Gene expression programming for Efficient Time-series Financial Forecasting

Manal Alghieth, BA, MSc

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at De Montfort University

August 2016

Abstract

Stock market prediction is of immense interest to trading companies and buyers due to high profit margins. The majority of successful buying or selling activities occur close to stock price turning trends. This makes the prediction of stock indices and analysis a crucial factor in the determination that whether the stocks will increase or decrease the next day. Additionally, precise prediction of the measure of increase or decrease of stock prices also plays an important role in buying/selling activities. This research presents two core aspects of stock-market prediction. Firstly, it presents a Network-based Fuzzy Inference System (ANFIS) methodology to integrate the capabilities of neural networks with that of fuzzy logic. A specialised extension to this technique is known as the genetic programming (GP) and gene expression programming (GEP) to explore and investigate the outcome of the GEP criteria on the stock market price prediction.

The research presented in this thesis aims at the modelling and prediction of short-to-medium term stock value fluctuations in the market via genetically tuned stock market parameters. The technique uses hierarchically defined GP and gene-expression-programming (GEP) techniques to tune algebraic functions representing the fittest equation for stock market activities. The technology achieves novelty by proposing a fractional adaptive mutation rate Elitism (GEP-FAMR) technique to initiate a balance between varied mutation rates between varied-fitness chromosomes thereby improving prediction accuracy and fitness improvement rate. The methodology is evaluated against five stock market companies with each having its own trading circumstances during the past 20+ years. The proposed GEP/GP methodologies were evaluated based on variable window/population sizes, selection methods, and Elitism, Rank and Roulette selection methods. The Elitism-based approach showed promising results with a low error-rate in the resultant pattern matching with an overall accuracy of 95.96% for short-term 5-day and 95.35% for medium-term 56-day trading periods. The contribution of this research to theory is that it presented a novel evolutionary methodology with modified selection operators for the prediction of stock exchange data via Gene expression programming. The methodology dynamically adapts the mutation rate of different fitness groups in each generation to ensure a diversification

balance between high and low fitness solutions. The GEP-FAMR approach was preferred to Neural and Fuzzy approaches because it can address well-reported problems of over-fitting, algorithmic black-boxing, and data-snooping issues via GP and GEP algorithms

Acknowledgements

Undertaking this PhD has been a truly precious experience for me and it would not have been possible to do without the support that I have received.

I firstly thank God for giving me the opportunity to conduct this research and helping me throughout the journey of my PhD.

I would like to express my heartfelt appreciation to my late husband, may God bless his soul, for his early encouragement and for the seed he planted inside me to start this PhD , he has been my courage to complete this journey.

I extend my thanks to both of my supervisors, Yingjie Yang and Francisco Chiclana, for their constant guidance and support.

My deep gratefulness goes out to my family for the endless support and for always believing in me.

My sincere thanks to my friends who have always been true friends, sharing, inspiring and encouraging.

I gratefully acknowledge the fund received from Saudi cultural bureau and Qassim University

Contents

Publication Relating To PhD Thesis:

Conference:

[1] Development of 2D Curve-Fitting Genetic/Gene-Expression Programming Technique for Efficient Time-series Financial Forecasting Alghieth, Manal ;Yang, Yingjie; Chiclana, Francisco (IEEE Xplore, 2015) International Symposium on INnovations in Intelligent SysTems and Applications. Madrid, Spain.

[2] Development of a Genetic Programming-based GA Methodology for the Prediction of Short-to-Medium-term Stock Markets. Alghieth, Manal; Yang, Yingjie; Chiclana, Francisco (IEEE, 2016) world Congress on Computational Intelligence. Vancouver, Canada.

List of Acronyms

ACO - Ant Colony Optimisation

ADLPSO - A Dynamic Particle Swarm Optimization Algorithm

AI – Artificial Intelligence

ANFIS – Adaptive Neuro Fuzzy Inference System

ANN - Artificial Neural Networks

CART - Classification and Regression Tree

CBR – Case Based Reasoning

CPSFC - Chaotic Particle Swarm Fuzzy Clustering

DEPM - Differential Equation Prediction Method

DFDs – Data Flow Diagrams

DSDM - Dynamic Systems Development Method

ERDs - Entity Relationship Diagrams

FAMR-Fractional Adaptive Mutation Rate (the proposed)

FIS - Fuzzy Inference System

GA - Genetic Algorithms

GEP - Gene Expression Programming

HMM - Hidden Markov Models

JAD - Joint Application Development

LDA/MDA - Linear and Multi-Variate Discriminant Analysis

LLWN - Local Linear Wavelet-based Networks

LVQ - Learning Vector Quantisation

MAPE - Mean Absolute Percentage Error

MRL - Morphological-Rank-Linear

OOAD - Object Oriented design and Analysis

OPELM - Optimally Pruned Extreme Learning Machine

OPKNN - Optimally K-Nearest-Neighbours

PSO - Particle Swarm Optimisation

RAD - Rapid Application Development

RBF - Radial Basis Function

RMSE - Root-Mean Square Error

RPNN – Recurrent Artificial Neural Networks

RST - Rough-Set Theory

SDLC - Systems Development Life Cycle

SNR - Signal-to-Noise Ratios

SVM - Support Vector Machines

SWPM - Sliding Window Prediction Method

TSK - Takagi-Sugeno Kang

VPRS - Variable Precision Rough Set

1. Chapter1: Introduction

1.1 Background Information

Fox (2011) notes that financial markets existed as early as the 16th century, when various provinces in Europe collaborated in trading activities, which eventually to the formation of common exchange Dutch bank. Throughout the centuries, these activities would be replicated on a worldwide scale and involve numerous other parties and endeavours, which, together, eventually founded the global financial market (Fox, 2011). Central to this set up is prediction, in which various models and systems are used to speculate the performance of different entities in the market. For instance, recently, there have been numerous debates regarding the financial impact Brexit will have on various market aspects including the sterling pound's value, which since the exit vote has considerably reduced.

Accordingly, as shown in the Brexit example, where the financial forecasts were central to the debates to leave or remain in the EU (European-Union), prediction helps in major decision-making processes. Furthermore, a major area impacted by forecasting is investment, where entities, for instance, stocks, that are projected as having a favourable future performance attract more investors and those associated with considerable falls or high investment risk, do not draw buyers. Therefore, depending on the prediction results, investors, who considerably influence the performance of various economies, are drawn to empower involved states, and hence drive the global development agenda. Therefore, financial markets forecasting is crucial and provides critical information that considerably influences the decisions having a global impact (Satchell & Knight, 2011; Daníelsson, 2011).

According to Kumar and Ravi (2007), financial trend research is broadly classified into:

- Statistical techniques
- Artificial Neural Networks (ANN)
- Case-based reasoning (CBR)
- Decision trees

- Operational research (OR)
- Evolutionary computing research
- Rough-set-based techniques
- Fuzzy inference systems (FIS), support vector machines (SVM) and isotonic separation-based approaches

The study of Kumar and Ravi (2007) shows important trends in current financial stock modelling approaches. For instance, the review indicated that statistical techniques are no longer used. Moreover, integral Artificial Intelligence (AI) approaches such as the ANN were the most frequently employed technique for stock market prediction and in terms of usage, is followed by rough set (RS) theory, CBR, OR, FIS and SVM-based methodology, in this order. According to Kumar and Ravi (2007), a consensus over integrated (hybrid) methodology has been realised because majority of lately conducted researches report outstanding accuracies when comparing stand-alone techniques.

Unlike contemporary methods that use exact equations and mathematical formulations, fuzzy logic presents an alternative way to realising real-world systems that anticipate or model underlying prediction problems by using vagueness that is based on expert knowledge and experience. The methodology was given by Zadeh *et al.* (1996) to deal and present approximate reasoning instead of fixed or exact logic. The ability of fuzzy inference systems to present values between ranges instead of fixed variables makes it an ideal paradigm for numerous areas including those involving control systems and (AI). Nonetheless, this methodology predominantly relies on human expertise to tune and apply its rules to specific real-world scenario. For highly complex systems, the application is a tedious task that requires hundreds, and often, thousands of linguistic rules to be written when modelling a system. FIS has been combined with various models to predict financial systems. For instance, Kablan (2009) developed an ANFIS (Adaptive-Neuro-Fuzzy-Inference-System) that combined market patterns and fuzzy reasoning to make trading and financial forecasts. Other approaches that help allow FIS to predict financial systems include employment of neural networks, observation models and trading frequency systems. The FIS has the potential of enabling financial systems prediction; however, Kingdon (2012) notes that in most cases, this depends on the developer's expertise and understanding of the particular market of application. Furthermore, developed systems require consistent updates because of rapidly changing financial market factors.

In addition, conventional time series analysis systems are based on centralised instructions, constraints and rules. One such example can be that of auto-regressive moving average (ARMA) models that have been widely applied to the field of financial analysis (Balakrishnan 2010, p. 321). Contrary to ARMA models and fuzzy logic, ANNs do not operate over a set of predefined rules or mathematical equations. When subjected to training data, the neural network learns from the irregularities of data and hence formulates its own rule base. Therefore, the data is not explicitly elaborated in the form of mathematical formulation, which makes ANN unique in generalisation over a wide range of real-world input-output data pair cases.

Based on the characteristics stated above, ANNs bear several advantages over calibrated modelling techniques. Most notably is ANNs' ability to generalise over a wide range of test cases of the data the network is training for. Since the network has the ability to generalise, ANNs can, to some extent, even predict missing, sparse or low-quality data as well as the quality well suited for highly non-linear and uncertain cases of financial forecasting. Moreover, ANNs are well-known for their ability to handle input variables in a parallel fashion, which enables handling of large datasets in a swift manner.

1.2 Problem Statement

Since the development of financial markets, various systems have been used for forecasting and over the years, prediction accuracy has been considerably improved. However, as noted by Satchell & Knight (2011), modern financial systems are generally highly complex and prone to several system disruptions ranging from unknown market uncertainties to unanticipated issues of a social, economic as well as political nature. As a result, these factors have led to considerable inefficiencies in financial forecasting, which according to Liu *et al.* (2006, p. 314), have been primarily caused by poor handling of information and its consequential uncertainties.

Prediction of non-probabilistic systems presents a significant challenge to theoretical modelling methods. Financial forecasts, and particularly, uncertain trends in stock markets, business performance indices and economical predictions are largely regarded as "non-linear" or "near-zero-probability" systems. The main challenge these aspects present to ongoing forecasting research is prediction complexities (Liu *et al.* 2006, p. 13). Stock market predictions have been of high interest to traders because when correctly forecasted, one could potentially make high returns over a short period. In addition, stock market predictions are widely employed by companies and individuals to engage in profitable stock trading in the financial market. However, the rise and fall of stock prices is non-linear and characterised by a diverse range of factors, which makes a direct or expert-driven rule-based approach to stock trading an extremely tedious and unreliable technique.

In addition, the number of factors affecting stock prices is so massive and this makes it impossible even for an experienced financial specialist to use a combination of these factors for prediction. Furthermore, many of these factors are intangible; that is, even though company profits can be predicted with a reasonable level of accuracy, it is not possible to predict a plethora of qualitative factors such as staff skills, trading experience, competitive edges, and corporate reputation. Most importantly, in real stock markets, humans are actively involved and often make completely nondeterministic and irrational decisions that directly affect the future stock prices of a firm. In fact, the prediction impossibility of stock process is often compared with chess, which, as noted by Norton (2013), has $1*10^{120}$ possible moves. Furthermore, the prediction difficulty is increased by the fact that historical models, which are a conventional forecasting business tool, are not as effective when applied to financial markets. The effectiveness of these models is considerably hampered by the multitudinous market factors that, at individual levels, have a significant influence on stock prices (Kingdon, 2012). Nonetheless, Kingdon (2012) points out that to deal with the prediction impossibilities, investors can employ measures such as investment diversification, working with professionals or established companies, and focusing on long-term results, as opposed to short-term ones. A notable thing with these measures is that rather than addressing prediction complexities, they endeavour to reduce risk associated with forecasting impossibilities and hence fail to provide a lasting prediction solution. Therefore, stock price prediction in the financial market remains a major challenge for traders and investors.

