	0.452372362		0.022619	0.0005116

STOCK PRICE PREDICTION VALIDATION DATA FOR IBM, NYSE (PARTIAL)

Date	High Price	Predicted High Price	Forecasting Error (%)	MAPE	MAD	RMSE
1-Mar-16	134.64	137.2290225	1.922922254	0.87053416	2.589023	6.703038
2-Mar-16	137.44	137.542491	0.074571433		0.102491	0.010504
3-Mar-16	137.96	139.3868376	1.034240041		1.426838	2.035865
4-Mar-16	139.42	140.5829261	0.834117112		1.162926	1.352397
7-Mar-16	140.51	140.4626231	0.03371778		0.047377	0.002245
8-Mar-16	140.35	142.5220042	1.547562676		2.172004	4.717602
9-Mar-16	142.17	141.6724808	0.349946655		0.497519	0.247525
10-Mar-16	141.47	143.3131481	1.302854363		1.843148	3.397195
11-Mar-16	142.92	143.8282358	0.63548546		0.908236	0.824892
14-Mar-16	143.19	144.5345353	0.938986903		1.344535	1.807775
15-Mar-16	143.33	146.4850069	2.201218807		3.155007	9.954069
16-Mar-16	144.88	147.6506893	1.912402915		2.770689	7.676719
17-Mar-16	147.32	148.0548367	0.498803099		0.734837	0.539985
18-Mar-16	147.51	149.63284	1.439116007		2.12284	4.50645
21-Mar-16	148.71	150.5790477	1.25684061		1.869048	3.493339
22-Mar-16	149.28	148.1899917	0.730177018		1.090008	1.188118
23-Mar-16	148.03	148.4939516	0.313417289		0.463952	0.215251
24-Mar-16	148.22	148.7364049	0.348404361		0.516405	0.266674
28-Mar-16	148.65	150.0443661	0.938019574		1.394366	1.944257
29-Mar-16	149.76	150.9063695	0.765471063		1.146369	1.314163
30-Mar-16	150.41	153.5053872	2.057966377		3.095387	9.581422
31-Mar-16	153.1	153.6858524	0.382659959		0.585852	0.343223
1-Apr-16	152.96	153.960154	0.653866371		1.000154	1.000308
4-Apr-16	153.52	152.3130722	0.786169768		1.206928	1.456675
5-Apr-16	151.6	150.8597119	0.488316721		0.740288	0.548027
6-Apr-16	150.15	149.7800794	0.246367367		0.369921	0.136841
7-Apr-16	149.6	150.3084755	0.473579847		0.708475	0.501937
8-Apr-16	149.99	152.7064381	1.811079457		2.716438	7.379036
11-Apr-16	151.95	150.60866	0.882750927		1.34134	1.799193
12-Apr-16	150.11	151.7115356	1.066908003		1.601536	2.564916

STOCK PRICE PREDICTION VALIDATION DATA FOR MICROSOFT, NASDAQ (PARTIAL)

Date	High Price	Predicted High Price	Forecasting Error (%)	MAPE	MAD	RMSE
19-Feb-16	52.28	53.10262496	1.573498388	1.111934817	0.822625	0.676712
22-Feb-16	53	52.2440023	1.426410761		0.755998	0.571533
23-Feb-16	52.37	51.6051487	1.460476032		0.764851	0.584998
24-Feb-16	51.5	52.20588127	1.370643235		0.705881	0.498268
25-Feb-16	52.1	52.69330547	1.138782083		0.593305	0.352011
26-Feb-16	52.68	51.6567134	1.942457486		1.023287	1.047115
29-Feb-16	51.65	52.65354016	1.942962557		1.00354	1.007093
1-Mar-16	52.59	53.02727629	0.831481826		0.437276	0.191211
2-Mar-16	52.96	53.00669357	0.088167614		0.046694	0.00218
3-Mar-16	52.97	52.48310571	0.919188768		0.486894	0.237066
4-Mar-16	52.45	51.61296855	1.595865482		0.837031	0.700622
7-Mar-16	51.8	52.21337489	0.798021021		0.413375	0.170879
8-Mar-16	52.13	52.88401572	1.446414199		0.754016	0.56854
9-Mar-16	52.85	52.99067799	0.266183523		0.140678	0.01979
10-Mar-16	52.94	53.04336139	0.195242516		0.103361	0.010684
11-Mar-16	53.07	53.78959417	1.355934001		0.719594	0.517816
14-Mar-16	53.59	53.67298201	0.154846078		0.082982	0.006886
15-Mar-16	53.59	54.36045455	1.437683427		0.770455	0.5936
16-Mar-16	54.6	55.08867958	0.895017538		0.48868	0.238808
17-Mar-16	55	54.87876033	0.220435772		0.12124	0.014699
18-Mar-16	54.97	53.86385751	2.012265754		1.106142	1.223551
21-Mar-16	53.93	54.23100819	0.558146097		0.301008	0.090606
22-Mar-16	54.25	54.33072316	0.148798444		0.080723	0.006516
23-Mar-16	54.24	54.20015017	0.073469459		0.03985	0.001588
24-Mar-16	54.33	54.32746599	0.004664105		0.002534	6.42E-06
28-Mar-16	54.29	54.97089401	1.254179428		0.680894	0.463617
29-Mar-16	54.86	55.45412409	1.082982297		0.594124	0.352983
30-Mar-16	55.64	55.49076035	0.268223678		0.14924	0.022272
31-Mar-16	55.59	55.52272915	0.121012506		0.067271	0.004525
1-Apr-16	55.61	55.69015391	0.144135787		0.080154	0.006425