Recently, major studies in the domain of artificial intelligence and pattern recognition for the prediction of time-series-based financial activity patterns in the stock market have been done. The majority of these efforts have mainly focused on ANN, hidden Markov models (HMM), and fuzzy inference/logic (FL) techniques. Moreover, areas of statistical analysis including temporal trend analysis via support vector machines, curve fitting via Kalman filters and graphical splines have already been explored and offer promising outcomes (Bisoi *et al.*, 2011). However, each of these models and approaches bears numerous limitations. For example, HMM is associated with expensive algorithms, slow training processes caused by numerous seed sequences and complex model choosing processes, which make it unsuitable for financial systems prediction. Additionally, trend analysis considerably depends on historical models, which for financial markets, are not a strong basis for trading decision-making processes. Therefore, while a wide range of options could be employed for financial systems predictions involved drawbacks necessitate the development of a more suitable methodology. Nonetheless, the domain of biologically and genetically driven techniques that are derived AI offers models capable of learning from historic feature information indices to predict, with reasonable level of accuracy, the future values of real-world time-series events. For instance, ANN is well-used in the prediction of stock market indices. However, regardless of their optimistic usage, commonly implemented feed-forwarded and principle-component ANNs do not deliver outstanding prediction accuracies (Blandis, 2002). Furthermore, AI algorithms, which are used in ANNs during financial market predicition, bear the negative implication of noisy data and over-training that affects the measure of generalisation. Besides, it is considerably difficult to select the most optimal set of input variables for the neural training process. Moreover, as the training requires a large dataset to enable the network to generalise, the training speed is generally very slow.

Based on these limitations, this research presents a novel GEP methodology. The methodology utilises a dynamic sliding-window-architecture over well-known stock datasets to predict future stocks based on two conditions: 1) medium-term stock training conditions spanning over durations of two months and above, and (2) short-term training-based prediction based on a weeklong sliding windows. The technique proposed in this work extends on existing GEP-based time-series prediction techniques by improving on the limitations of constant mutation rates via a proposed GEP-FAMR mechanism. During consecutive runs, evolutionary techniques bear mutation rates that present equal likelihood, for solutions with varied fitness rates, to undergo a gene-alteration process. Often, this leads to loss of fitter solutions; however, when applied to low-to-average fitness solutions, this mutation operator facilitates diversity and hence paves way for better solutions to survive.

To date, the majority of adaptive genetic mutation techniques address fitness-bound adaptations, which tend to increase the probability of mutation for low-fitness. The low-fitness mutation probability increment is so high such that the search becomes a random heuristic for low-fitness solutions and a pure Elitism approach for fitter solutions. In this paper, the proposed technique addresses a fractional mutation approach that adapts the mutation probability based on chromosomal sub-groups or containers to create clustered fitness groups. Accordingly, this research is different from other similar studies because of two major factors: it uses GEP/GP directly for optimisation and applies the proposed AFMR to improve the accuracy of forecasting.

1.3 Research Question, Aims and Objectives

Financial forecasting systems currently face challenges of non-linearity, missing data, noisy samples as well as large datasets. These problems have widely been addressed in the literature as discussed in the later sections and can be broadly be categorised as in the three core issues listed below:

- Ability of AI models to generalise training to address a wide range of financial datasets without recalibration.
- Ability of AI models to train with minimal computational overheads.

- Ability to enhance evolutionary operators to improve prediction accuracy.

Based on these challenges, the main question this research seeks to address is:
Can the representational parameters of an evolutionary algorithm achieve the balance between simplicity and accuracy?

The research is expected to achieve the following objectives:

- Objective 1: To explore the status of research in AI time-series analysis techniques and financial forecasting research.
- Objective 2: To explore various optimisation techniques reported in the literature that are used to improve learning/computational efficiencies of machine learning prediction algorithms.
- Objective 3: To investigate gaps in various research techniques and develop a novel methodology to improve prediction accuracy in financial time-series forecasting.
- Objective 4: To explore the impact of evolutionary optimisation algorithms over the computational efficiency of the machine learning technique developed in Objective 3.
- Objective 5: To compare the results obtained with the existing research baseline in order to establish the overall novelty achieved.

The complexity of modern financial systems causes difficulty in forecasting. Notably, the rise and fall of stock prices make it difficult to carry out financial predictions using expert-driven rule-based approaches. Besides, qualities such as staff skills and trading experience are unpredictable. Furthermore, peoples' decisions; which usually affect future stock-exchange prices, are difficult to forecast because individuals can make irrational choices. Accordingly, conventional forecasting tools such as statistical techniques are less effective in the modern financial setting; instead, AI approaches such as ANN, CBR and SVM-based methodology have been used increasingly in financial predictions. Particularly, modern businesses tend to use a hybrid of several AI techniques as a way of increasing accuracy. Recently, most studies have focused on ANN, Markov models and fuzzy inference/logic techniques. However, these approaches have various limitations such as slow training processes that require better methodologies. This research aims explore various optimisation techniques that are used in financial forecasting and establish ways of improving the accuracy and efficiency of forecasting models.

1.4 Summary of the Contributions

The proposed methodology investigates the effectiveness of applying GEP-based time-series prediction techniques to reduce the limitations of constant mutation rates by proposing a new mutation technique (FAMR) to ensure a diversification balance between high and low fitness solutions. Therefore, dynamic fitness-based adaptation was induced in the mutation process.

For a chromosome C, the mutation adjustment rule $R_{p(c)}$ would be defined as follows:

$$R_{p(c)} = \left(f_y/f_{max}\right) \times \left(p_{max} - p_{min}\right) \qquad (1)$$

Where f_y is an individual whose fitness is to be calculate adaptively and

f_{max} is the average maximum fitness of the top fittest chromosomes and

p_{max} and p_{min} are the minimum and maximum probabilities of the entire group

In Equation (1) we lower the probability of the fittest individual down with respect to its fitness.

In the next chapter, various concepts that are central to this research will be critically reviewed to explore the status of research in AI time-series analysis techniques and financial forecasting research.

2 Chapter2: Literature Review

Stock price forecasting has long been a focus of intelligent soft computing techniques to improve the predictability of financial systems. Accurate forecasting of time-based financial trends has become increasingly important because of rapidly changing trends in current global financial markets and the ongoing commercial uncertainties. Stock market forecasting provides investors with a general overview of the changing tendency of stock markets. Based on the forecasts, investors can make timely decisions on buying or selling stocks under bargains and avoid financial losses. A wide range of techniques that are not limited to econometric modelling but also involve artificial intelligence (AI) –based soft-computing techniques have been reported in various literature explore. These techniques include Statistical Analysis (SA), Artificial Neural Networks (ANN), Case Based Reasoning (CBR), Fuzzy Inference Systems (FIS), Decision Trees and Support Vector Machines (SVM) , among many others.

2.1 Financial Prediction Techniques

2.1.1 ANNs

ANNs generally operate over an undefined dataset, in which, when subjected to training data, the technique learns from irregularities and thereby, creates its own set of rules. The methodology heavily emphasises the comprehensiveness of data and, unlike fuzzy logic, is well-known for its ability to withstand noisy data and outliers (Liu, 2012). The methodology can also predict missing, sparse or low-quality values, which, because of nonlinearity, is the case with financial systems rife with uncertain data. Moreover, the genre is well known for its ability to handle input variables in a parallel fashion that allows handling of larger datasets efficiently; in turn, this quality provide ANNs with a significant edge over the manual rule-generation aspects of fuzzy logic. Additionally, these characteristics make ANN unique and with an ability to generalise over a diverse range of input/output data pairs and thus qualifies as an ideal replacement of the human-based fuzzy systems.

ANNs provide substantial benefits in financial investment and trading applications. The ability to generalise from a detailed training dataset enables ANN to model over a wide range of outlying cases, which makes it an ideal tool for time-series based non-linear systems such as stock price prediction, asset allocation and portfolio change forecasting. Wong and Selvi (1998) presented a detailed review of applications of ANN in the domain of finance. The review predominantly showed an increase in the use of soft-computing algorithms for the prediction of areas of bankruptcy prediction (Kumar & Ravi 2007), stock performance and selection analysis, life insurance assessment, bond trading forecasting, loan application reviewing, credit evaluation and future trends forecasting (Chen *et al.* 2012), and customer classification models (Ahn *et al.* 2011).

ANNs were initially employed by Connor, Martin, and Atlas (1994) where the outliers were "softly" removed from the data where the training was performed over the "outlier-filtered" data. The technique substantially improved the prediction accuracy of the system. However, in large-scale real-world systems, it is generally impractical to use "pre-training" clustering techniques for outlier removal. Moreover, there is a high probability that such a technique may also eliminate valid feature samples from the database as well. In order to address this issue, a hybrid ANN technique was proposed by Castillo and Melin (2002) via a neuro-fuzzy technique. The technique regulated the fuzzy membership functions by means of a single-layer feedforward neural network. The outcome of this work was far superior than generalised regression-based models. A similar work by Zhang and Berardi, (2001) utilised varied ANN structures over varied data partitions via varied initial random weights, random architectures and variable data and reported a considerable accuracy over conventional neural architectures.

Lately, ANNs research has moved into the analysis of noisy chaotic time-series prediction. According to Soofi and Cao (2002, p. 115), chaotic and non-linear time series prediction has significant effect on the economic and financial time series prediction. The characteristic is particularly prevalent in stock market prediction where the nonlinear feature data is normally marred by excessive noise. The optimum prediction of noisy time-series data was addressed by Leung, Lo and Wang (2001) via a radial basis function (RBF) neural network regression. In order to address the issue of generalisation against a large dataset, the technique utilises "cross-validated sub-space" method to identify a suitable number of hidden neurons to efficiently handle noise within the datasets. Recently, in-architecture neural network updates have been explored as Goh et al. (2009) created a neuron-level hyper-plane to separate noise from genuine feature samples. The technique, when combined with the nonlinear subspace, creates an optimal RBF predictor for variable signal-to-noise ratios (SNR).

Improvements in the neural architecture also involve the utilisation of the so-called "recurrent" ANN (RPNN), which facilitates long-term prediction (Han et al., 2004), local linear and wavelet-based transforms (Chen, Yang & Dong, 2006). Additionally, generalised regression-based ANN, counter-propagation technique, neural adaptive resonance regressions, CART decision trees, TreeNet-based data mining and random forests have also been used (Kumar & Ravi, 2007). Earlier on, in a ground-breaking research, Martinetz *et al.* (1993) presented an unsupervised technique based upon k-means clustering to present a "neural gas" network algorithm. The technique performed better compared to Kohonen-maps, K-means and Maximum-entropy, the methodology presented outstanding minimisation in vector quantisation coding distortion error. The methodology presented a faster convergence at a cost of higher computational effort.

Han *et al.* (2004) presented a recurrent ANN (RPNN) that facilitated long-term prediction by performing accurate multi-stop forecasts. The algorithm comprised of non-linearly operated internal nodes with outputs connected to mutual inputs. Based on the performance measurement via root-mean square error (RMSE) and prediction accuracy (PA), the technique outperformed Kalman filtering and Universal learning network (ULN) methodology for sun-spot prediction for a one-year duration. However, the algorithm did show a significant decline in performance for time durations of greater than 10 months.

2.1.2 FIS

Fuzzy Inference Systems employ FL (Fuzzy-Logic) to outline precise solutions to inexplicit problems. In the field of financial forecasting, a wide range of studies has been conducted to determine the application of FIS.

Castillo and Melin (2002) presented a fuzzy approach to handle a non-linear system and in turn, control financial dynamics by means of underlying mathematical models. In the following year, Yoshida (2003) introduced fuzzy logic to stochastic financial modelling in a bid to model expected European market prices based on the Black-Scholes formula. GenSo-EWS was developed as a toolkit by Tung *et al.* (2004) utilising the hybrid neuro-fuzzy methodology based upon the idea of a trained algorithm to detect banking sector failures based on inherent contributions of a set of financial covariates. Similarly, Tang and Chi (2005) performed an empirical investigation of 3344 listed firms to classify and predict trade credit risk by applying receiver operating characteristic (ROC) analysis to compare Logit performance with fuzzy logic. Wei *et al.* (2011) presented a hybridisation of adaptive neuro-fuzzy inference (ANFIS) methodology to implement a selective forecasting model. The hybrid methodology addressed three objectives. The first objective involved selection of pertinent technical indicators from a set of indicators by means of a correlation matrix. Wei *et al.* (2011) also aimed to create a fuzzy linguistic variables based on subtractive clustering based on a data discretisation method. The third objective involved the development of an FIS to extract rules of linguistic variables stated above by means of an adaptive neuro-fuzzy system (ANFIS).

Chang *et al.* (2009) presented a "tri-regression" K-means clustering based fuzzy neural network to implement a sales forecasting model. The methodology divided historic data into historical clusters to be used to predict the future monthly forecasts based upon a 5-yearly dataset. The approach was demonstrated to be superior from existing traditional approaches, including ANFIS by means of forecasted errors (RMSE) and accuracy of forecasted results (MAPE) techniques. Huang (2009) proposed a fuzzy c-means clustering methodology to predict stock market trends and optimal stock portfolio selection based upon an ARX prediction model, Grey system theory and variable precision rough set (VPRS) theory. The methodology compared the performance of the

VPRS-based approach with the rough-set (RS)-based approach using stock exchange data. Li *et al.* (2012) presented an evolutionary chaotic swarm fuzzy clustering (CPSFC) algorithm to improve two core aspects of predictive clustering namely: PSO and chaos search and combining a grading method with FCM algorithm. The combination of the gradient method produced better methods compared to PSO and GA methodologies. However, it must be noted that the proposed method is still computationally complex when compared with the PSO or GA-based algorithms specifically when large datasets with multiple dimensions are used particularly in time-series financial applications.

2.2 Limitations of these Techniques and Time-Series Analysis

However, despite the major advances made in financial forecasting methodologies, numerous limitations that necessitate the development of a completely new approach exist. These majorly are caused by difficulties in time series analysis. A time-series is regarded as a sequence of stochastic variables whose behaviour depends on a number of real-world factors or dependent variables that decide the values of the next variables, ahead in time, based on past trends. Time-series analysis provides tools to select models that are then used to predict future events as a statistical time-series problem. These statistical predictions are based on the notion that the observations are based on a probability distribution function. In time series forecasting, financial data is often marred by missing variables, data and noise. In addition, during the selection phase of such parameters before the training phase could commence, it is generally extremely difficult to ascertain the most optimal set of variables that could be used to train the underlying ANN for the FIS membership function optimisation. However, a major disadvantage of ANNs is over-training, which could lead to unstable prediction capabilities. Prediction instability is partially addressed by dividing the dataset into three groups that include training, test and validation sets. In these groups, the algorithm is stopped when the error margin repeatedly increases over a consecutive number of iterations. However, the process does not fully eliminate involved errors. The main drawback of ANFIS and FIS systems arise from prediction complexities as well.

This dissertation aims to address the limitation by improving an evolutionary computing algorithm to improve the overall modelling and prediction accuracy. Based on the existing machine learning research limitations discovered in this literature survey, particularly in handling time-series data, this research focuses on the investigation, extension and evolutionary technique to improve and present a novel stock trend prediction technique.

2.3 Genetic Algorithms for optimisation of time-series-based systems

John Holland originally introduced genetic algorithms in the 1970s in a bid to study the notion of adaptation as it originally occurs in natural environment (Holland 1975). Holland's GA is a technique to encode real-world problems whose solution is sought onto genetic chromosomes, which are made up of binary bits comprising of "ones" or "zeros". Each chromosome represents the "body of a solution" which is evaluated for its fitness against an objective function. Based on the principles of natural evolution, each solution is tried in subsequent generations by means of specialist evolutionary operators such as crossover, mutation and inversion in a bid to retain the best possible child chromosomes and eliminate the worst ones. Thus, during a course of genetic run, the best available child solution is further improved by adapting the best trait of its parents while discarding the worst-ones. GA algorithms are not directly used for recognising patterns or identifying system properties. The algorithms are used to optimise the parameters of operational AI techniques such as neural networks, support vector machines or fuzzy inference by optimising their parameters.

GAs were initially utilised for sub-functional optimisation of parametric tuning of various AI algorithms. For instance, Kim and Han (2000) report the utilisation of GA for feature discretisation and tuning of connection weights of an ANN algorithm. Similarly, Genetic Monte Carlo method has been used to train Bayesian Neural Networks for the non-linear time-series forecasting (Kocadağlı, 2014). However, as the baseline concept still use ANN, these algorithms still pose a limitation of getting stuck into a local minima; the so-called random walk problem and the computational inefficiency for problems with high number of variables such as bin-packing, travelling salesperson, location-allocation and multi-variable stock-market prediction problems. Despite the comparative efficiency of GA, its inherent hill-climbing process is still computationally complex. Straßburg et al. (2012) recently proposed a parallel GA technique utilising the multi-core architecture for the prediction of stock market trading rules.

Standalone “GA” were recently integrated by Stepanek et al. in an existing multi-agent stock market simulation algorithm. Rough-set theory (RST) integrated with GA by Cheng et al. (2010) reported market trend identification (bullish or bearish). RST did improve the accuracy marginally when compared to GA, thereby proving its better capability in generating rules from data pairs. However, it also reported the shortcomings of individual RST and GA in the prediction of market trends especially in the presence of varying and conflicting market parameters. A Modified GA methodology was proposed by Araújo et al. (2009) based on a morphological-rank-linear (MRL) filter capable of finding sub-time-series anomalies to filter out initial sub-optimal parameters, thereby improving the overall prediction rate. A multi-objective stock index tracking methodology was proposed by Ni and Wang (2013) based on profitability, stability and volatility of stocks. The domain has recently been extended via the generation of an ensemble of recurrent neural networks via evolutionary multi-objective GA that tune the recurrent neural architecture with the best set of models selected on the basis of a Pareto front (Smith & Jin, 2014).

2.4 Evolutionary algorithm in time series optimisation

The evolutionary computing domain has been extensively exploited for the stock market prediction algorithms. However, general issues of over-fitting, data snooping bias and black-boxed characteristics are cited as factors that challenge the reliability of these algorithms. Genetic Programming (GP) has been used to generate trading rules as an alternative to the well-known buy-and-hold approach of stock buying. Potvin et al. (2004) focused on individual stock-based adjustment technique to predict short-term fluctuations. Long-term prediction, on the other hand, offers its own complexities due to over-time error accumulation and inherent uncertainty due to large time-durations involved. Therefore, in stock prediction, long-term prediction takes a completely different perspective compared to the medium-to-short term case.

Sovilj et al. (2010) utilised optimally pruned extreme learning machine (OPELM) and optimally k-nearest-neighbours (OPKNN) models that are known for their fast training times and accurate predictions. The former OPELM algorithm relies on a single-layer fast feed-forward neural network, whereas the later OPKNN algorithm relies on k nearest neighbours as the kernel function. OPKNN is faster than its former counterpart OPELM algorithm due to its deterministic nature as it does not use any additional parameters. Another approach to long-term time-series forecasting was proposed by Stojanović et al. (2014) based on mutual information (MI) used in regression tasks for feature selection. The methodology's main contribution has been to differentiate useful data from outliers by recursively reducing uncertainty in the training set by eliminating noisy datasets from the training stage. In a similar cash flow prediction domain, Weighted SVM and fuzzy logic were combined-mapped to fast-messy GA chromosomes in a bid to address vagueness via fuzzy logic and improve temporal prediction via improved input/output mapping (Cheng & Roy, 2011).

In addition, a fuzzy-genetic approach was adopted by Aladag et al. (2014) to remove the so-called "lagged variables" from a fuzzy time-series predicting regression via a GA-based selection process. The work evaluated its algorithms against the Taiwanese stock exchange data and reported a marked improvement in the removal of fuzzy uncertainty from the predicted outcome. A similar study by Cai et al. (2013) encoded entire fuzzy inference systems along with the parameters into a GA-based run. Wei (2013) offered a GA-weighted adaptive network based fuzzy inference system (ANFIS) methodology to tune the membership functions of a fuzzy inference system. Xiao-qin (2012) implemented a similar fuzzy-genetic technique to fuzzify the parameters of the GA. These GA implementations bear significant advantages to financial marketing forecasting. As noted by Hewahi (2015), GA parameters can be integrated with approaches such as HMM and ANN to realise highly efficient financial forecasting systems. The combination achieves high prediction capabilities because the developed system obtains the strengths of each forecasting model.

Additionally, Particle Swarm Optimisation (PSO) is a domain of evolutionary computing algorithms commonly used to predict data patterns based upon a set of temporally varying parameters similar to a flock of birds. Similar to other evolutionary computing approaches, this problem also suffers from getting stuck into local minima. Pulido et al. (2014) introduced PSO to and ensemble of ANNs to tune a Type1/Type2 fuzzy system. The technique eliminated the need of manual adjustment of fuzzy rules via a set of ANNs whose parameters were dynamically tuned via the PSO routine. Similar ideas have also been used to individually analyse internal stock market trends via dynamic fuzzy models and reported several findings. One of the results is that GAs offer better performance due to their faster solution finding capabilities via their hill-climbing nature (Cheng et al. 2010, Araújo et al. 2009). In addition, it has also been shown that multi-SVM is more capable in predicting multiple stock activity periods when compared to dual SVM (Cheng & Roy, 2011). Furthermore, a study by Cheng and Roy (2011) found out that SVM outperform ANN and discriminant analysis-based approaches. Additionally, fuzzy model improve accuracies in any time-series models (Cai et al. 2013; Wei 2013; Xiao-qin 2012).

Burke and Newall (1999) also evaluated a multi-stage evolutionary search heuristic to search for most optimal GA solutions within a search space. The technique focused more on earlier and intelligent selection of chromosomes in a bid to improve the optimisation convergence earlier on in the genetic run based on a set of hard and soft constraints. Later on, Dipti et al. (2002) focused more on constraints satisfaction and local search improvement measures. This technique offered a potential to handle restricted datasets for instance if incomplete historic data was available for training. Yet, these techniques were still limited in terms of specialist selection of operators to generate newer solutions (child chromosomes). The majority selected single-cross-over techniques which did not change the child solutions substantially from their parents. Bhatt and Sahajpal (2004) and Sabri et al. (2010) utilised multiple crossover technique to improve the fitness by improving the swapping probability of “good traits” of previous parental prediction solutions. Further improvements were obtained via constraints conflict minimisation by Ghaemi et al. (2007), rule readjustment (Karova 2004), localised solution search (Abdullah & Turabieh 2008) and 3D chromosomal structuring (Sigl et al. 2003).

Similarly, group behaviour of birds was also adopted in evolutionary research and commonly termed as the Particle Swarm Optimisation (PSO). PSO is an iterative methodology aimed at incrementing the suitability of a solution by iteratively improving the output against a quality measuring function. Nenortaite and Simutis (2005) and Junyou (2007) used ANN to analyse historical data and PSO used to train ANN for improved stock prediction. Yet, the algorithm mainly suffers from issues of partial optimisation and an overly sparse solution space where the latter is particularly an issue in a rich solution space like the stock market. Tabu-Search (TS) is another meta-heuristic solution search method which employs numerical optimisation to improve the fitness of its best solution. The hybrid "Tabu-Monte-Carlo" algorithms and improved multi-neighbourhood structures are used by Khang et al. (2010) and Al Tarawneh and Ayob (2011) to increase the global fitness objective function and reduce the convergence time. However, TS is known for its extra computational cost during its local search which induces an extra complexity in the approach. Moreover, stock and financial time-series problems are known to have higher number of objective functions which may lead to design difficulties in TS algorithms.

Moreover, Ant Colony Optimisation (ACO) systems are another genre developed by Marco Dorigo. The ACO was inspired by ants' ways of finding solutions by depositing a chemical substance called pheromone. When ants discover a shorter or easier path, they deposit considerable amounts of pheromone that act as the path's trail. Gao (2008) integrated a neural regression with ACO to improve prediction of short-term stock market. A unique, multi-pheromone regression with varied evaporation speeds was used by Hsien-Kai Hsin et al. (2011) used exponential moving average to improve in a similar case but different application time-series prediction. However, unlike the GA, ACO algorithms are known for uncertain convergence times and random decisions of varied suitability which makes it difficult to analyse the reliability of ACO regressions.

In the field of time series forecasting, this genre of evolutionary computing has been used extensively. For instance, the technique is utilised in the prediction of chaotic time series prediction by (Szpiro 1997) to predict noisy and sun-spot series data as well as Henon and universal maps (Yadavalli et al. 1999). The algorithms are also utilised by

Chen and Chung (2006) via higher order fuzzy time series to predict university enrolments. The technique was also used in conjunction with support vector machines (SVM) to classify lightening in higher space as SVM have a reputation for classifying in high-dimensional spaces without overfitting (Eads et al. 2002). GAs are most frequently used in the literature to optimise the training process of ANN. The integration aims to determine better neural architectures and improve the overall generalisation capabilities of ANN. The underlying notion is related to the fact that ANN parameter tuning is generally a manual task which is normally done via a hit-and-trial error method. Hence, GA is also used to train ANN in the case of missing observations or noisy data marred by a large number of outliers (Hung 2008, Baragona et al. 2001). Moreover, the technique is extensively used in real-world problems where the search space is large enough to attain an optimal solution merely by a brute-force solution. Most well-known problems that are NP (non-polynomial)-Hard, such as traffic flow and routing, time series weather forecasting and other flow or route optimisation problems, are NP-complete which means that a viable solution by brute force for these problems is not achievable in finite time.