STOCK PRICE WITH FACTORS DATA TO MEASURE THE AFFECT FOR DSE (PARTIAL)

LOW	HIGH	AVERAGE	CLOSE	Gold Price	Dollar Price	Bank Interest Rate	FDI	Inflation
499.00	510.90	507.03	507.07	2905.00	69.25	8.50	861736237.16	9.00
490.00	510.00	504.89	504.76	2905.00	69.25	8.50	861736237.16	9.00
478.00	504.00	500.83	500.71	2905.00	69.24	8.50	861736237.16	9.00
490.00	502.80	499.15	499.16	2905.00	69.24	8.50	861736237.16	9.00
490.00	508.00	502.03	502.23	2905.00	69.24	8.50	861736237.16	9.00
501.00	505.00	502.69	502.71	2905.00	69.27	8.50	861736237.16	9.00
496.10	505.00	499.16	499.17	2905.00	69.26	8.50	861736237.16	9.00
490.50	499.80	494.08	493.96	2905.00	69.25	8.50	861736237.16	9.00
480.00	500.00	491.54	491.51	2905.00	69.26	8.50	861736237.16	9.00
470.00	495.00	491.04	491.13	2905.00	69.26	8.50	861736237.16	9.00
480.00	500.00	497.24	497.19	2905.00	69.25	8.50	861736237.16	9.00
491.10	495.50	492.41	492.43	2905.00	69.25	8.50	861736237.16	9.00
490.00	495.00	491.55	491.57	2905.00	69.25	8.50	861736237.16	9.00
443.20	498.00	494.66	495.08	2905.00	69.25	8.50	861736237.16	9.00
491.00	501.90	497.04	497.14	2905.00	69.25	8.50	861736237.16	9.00
445.60	495.00	489.34	489.68	2905.00	69.25	8.50	861736237.16	9.00
480.00	489.00	486.30	486.24	2905.00	69.25	8.50	861736237.16	9.00
460.00	489.00	479.42	479.44	2905.00	69.25	8.50	861736237.16	9.00
451.00	475.10	457.10	457.13	2905.00	69.25	8.50	861736237.16	9.00
440.00	459.00	454.73	454.65	2905.00	69.25	8.50	861736237.16	9.00
430.00	452.00	443.94	444.02	2905.00	69.25	8.50	861736237.16	9.00
425.00	445.00	435.57	435.57	2905.00	69.25	8.50	861736237.16	9.00
400.00	456.80	440.84	441.22	2905.00	69.25	8.50	861736237.16	9.00
438.00	445.00	440.67	440.57	2995.00	69.25	8.50	861736237.16	9.00
438.40	442.00	440.11	440.09	2995.00	69.26	8.50	861736237.16	9.00
421.50	433.00	426.42	426.33	2995.00	69.31	8.50	861736237.16	9.00
400.00	440.00	437.12	437.19	2995.00	69.30	8.50	861736237.16	9.00
425.00	440.00	435.05	434.56	2995.00	69.30	8.50	861736237.16	9.00
419.90	435.00	430.96	430.99	2995.00	69.31	8.50	861736237.16	9.00
410.00	431.90	429.24	429.31	2995.00	69.31	8.50	861736237.16	9.00
400.00	428.50	418.99	419.15	2995.00	69.29	8.50	861736237.16	9.00