However, because of the highly complex nature of stock markets, factors tend to interact with each other in a non-deterministic way and all these interactions cannot be modelled because of increased computational complexity of the underlying system. Models can be improved via dimensionality reduction techniques. Moreover, all of the sub-methodological techniques discussed above somewhat improved the generations of offspring. Yet, the majority of these techniques are still restricted by factors such as extremely long convergence durations and inability to select better selection routines. Furthermore, classic GAs are incapable of accurately predicting the best time to stop a genetic run with the highest fitness and this leads to the loss of best solutions. These incapabilities gave way to techniques that retained best performing individuals amid the risks that they probabilistically mated with one of the least suitable solutions and generated poorer offspring. Based on the analysis of various algorithms, GA offers a promising opportunity if integrated with Randomised Algorithms. GA is predominantly dependent upon a set time-period, number-of-generations or a fitness criterion to terminate its execution. There still exists a risk of premature termination. If the algorithm is continued indiscriminately, the likelihood of GA continuing forever is very high thereby resulting in a computationally intensive solution. It is envisaged that a Randomised Algorithm approach will improve the outcome, which can be integrated into classic GAs and thereby, enhance the prediction models used in financial market forecasting.

2.5 Gene expression programming in time series optimisation

Much research to date has been limited to utilising GEP to optimising the operation of other machine learning regressions including artificial neural networks (ANN) and fuzzy logic. In the domain of stock forecasting, Gene Expression Programming is addressed in the existing literature by Bautu et al. (2010), Garg et al. (2013), Ye et al. (2009) and Hongbin et al. (2010). However, the majority of this work focuses on optimising ensembles of other AI algorithms such as Artificial Neural Networks (ANN), Support Vector Machines (SVM) and Naive Bayes.

Bautu (2010) primarily focused on modelling an ensemble of regressions based on the Efficient Market Hypothesis (EMH). The theory is based upon the fact that it is not possible to beat the market because stock market efficiency always incorporates all the factors affecting the market such as socio-political changes, natural disasters and other unexpected localised fluctuations. This means the stocks always trade at their fair value, which makes it impossible for traders to purchase shares at lower-than-market or higher prices. Consequently, it is not possible to outperform market via expert-driven stock selection or market timing. Therefore, EMH states that the only way to obtain better returns is by purchasing higher-risk investments. The underlying idea of this work can lead to generating best performance equations from training data, which can then be used as relative fitness functions in the proposed work.

Bautu et al. (2010) utilised an evolutionary perspective to uncover quantitative patterns in the stock market performance. Contrary to the proposed approach, the GEP approach in this work utilises logical expression trees with a binary classification approach. The approach achieves diversity by starting the genetic run at random seeds. The approach uses an ensemble of GEP regressions that run against input/output data pairs. This approach, despite its merits of an increased confidence in multiple models running in parallel to increase the classification reliability, is still time-consuming compared to an approach where an optimal solution is obtained via single genetic classification run operating on a generation of chromosomes that started at their own pace. Hence, GEP in this approach is not directly used to predict certain stock prices but to identify the most optimal set of other regressions including Naive Bayes, Support Vector Machine, and Multi-layer Perceptron, that provided the best classification outcome.

Similarly, extended work done by Barbulescu and Bautu (2012) also focuses on combining ARMA and GEP-induced models to exploit ARMA's capability to identify linear trends and GP to classify nonlinear trends in the data. As the focus of this research primarily stays on ARMA models, the identification and estimation is highly likely to be distorted by outliers. Additionally, work done by Grosan and Abraham (2006) also proposes the performance analysis of an ensemble of four measures namely Root Mean Squared Error (RMSE), Maximum Absolute Percentage Error (MAP), Correlation Coefficient (CC), and Mean Absolute Percentage Error (MAPE). Similarly, the work by Garg et al. (2013) also used Genetic Programming (GP), focusing on model selection criterion and data transformation. Hongbin et al. (2010) focused more on a hybrid GEP-ANN approach with an objective to improve complex problems, enhance learning speed and generalise of Back Propagation ANN. The main objective of this research was to improve the prediction accuracy of ANN.

Similar work done directly in time-series analysis of stock market via GEP was reported by Duan et al. (2011). Two methods, either the Sliding Window Prediction Method (GEP-SWPM) or Differential Equation Prediction Method (GEP-DEPM), are commonly used in this form of research. The former (GEP-SWPM) technique only addresses a past-to-future data directly. The latter (GEP-DEPM) approach proposes a differential equation prediction method to mine a differential equation from training data to forecast future stock values. However, both the approaches only evaluate their outcomes on sunspot activity. Moreover, differential equations predictions are known to have a higher computational complexity compared to standard quadratic with a comparatively high sensitivity to noisy data (Chen et al., 2011).

Based on the review given above, the majority of time-series prediction approaches rely on a fixed-length time-duration, which is used as an input vector to predict the identification of current time instance. A variety of AI approach is integrated with GA to address the problem of variability of prediction. However, a global solution that can incorporate numerous time-series length models to generate an optimal prediction system is still non-existent.

The next chapter investigates and implements an existing methodology to act as a benchmark for the later research. The chapter helps achieve Objective 3, which aimed to investigate gaps in various research techniques and develop a novel methodology to improve prediction accuracy in financial time-series forecasting. The main methodologies covered are FIS-related systems and sliding window prediction module (SWPM).

3 Chapter3: Development of a Neuro-Fuzzy Financial Forecasting System

Fuzzy Logic theory was laid out by Lotfi A. Zadeh, who invented the fuzzy sets and the mathematical basis of the fuzzy set theory in 1965 (Zadeh, 1996). The research in fuzzy logic has drawn substantial attention during the past two decades and the paradigm is now widely used for predicting nonlinear and uncertain systems drawn from a wide range of real-world domains including signal data mining (Zhu, Wang & Ren, 2010), information retrieval (Yih-Jen et al., 2005), finance (Lee & Smith, 1995) and various real-world forecasting systems including stocks, resource demand and supply, power requirement, and sensor networks (Hiemstra, 1994; Ghalia & Wang, 2000; Qiaolin, Jing & Jianxin, 2005; Ollos & Vida, 2011). However, despite a continued and high demand for this soft-computing technique, the technique has a number of limitations. Fuzzy Inference Systems (FIS) generally require considerable human intervention to accurately and realistically predict certain situations and therefore, potentially leads high chances of human-based error occurrences in the system. Moreover, with the increase in the system variables, the complexity of the system increases substantially thereby leaving pure fuzzy logic only suitable for moderate-complexity systems. For instance, a weather forecasting system taking humidity, average day temperature, UV and Real Feel as four variables with three levels of temperatures to be forecasted as low, medium and high will require a weather expert to generate $4^3 = 64$ rules to cover the whole system. These limitations make a pure FIS unsuitable for complex systems such as those of financial markets forecasting.

3.1 A Sliding window-based FIS system to identify forecast anomalies

Based on the research question elaborated earlier in chapter 1 part 1.2, this section presents the methodology along with the resultant addition to the state of current knowledge.

The first module of the system was aimed at using a financial domain analyst's expert knowledge to build a fuzzy inference system by manually tuning the membership functions. However, because this first module depended on a single data source and thus was associated with considerably high potential bias and number of assumptions, as assumed by the domain analyst expert, it was eventually replaced with the proposed neuro-fuzzy paradigm. In this second model, the ANN was used to learn from historic data, which, unlike the first model, provides wide-ranging information and hence considerably minimises potential bias and assumptions; in this case, ANN acted as the expert AI-based trainer. The outcome of this module broadly entails an ANN-tuned FIS that was aimed to predict time-series-based data to predict future forecasts. The features extracted for this model bases its data clustering on a balanced sliding window kurtosis value taken on both sides. The resultant statistical trend provides the assessment of subtle data anomalies within complex time-series forecasts. The sample stock data to explain the underlying concept was downloaded from Yahoo Finance (2012). The data contains a daily trading volume of stock volume and prices from 12/04/1996 to 31/08/2012 containing the following five parameters:

1. Open (share price)
2. High (share price)
3. Low (share price)
4. End-of-day Close (share price)
5. Volume (trade volume in US$)

The data is extracted for adaptive neuro-fuzzy training based on a sliding-window operation as shown in Figure 1.

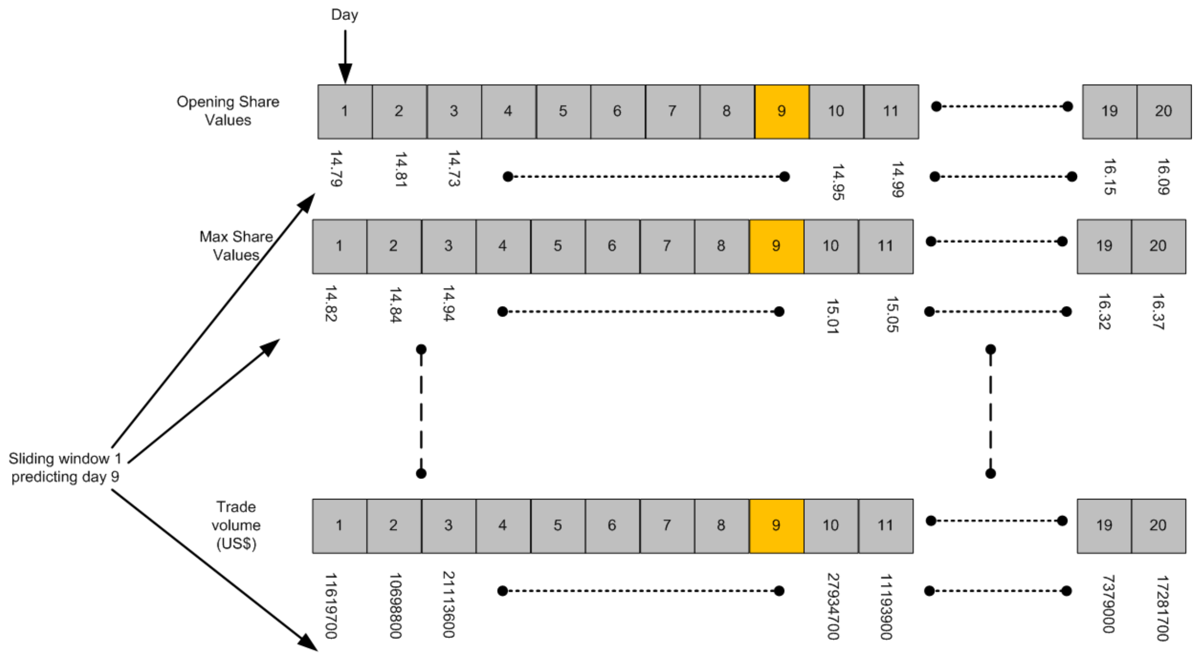


Figure 1: Elaboration of a sliding window operation to gather feature vector data for neural network training.

Based on the single-step (one-day) sliding window operation, a feature vector containing a set of input vectors and the output (closing value) will be obtained in a row-wise fashion. Each row will represent a single day prediction based on the previous 'n' number of days where n = 8 in the case shown in Figure 1. The feature vector format for two single-step functions of the sliding window operation is shown in Table 1.

Input Feature Vector (i = 1) Single instance of sliding window operation to predict the value of Day 9								Output
Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9
20/08/2012	21/08/2012	22/08/2012	23/08/2012	24/08/2012	27/08/2012	28/08/2012	29/08/2012	30/08/2012
14.99	14.95	14.95	14.9	14.82	14.92	14.84	14.73	14.81
15.05	15.01	14.99	14.97	14.94	14.93	14.87	14.94	14.84
14.88	14.88	14.86	14.82	14.77	14.77	14.69	14.7	14.64
14.96	14.97	14.92	14.87	14.92	14.85	14.72	14.84	14.67
11193900	27934700	9168400	12463000	8650400	10054000	12706400	21113600	10698800
14.96	14.97	14.92	14.87	14.92	14.85	14.72	14.84	14.67
Input Feature Vector (i = 2) Single instance of sliding window operation to predict the value of Day 10								Output
Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
21/08/2012	22/08/2012	23/08/2012	24/08/2012	27/08/2012	28/08/2012	29/08/2012	30/08/2012	31/08/2012
14.95	14.95	14.9	14.82	14.92	14.84	14.73	14.81	14.79
15.01	14.99	14.97	14.94	14.93	14.87	14.94	14.84	14.82
14.88	14.86	14.82	14.77	14.77	14.69	14.7	14.64	14.59
14.97	14.92	14.87	14.92	14.85	14.72	14.84	14.67	14.65
27934700	9168400	12463000	8650400	10054000	12706400	21113600	10698800	11619700
14.97	14.92	14.87	14.92	14.85	14.72	14.84	14.67	14.65

Table 1: Sample data taken from Yahoo (2012) for day nine and day ten based upon the previous eight day values.

3.2 Integration of stock data to train an FIS

This stage utilises the generalisation abilities of ANN to tune the linguistic and membership function properties of the FIS stated above. It is envisaged that this hybridisation can expedite the overall rule generation capability of the FIS by tuning the underlying membership functions. However, as the correct type and most optimal number of parameters used to train the neural network are not yet known, an additional optimisation technique is still required to search the best model configuration.

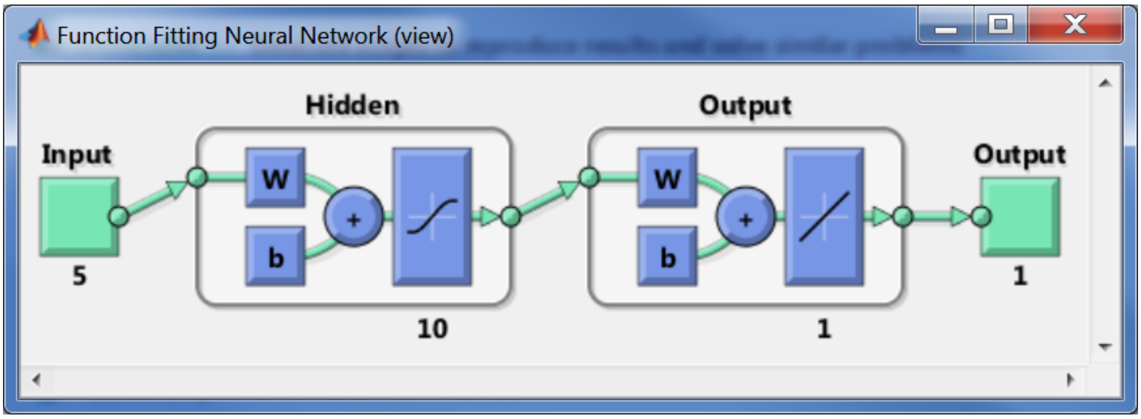


Figure 2: A 5 x 1 input-output NN overview with 10 hidden neurons, a single output neuron and a single decimal output predicting the next-day share value trained via feature values shown in Figure 1.

Based on the way the "if-then-else" rules are derived, the majority of FIS are divided into two main genres of Mamdani and Takagi-Sugeno Kang (TSK) Systems. Mamdani is generally used in the majority of cases, but TSK is well known for its computational efficiency and compactness. The TSK method was proposed by Takagi and Sugeno in order derive a set of rules from input/output training data pairs. An important aspect of TSK is its crisp outcome which significantly reduces its computational complexity when compared to its Mamdani counterpart.

A typical rule in a Takagi-Sugeno-Kang FIS is shown in Equation (2):

$$i : IF\ x\ is\ A_i\ and\ y\ is\ B_i\ THEN\ f_i = p_i x + q_i y + r_i \qquad (2)$$

In Equation (2), i is the rule number and A_i and B_i are corresponding fuzzy sets to each linguistic label domain, f_i is the output set covered by the fuzzy rule in the fuzzy region and $p_i,\ q_i\ and\ r_i$ are linear output parameters.

3.2.1 FIS rule-based generation via subtractive clustering and grid-partitioning

Rule-based in fuzzy systems are generated by expert engineers with in-depth knowledge of the underlying domain either when a good database is not available or does not cover the whole modelling scenario. However, to generate a comprehensive rule-base, which portrays the exact relationship between the input/output feature sets, the variable space must be efficiently clustered.

For instance, in a fuzzy c-means clustering algorithm, each data point belongs to each of the clusters based on some degree of membership. Therefore, the closer a point is to the mean position of a cluster, the higher its membership to that cluster is. For instance, the weight of a person may be attributed to two different clusters of individuals with one cluster classified as those being obese and the other being of average weight. Based upon a specific data point's (person's) weight's distance to the centre point of both of these clusters, the data points membership can be 0.33 Obese and 0.67 Average_Weight, effectively assigning him/her to belong predominantly to an Average_Weight cluster.

In the time-series-based stock value prediction case, the rules are drawn from multiple variables including opening, high, low and trading volume values. These variables can be bound to the input-space via a number of partitioning methodologies including grid (Ishibuchi & Nakashima, 1998), tree (Jae Hung & Sethi, 1995) and scatter partitioning (Shinn-Ying et al., 2004). The earlier, grid-based clustering is generally deemed appropriate for systems with lower number of membership functions and input variables. This is primarily due to the fact that the methodology's computational complexity increases exponentially with the increase in the number of membership functions and input variables (Mathworks, 2014b).

For the proposed system, estimated Equation (3) elaborates on the number of rules required for a complete FIS with two input variables (ϑ_t and δ_t) and three membership functions, namely LOW, $MEDIUM$ and $HIGH$ where ϑ_t is the trade volume and $\delta_t$is the stock value at a time-instance t. The total number of rules $\mathbb{R}$ for this complete FIS can be defined as follows (3):

$$\mathbb{R} = \sum_{i=1}^{N} \mathcal{V}^{\sum_{i=1}^{M} MF} = 2^3 = 8 \qquad (3)$$

In Equation (3), N is the total number of variables $\mathcal{V}$ and M is the total number of membership functions desired in the implemented system. The mapping of these rules via a grid-partitioning system for trade-volume and stock-value input variable is shown in Figure 3 below.

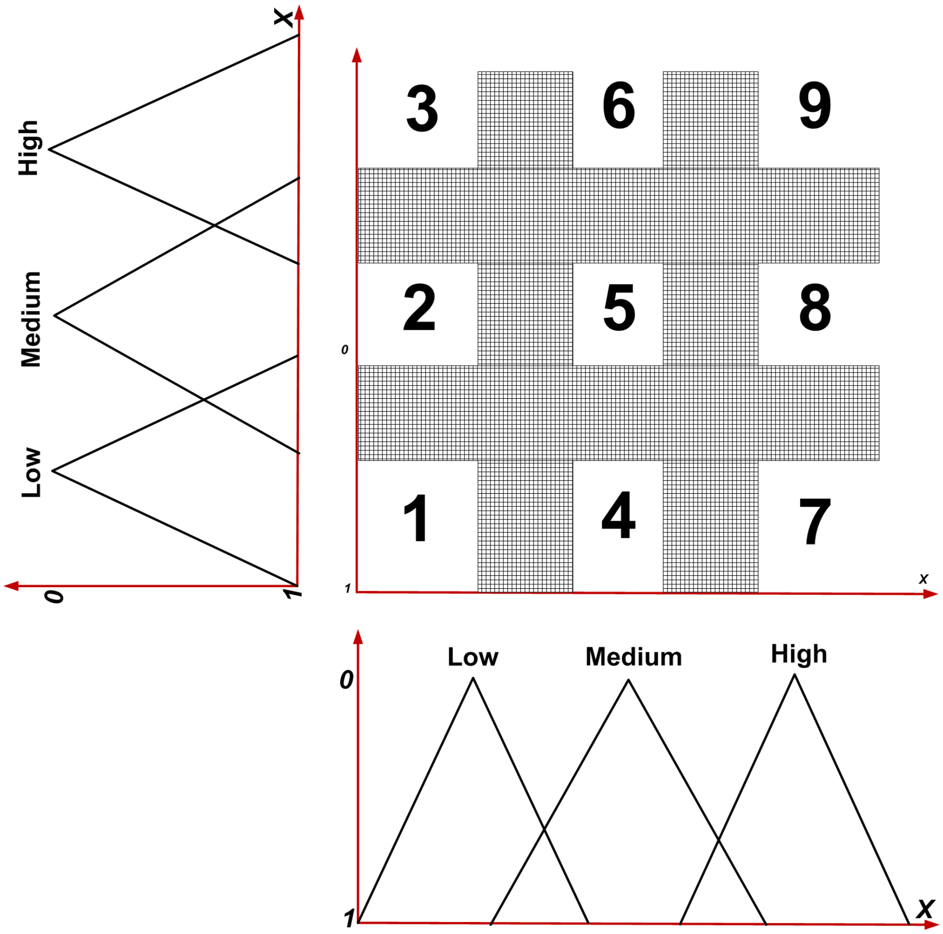


Figure 3: A fuzzy-rule sub-space for a two-variable stock-market prediction system.

Figure 3 represents an example of a, two-input, two-rule, TSK-type fuzzy reasoning methodology. First order TSK systems can be defined and visualised as a pointer which moves linearly in an outer space based on the value of the antecedent variables. As each rule in the FIS database is associated to the input variables, the TSK system is suitable for systems requiring interpolation of multiple linear inputs. It must however be noted that for a system with significantly high number of input variables and membership functions, the number of rules ℝ will increase phenomenally. For such a case alternative techniques such as subtractive clustering was proposed on the basis of a single-pass algorithm for number of cluster and centre estimation (Chiu, 1994).

3.2.2 Formulation of a neuro-fuzzy approach for financial time-series estimation

The proposed system implements a neuro-fuzzy approach where the ANN technique is used to tuned FIS parameters. The resultant methodology is widely known as an adaptive network based fuzzy inference system (ANFIS) which utilises training feature data to induce fuzzy rules via neural training-based weight adjustment.

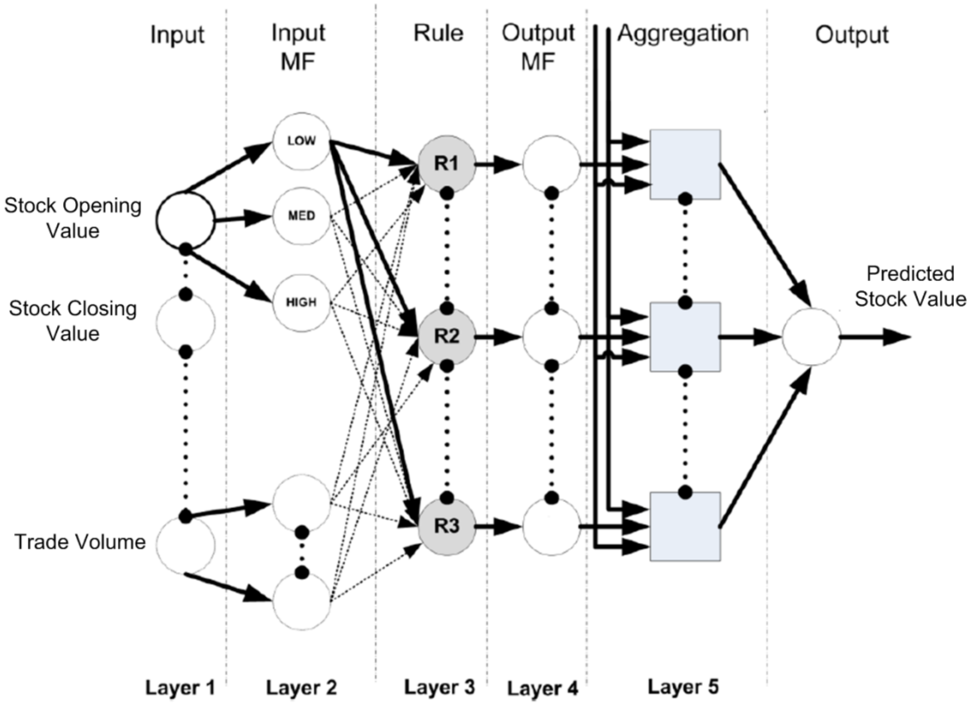


Figure 4: The design of a proposed TSK FIS based ANFIS framework utilising 4 input variables, respective input membership functions, rules and aggregation as hidden layers and output stock prices as the predictive outcome.

A wider framework for the proposed TSK ANFIS to predict stock prices is shown in Figure 4 above, where each layer is further explained below.

Layer – 1: Calculation of membership values for the premise parameter

The nodes in this layer are adaptive and the node output is the extent to which the given input fulfils the underlying (associated) linguistic variable associated with this node defined as follows (Lughofer & Klement, 2004):

$$\mu\, A_i(x_1) = {}^{1}\!\Big/_{(1 + |x_1 - {}^{c_i}/_{a_i}|)^{2b_i}} \qquad (4)$$

In Equation (4) $\mathrm{x_1}$ is the input to the node and $\mathrm{a, b, c}$ are adjustable factor variable termed as premise parameters. The layer outputs the membership values of the premise part where an ANN back propagation algorithm is used during the learning stage. The premise parameters are used to define membership functions which are generally fine-tuned via a Gradient-Descent method. As the subsequent values of the parameters change, the linguistic term's membership function $\mu\, \mathrm{A_i(x_1)}$ changes as well. That is, the closer a parameter is to a certain membership, the clearer its association to a certain group is. In other words, the membership grade of a fuzzy set specifies the degree to which the given input satisfies the quantifier

In the proposed stock price prediction problem, if closing price at time instance t is δ_t^i which is an input variable with three membership values of HIGH, MEDIUM and LOW, the three nodes are kept in the Layer – 1 and denoted via various membership function types.

Layer – 2 : The fuzzification layer

In Layer – 2, the nodes are kept fixed with each expressing one linguistic variable (e.g. MEDIUM) mapped to one input variable in layer 1. The output at this layer is a membership value between 0 – 1 specifying the extent to which an input variable belongs to a specific set. This extent is also regarded as the firing strength of the rules.

The node multiplies input signals from the preceding layer and generates a weight product given below (ANFIS 2013):

$$\omega_1 = \mu\, A_i(x_1)\mu\, B_i(x_2) \tag{5}$$

The total number of nodes in Layer – 2 is represented by $\mathbb{R}^n$, which for an $n = 4$ variable complete FIS system with three membership function levels generates a total of $3^4 = 81$ rules. The rule strength calculation is further shown in Figure 8 where a clustering algorithm as elaborated earlier on decides the initial number and type of membership function to be allocated to each of the variable type.

Layer – 3: Rule-strength normalisation

The output to this layer, represented by a fixed number of nodes, is the rule's antecedent part which is the firing strength of the fuzzy rule in its normalised form represented as a $\mathbb{T} - \text{norm}$. The i^{th} node in this layer calculates the i^{th} rule's firing strength ratio to the firing strength of the sum of all rules as described in Equation (6) (ANFIS 2013).

$$\overline{\omega}_i = \frac{\omega_i}{\sum_{j=1}^{R} \omega_j} \tag{6}$$

In Equation (6), ω_i is the firing strength of the i^{th} rule computed in the previous Layer – 2 which is represented via $\mathbb{R}^n$ total number of rules.

Layer – 4: The Rule-Consequent Layer

The nodes in this layer are not fixed and adaptively change where, for every i^{th} node, a linear function is computed whose coefficients are adapted by an error function. The error function in Equation (7):

$$\overline{\omega_\iota f_\iota} = \omega_i(p_i x_1 + q_i x_2 + r_i) \tag{7}$$

In Equation (7), ω_i is the weight output of the input layer (Layer – 4), whereas p_i, q_i, r_i are the parameter set where i represents the total inputs to the system. These parameters are also called the "consequent parameters".

Layer-5

The single node in this layer is fixed node as sum where at this stage the overall subsequent output is computed by summing all the input signals. The final summation is shown Equation (8) as follows:

$$\sum_i \overline{\omega_i} f_i = \sum_i \omega_i f_i \Big/ \sum \omega_i \qquad (8)$$

For forecasting problems as defined under the scope of this project, FIS and ANN techniques can be integrated in two different ways. Fuzzy logic can be used to tune neural parameters to improve its generalisation capabilities. This induces fuzzy logic into neural layers, inputs/outputs, weights, and aggregation and activation functions. This sort of modelling does not address the uncertainty improvement objective of the proposed research. For instance, introducing a fuzzy activation/triggering function via fuzzy rules may not necessarily improve the overall prediction accuracy of the output. Alternatively, neural models can be used to improve various aspects of an FIS. Fuzzy logic is well-known for its ability to induce intelligence via expert-defined-rules. However, the tuning of linguistic labels, membership functions, and adjustment of rules require a lot of time. For ANNs, it is extremely complex to gain the structural understanding of a trained neural network. Moreover, a low learning process along with the hidden layers makes the real-time analysis of ANN a cumbersome task. Nonetheless, the ability of ANN to learn from input/output datasets makes it an ideal AI methodology to tune generally manual and expert-controlled antecedent/consequent pairs from fuzzy logic rule data (Garcia & Mendez, 2007).

This chapter aimed to investigate and implement an existing methodology to act as a benchmark for later research. Particularly, ANFIS system was employed for this analysis. The outcome obtained from this analysis will be help build the thrust of the main methodology of this thesis: GP/GEP and act as a benchmark for making comparison.

More details about the algorithm are further explained in chapter 5 and 6.In the next chapter, these gaps are used to develop a novel methodology (GEP) that helps improve the time-series-based prediction accuracy of the stock price forecasting systems.

4 Chapter4: Gene Expression Programming For Optimisation In Stock Prediction Algorithms

This chapter provides a novel detailed methodology of the proposed evolutionary optimisation algorithm used to improve the time-series-based prediction accuracy of the stock price forecasting system and therefore, helps achieve the study's Objective 4.

4.1 The Performance Of The Proposed GEP

The proposed GEP technique undergoes an iterative genetic selection process, which is represented by three genetic operators. The first selection-operator selects a candidate solution (chromosome) from the current generation of chromosomes that is then probabilistically manipulated to be either included or eliminated to the next generation based upon a set of fitness criteria. Before being selected into the next generation, a candidate chromosome may (probabilistically) undergo the phases/operators of Crossover and/or Mutation. Crossover probabilities in the literature are typically reported between 0.8 and 0.95 (Song, 2015, p. 190). Based on the principles of Darwinian Theory of Evolution, the best candidate solutions have more likelihood of surviving in their "sphere of life" and therefore may live long-enough to mate with other "better" chromosomes to generate offspring with better traits to survive longer. The underlying idea of improving the prediction rate of a time-series stock sequence is also based on the notion that the best parametric combinations based on previous stock values will survive and move ahead to cross with similar solutions and create even better solutions.

Unlike the Genetic Algorithm and Genetic Programming work to date, the proposed method aimed at a more direct approach to solving stock value prediction by using genomes whose strings of numbers represent symbols. These strings of symbols can further be expressed as equations, grammars or local mappings. The genome is then mapped to a binary tree which can be walked-through to evaluate the underlying equation.

Contrary to the GP and GEP techniques in the literature, the proposed technique builds a model predicting the future function values based on the previous history of stock values. The technique calculates closed-form equations of the most optimal dataset from real-world stock market data. The closed-form equations thus obtained are the comparative fitness-functions for the GEP run. The underlying principle is extremely powerful as, unlike conventional GAs, it does not map to hard values but to symbols that iteratively combine to give equations that are nearest to the closed-loop fitness equations derived from the existing training data. In order to develop closed-form equation, stock value variations were treated in two different modes, namely medium-term and short-term modes. A range of selection operators are reported in the literature and when employed, present various pros and cons. Some of these operators are inherently biased towards better solutions and therefore, prevent the overall genetic process to bring the intended diversity in the solutions. These solutions are further discussed in the following sections.

4.2 Roulette-Wheel Selection

As the name suggests, this operator selects the candidate solutions based on their fitness share mapped to a roulette wheel. Therefore, the best performing candidate solutions (chromosomes) get higher chances of selection to the next generation provided that they are selected probabilistically. In this way, the "roulette-wheel" exhibits large slices for candidates containing high fitness levels. As shown in Figure to select a chromosome, the wheel is spun in the number of times the total candidates appear in the generation. Since the best solutions have the largest share on the roulette wheel, their chances of being numerously selected and then subsequently crossed (via the Crossover operator) are high.

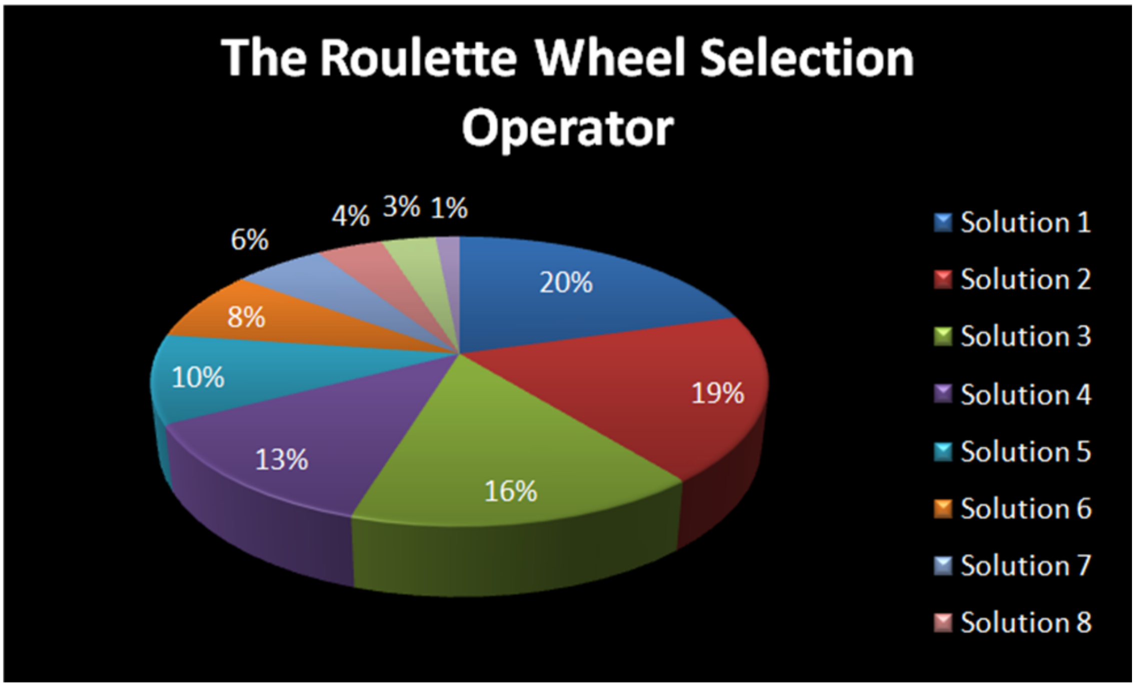


Figure 5: The next generation Selection process based on Roulette-Wheel algorithm.

When the wheel is spun during each selection process, a random number or pointer is generated returning a decimal value between 0 and 1. Each spun or random pointer selection process therefore has a higher probability or likelihood to stop at fitter individuals as they comprise of a larger portion of the roulette wheel. The algorithm is detailed further in Table 2 below.

```
for all members of population
   sum += fitness of this individual
 end for
for all members of population
   probability = sum of probabilities + (fitness / sum)
   sum of probabilities += probability
 end for
loop until new population is full
   number = Random between 0 and 1
for all members of population
   if number > probability but less than next probability
   then you have been selected
 end for
 end
   create offspring
end loop
```

Table 2: A roulette-wheel selection algorithm for next generation selection

Despite its capability to find a healthy genetic generation with numerous high-fitness candidates, the RWS presents a typical drawback of RWS technique. If the fitness of a candidate in the generation differed substantially from the remaining candidates, the overall diversity of the generation decreases phenomenally. That is, at the end of the genetic run, a group of very similar candidates are found.

For example, if the best performing candidate's fitness generated to be 80% of the entire wheel length, the selection probability of other algorithms would become very unlikely and thereby result in the elimination of other good solutions. This phenomenon may result in limiting the genetic process to an unchanging fitness maximum as the offspring in the subsequent generations are very similar to their parents.

A few alternatives catering to the RWS algorithm's limitation are reported in the literature and can be used to address this problem. Such alternatives are $\mu + \lambda$, Tournament, Boltzmann, Rank-Based Selection (RBS), Stochastic Universal Sampling (SUS), Steady State Selection (SSS), Rank and Scaling, and Sharing (Gen and Cheng 2000, p. 9); for this research, RBS, RWS and elitism-based selection algorithms are used.

4.3 Rank based Selection (RBS)

The RBS algorithm is different from the RWS in the sense that it initially assigns a rank to each candidate solution in the previously termed roulette wheel. In this way, the poorest performing candidates get a fitness of 1 with the ranking getting higher with improving candidate fitness. It can be noticed from Figure 6 (a) and 6 (b) that lesser-fit candidate have a fairer chance of being selected in the next generation. For the individual with least fitness, there is still a 2% compared to 1% in RWS whereas the best fitness candidate's selection chance has reduced from 27% to 18% thereby reducing its monopolising tendencies for the next generation selection process.

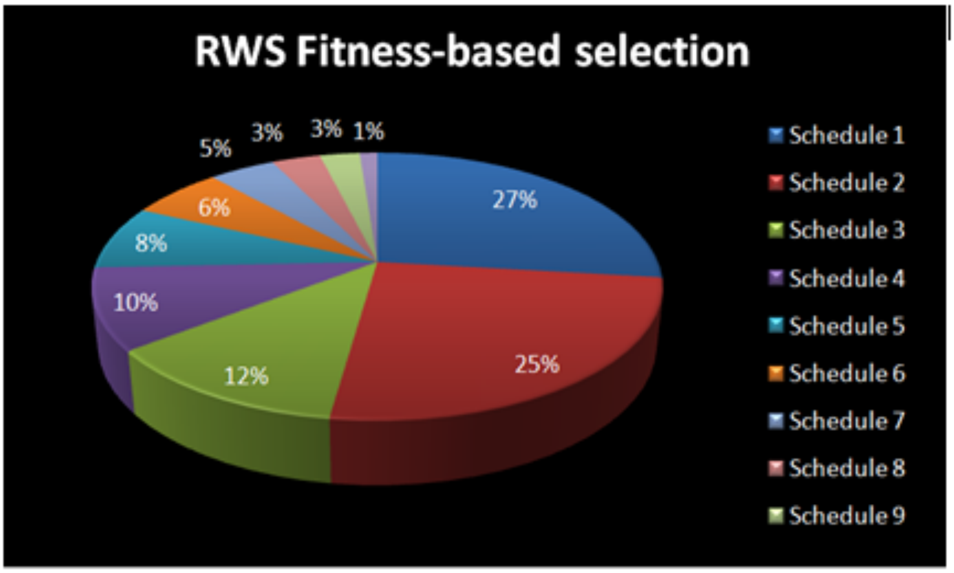


Figure 6 (a): RWS Fitness-based selection.

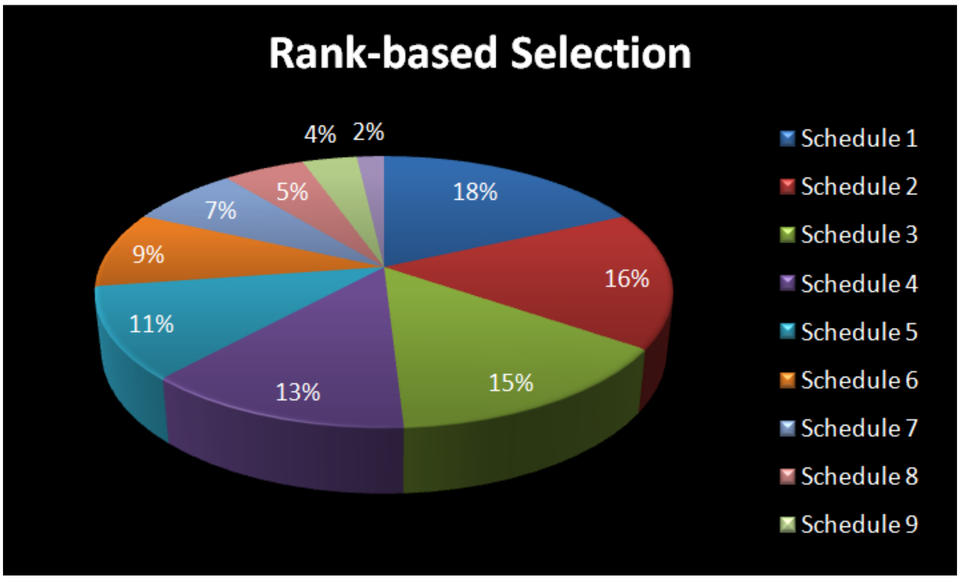


Figure 6 (b): Rank-based Fitness selection.

Yet the RWS-upgraded rank-based selection algorithm still posed other challenges during the convergence phase of the run. Since the least-fit solutions also had comparably better chance of getting selected, the algorithm spent a substantially higher time to reach a certain fitness criteria. Off-course in the end the solutions were more diversified and hence the user had a broader choice to select the best of the best from the final generation. The fitness criteria for the proposed stock exchange technique are discussed later on in this thesis.

Although the SSS technique was not used in this work, it is worth being mentioned as a solution to the shortcomings of the RWS and RBS. The RWS and RBS problem of a computationally cost-intensive convergence time was countered in the SSS via the principle that just a large part of a generation's candidates (AI models) survive to the next generation and the remaining were eliminated.

Nonetheless, SSS technique still bears a high possibility of losing the fittest candidate chromosome during the Selection process, which will downgrade the overall fitness of the entire genetic run. That is, despite having higher chances of improving the fitness of scheduling solutions in subsequent runs, the overall running time of the algorithm is still envisaged to be longer. Ideally, this technique seems to be a trade-off between running time and fitness achieved. The technique therefore does not seem to offer substantial improvements in the proposed work of this research and hence its possible exploitation is considered for further research. In fact, an improved generation fitness quality without overly reducing the diversity of genetic population can be done via a well-known method called Elitism.

4.4 Elitism-based selection

Elitism is an established evolutionary principle in GA that retains the fittest solution from a generation i to the next population $\mathrm{i} + 1$. The methodology established its robustness during the selection phase and the subsequent convergence time of a GA (Vasconcelos et al. 2001)

Elitism-based selection is not a mandatory action during a genetic run. It is a way to retain the best-performing chromosome in the generation to prevent a probabilistic loss of good solutions due to crossover and mutation. The focus of this research was to improve stock market prediction via various optimisation methodologies. Hence, Elitism was explored and its effect on the results was reported which forms the very basis of improvement on others' work. The technique used in this methodology is being dynamically improved on certain genetic population traits. For instance, if the population contains negative outliers (low fitness solutions) that may probabilistically cross with the best (Elite) chromosome, then both the offspring and the Elite solution may be retained while eliminating the low-fitness parent outlier.

The selection method aims to keep a single or a set of solutions in the generation while moving from one to another. Elite, Rank and Roulette wheel selections are three selection methods with the justification of Elite already explained above.

4.5 The Crossover Operator

Once a candidate is selected to be moved into the next generation, it is probabilistically crossed with a selected candidate. This step is known as the Crossover process and ensures that the good (or some bad) traits of high fitness individuals are shared by child candidates in the next generation. Figure 7 shows a single-point crossover which was used in performing chromosomal crossover for the encoded stock prediction algorithm solutions. A number of crossover techniques have been reported in the literature including single-point, two-point, uniform, and arithmetic crossover. All these techniques have their own merits and demerits though the proposed approach uses the common single-point crossover as shown in Figure 7. The single-point-crossover mechanism swaps the chromosomal contents of two parent chromosomes from generation 1 at its crossover point and swaps them from each other to generate two child chromosomes that are to be included in the next generation. The subsequent fitness of these child chromosomes is evaluated in the next generation where the children with low fitness are probabilistically eliminated based on an RWS or RBS operator. Typical Crossover probabilities range from 80 – 95% in the majority of literature.

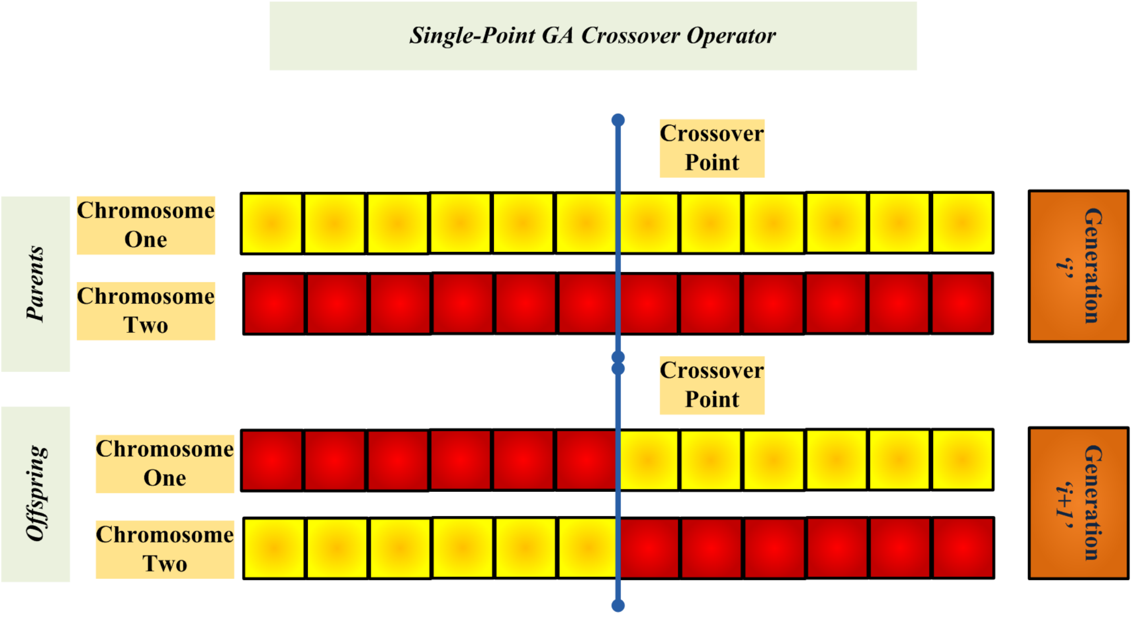


Figure 7: A single-point GA crossover process between two parent chromosomes generating a subsequent next generation.

Yet, despite crossing parent candidates in a hope to get good child candidates, there still remains a substantial risk of the entire genetic run getting stuck and converging to a constant, non-improving fitness of generation only because children are very similar to each other. This problem is addressed in the GA via a so-called "mutation stage" operation, which randomly changes the chromosomal details of a small part of these chromosomes.

4.6 Mutation Operator to Induce Search Space Diversification

In the genetic selection process, a mutation operator is used to slightly alter the structure of a chromosome by a very low probability. This must be done with a low probability to prevent the GA from converting into a random search algorithm. Figure 8 presents a genetic mutation process where the bits of a chromosome are flipped from 0 to 1 or vice versa.

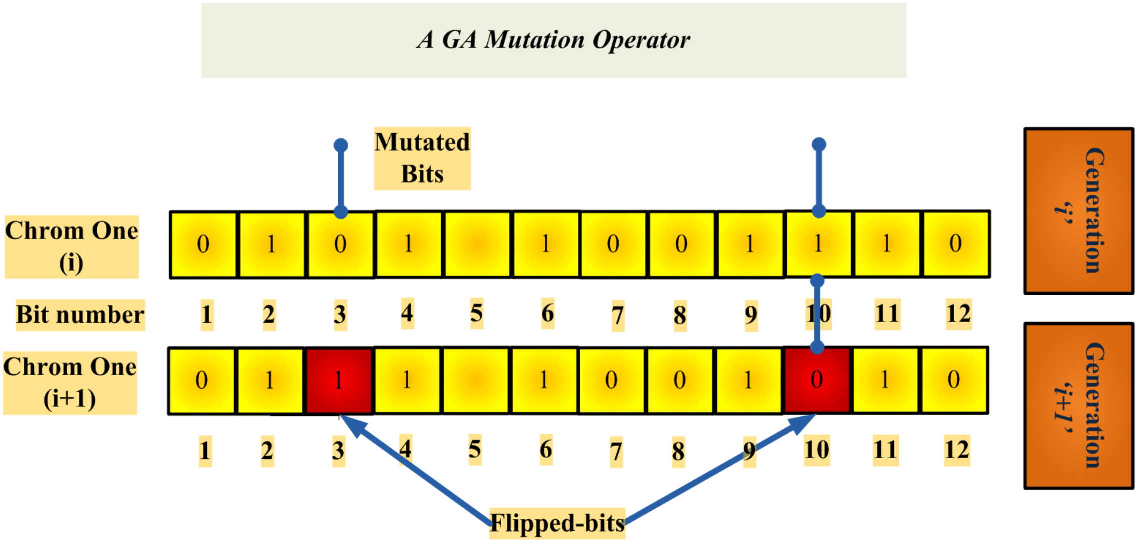


Figure 8: A GA mutation operator used to induce diversity in subsequent populations via bit flipping.

It must be noted that mutation must be performed with a fairly low probability of less-than $0.5 - 1\%$ percent. A high mutation probability changes a mutation process to a randomised process instead of a genetic run.

The convergence of the GA/GEP implementation criteria was based over a rate of fitness change measurement criteria where if the fitness of the best solution did not change for 10+ genetic runs, a convergence was assumed and the run terminated.

The genetic run iterated on the Elitist approach which aimed at retaining the best stock prediction AI solution over the subsequent runs. The outcome of the methodology is critically analysed in the results Chapter 6 (Vasconcelos *et al.* 2001).

4.7 The optimisation problem

The main stock prediction problem can be described as follows: Provided current stock value(s) and the trading volume, the task is to predict the next time-instance stock market value. The hypothesis assumed here is that the stock values during a trading day change gradually and display an organised relationship between current and future stock values. This relationship can be described as an algebraic function whose derivation is the objective of this research. The problem particularly focuses on the closing stock value in conjunction with the volume traded, which can be elaborated as shown in equation 9 below:

$$\hat{\sigma} = \mathrm{F}(\vartheta, \mathbb{C}) \tag{9}$$

In Equation (9), $\mathbb{C}$ is a generic coefficient dependent upon the surrounding circumstances of a stock market. In polynomial terminology, (9) can be extended to the equation given below:

$$\hat{\sigma} = \mathbb{C}_0 + \mathbb{C}_1\vartheta_1 + \mathbb{C}_2\vartheta_2 + \mathbb{C}_3\vartheta_3 + \cdots + \mathbb{C}_n\vartheta_n \tag{10}$$

Based on the equation given in Equation (10), an optimal solution becomes the most optimal equation with its weight constant $\mathbb{C}_n$ calibrated via a gene expression run. The proposed approach in this paper therefore utilises GP as a medium to search a solution space for the most optimal computer programme where various models are represented in a tree-like hierarchy where each node represents a set of programmatic expressions that are evolved during a genetic run.

4.8 Evolutionary Gene Expression Programming

Gene Expression Programming (GEP) is regarded as an extension of the GA where the chromosomes take form of programming-based solutions instead of decoded genetic phenotypes which contain fixed-length binary chromosomal strings. Similar to conventional GA, GEP has the capability to generate computer-based models based on natural genetic evolution. Conventional GA creates a sequence of numbers that represent solutions, whereas GEP solutions are computer programmes that follow a tree hierarchy where each node represents a programme expressed in a functional programming language such as Matlab, C++/C# or Java. Therefore, in GEP, the algorithm optimises a family of computer programmes based upon the contextual fitness scenario. Similar to the GA, GEP contains linear, fixed-length chromosomes and hierarchical structures similar to genetic programming trees. Each of the GEP solutions is represented via a tree-node or chromosome. Each generation in a GEP contains a number of these chromosomes that go through a process of recombination to induce the so-called diversity in a genetic population technically regarded as the solution search space. In GEP, including recombination there are eight operators out of which this research utilises the crossover, mutation and selection operators to induce the diversity in the running generation.

Each programme therefore evolves a structured tree where the underlying model is then converted into an equation y while traversing from left to right as shown in Figure 9.

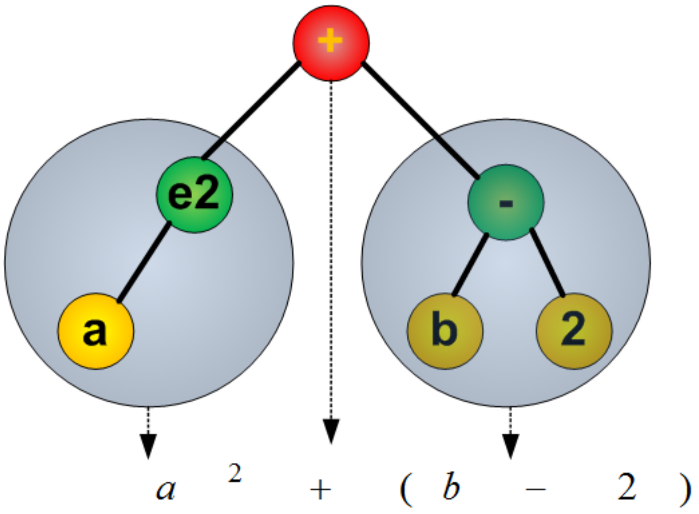


Figure 9: Representation of tree-hierarchy chromosomal representation of GEP.

A wide range of GEP approaches have been reported in the literature including linear and graph-based approaches. The technique proposed in this paper is focused primarily on the linear approach. In the linear case of GP, programmes are represented as sequence of programmes that are encoded or decoded as nonlinear entities. A linear interpretation system is substantially faster than the tree-based system. The algorithm uses fixed-character-solution-strings to encode solutions to the problems. These strings are eventually annotated as parse trees bearing different sizes and shapes, and are called GEP expression trees (GEP-ETs). During a GEP genetic run, the GEP results in the evolution of additional complex programmes made-up of numerous, simpler sub-programmes. As a typical GEP gene contains fixed-length character strings, these can be drawn from any element from a function-set like $\{\cos, \mathrm{sine}, +, -, \mathrm{x}, \backslash\}$ and a terminal set like $\{-1, 3.14, \mathrm{a}, \mathrm{b}, \mathrm{c}\}$. A typical gene thus generated is shown in Figure 9.

The gene or the chromosome in a GEP language therefore comprises of a head and a tail where the head include symbols that represent programming functions and terminals. The tail however only contains the terminals. For any problem therefore, the head length is chosen and the tail length is worked-upon by the algorithm as a function of h and the total number of arguments ζ of the function with more arguments commonly known as the maximum arity (Peng et al., 2014). The terminal can therefore be evaluated as:

$$\tau = \mathrm{h}(\zeta - 1) + 1 \qquad (11)$$

For a gene with a set of functions $\mathcal{F} = \{\mathfrak{R}, *, /, -, +\}$ with a terminal set $\mathrm{T} = \{\mathrm{x}, \mathrm{y}\}$; $\zeta = 2$; if we choose the head length $\mathrm{h} = 25$ then $\tau = 25(2 - 1) + 1 = 26$ and total gene length $\Psi = 25 + 26 = 51$.

Based on the capability of GEP to evolve and improve programmatic expressions, the rationale behind the usage of GEP for stock prediction can be elaborated into three points:

1. The chromosomes representing solutions are simple entities which can be replicated (i.e. recombined, transposed, mutated) to generate probabilistically more

efficient solutions. This approach offers a promising potential to predict forecasting based on time-series data based on equations trained on existing training data.

2. The expression trees (ET) formed in GEP express chromosomes, which then change based on crossover. This "tree-walking" operation reproduces genetically evolving equations which can be compared to original curve-fitting equations to estimate fitness values of each chromosome. Again, contrary to standard brute-force search mechanisms, this ability can iteratively estimate iterative fitness of all the chromosomes in a generation. The trait makes it possible to generate multiple best solutions that can then be used to predict a forecast value based upon previous data.

3. According to Ferreira (2001), the GEP algorithm surpasses GP by more than four orders of magnitude. The performance improvement of GEP is of crucial importance particularly during the processing of massive real-world datasets.

4.9 Shortcomings of evolutionary hybrid methodologies in time-series prediction

Despite having a large body of literature covering GA, the reliability of GA in future time series predictions is still looked upon with doubt. GA can often present over-fitted results based on existing temporal data. The literature does show an increased application of evolutionary computing techniques to hedge funds (Yan & Clack, 2011). However, based on safety issues, the models are extensively evaluated against back-tests before they are actually used in actual production cases. This often leads to months of testing before a model is actually allowed to run on real-life cases. Model reliability in similar situations is improved by separating the sampling universes while keeping the confirmation back-tests separate. The strategy is to make a GA from data till time t and then subsequently testing it till $\mathrm{t} + \mathrm{N}$ before actually trusting the underlying model.

As discussed earlier, despite its own strength, the three biggest weakness of the GEP/GP are over-fitting, data snooping bias and black-box operation.

4.9.1 Data over-fitting

Similar to conventional ANNs, data over-fitting in GEP occurs when possible crossovers between a fitter and weaker chromosome take place resulting in gene parameters generating weaker solutions in the subsequent generations. This can be controlled via a number of ways. The simplest form is to monitor a genetic run for its learning error and stop the run as soon as a consecutive rise in error rate is noted for a number of runs.

However, data over-fitting can be avoided in three ways: 1) Providing an indirect bias towards simplicity of solutions or setting a penalty against complex solutions, 2) Restricting the number of models considered in a run, and/or 3) Using a validation dataset. The lateral validation dataset cannot be an ideal case for a time-series case, as a validation dataset may belong to a different time with variable circumstances which do not reflect the overall company stock profile (Chan, 2011).

4.9.2 Data snooping

As noted by Neuhierl and Schlusche (2011), data snooping, in which a researcher develops various inferences from data without any pre-plans, is not a drawback in itself. The main challenge that data snooping presents is the likelihood of high researcher bias when making conclusions. The problem of data snooping in GEP can be avoided by well-organised mutation operators/functions. However, setting a specific mutation rate to avoid snooping is another challenge where a very high mutation rate may reduce a GEP run into a randomised search. On the other hand, if the mutation probability is too low, the function does not serve its purpose of diversifying the genetic population.

4.9.3 Black-box operation

An average evolutionary run acts as a black-box where a continual hill-climbing process finally terminates into an optimised fitness function. A GEP run cannot be changed apart from its specialised selection operators (e.g. crossover and mutation) in order to prevent the algorithm from either turning into a random search algorithm or getting stuck into a local minima, which could adversely affect the effectiveness of the developed system. To avoid the eventuality, the crossover and mutation operators are used in this research.

In this research, the GEP-FAMR approach was employed instead of Neural and Fuzzy approaches to help address the problems of over-fitting, algorithmic black-boxing, and data-snooping issues in GP and GEP algorithms. The GEP-FAMR approach provides an organised runs monitoring system, in which validation tests are also conducted and thus helps address over-fitting and data snooping problems. To train data, stock market conditions must be factored; nonetheless, depending on the conditions, the frequency of training ranges between once a week and in two months.

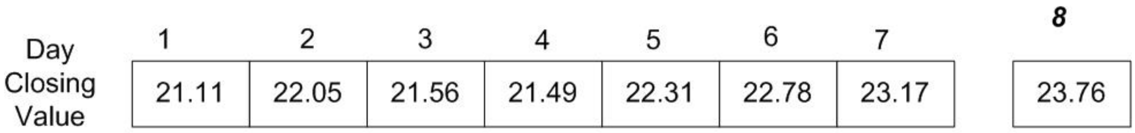
4.10 Proposed GEP for Time Series Prediction

Based on the description of the problem provided in the above sections, the adaptation of GP/GEP to the proposed algorithm is given below.

4.10.1 Fitness Evaluation

The most basic fitness rule in this methodology is taken to be the rate of change of stock prices over a certain historical data pattern (medium or short-term).

In the proposed GEP, two algebraic functions representing linear and exponential equations are used to derive linearly and exponentially regressive equations. These equations represent training data as a line on a 2-D plain that are used to represent a relationship between the current and future stock value (RegCal 2015).



Day	1	2	3	4	5	6	7	***8***
Closing Value	21.11	22.05	21.56	21.49	22.31	22.78	23.17	23.76

Figure 10: Representation of 7-day time-series data.

For instance, for data shown in Figure10, mapping the days on ‘x’ and Day Closing Value on ‘y’, the equation for the curve can be obtained by the method of least square regression where the best fit line is computed via the equation given below:

$$y = mx + b \tag{12}$$

Equation (12) satisfies the condition that the sum of the squared vertical distance between two points and the line is minimised. An exponential regression curve of this curve $y = ax^c$ or the logarithmic regression curve is then obtained via the equation (13):

$$y = a + c\ln(x) \tag{13}$$

The slope of the regression curve is thus given by the formula:

$$m = {\sum xy - n(\bar{x})(\bar{y})}\Big/{\sum x^2 - n\overline{(x)}^2} \tag{14}$$

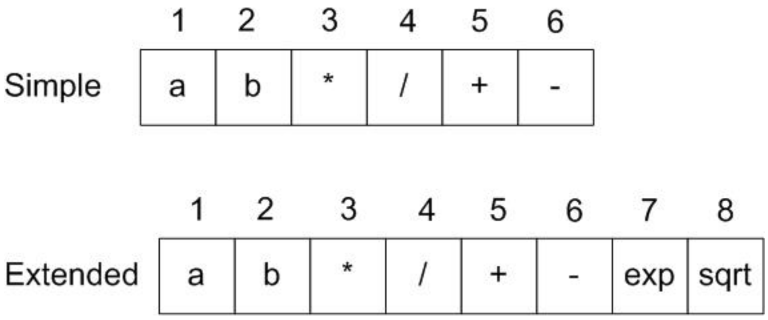
Based on the equation (14), a correlation coefficient showing how close the line fits can be computed via the following equation:

$$\sum xy - n(\bar{x})(\bar{y})\Big/\sqrt{(\sum x^2 - n\overline{(x)}^2)(\sum y^2 - n\overline{(y)}^2)} \tag{15}$$

The correlation coefficient value ranges from 1 to -1 where, if the correlation is close to zero, it means the data does not exhibit a linear relation. Similarly, equations for exponential, power and logarithmic regression curves can be transformed in their respective linear forms. For instance, equation $y = ax^c$ can be linearly transformed by taking the natural logarithm of both the sides.

$$\ln(\mathrm{y}) = \ln(\mathrm{a}) + \ln(\mathrm{c})\,\mathrm{x} \tag{16}$$

The simple (linear) and extended (exponential) function sets used in this paper basically mean linear and exponential function sets. The exponential sets were eventually transformed into linear equations via natural algorithm. The function sets used in the methodology were as shown in Figure 11 below.



	1	2	3	4	5	6
Simple	a	b	*	/	+	-

	1	2	3	4	5	6	7	8
Extended	a	b	*	/	+	-	exp	sqrt

Figure 11: Simple and extended function sets used in the GEP

Hence, for data shown in Figure 10 above, the two linear and exponential equations can be obtained as follows,

$Linear: y = ax + b$

$$y = 0.2996x + 20.8686 \tag{17}$$

$Correlation: 0.8741$

$Exponential: y = cd^{x}$

$$y = 20.8951(1.0136^{x}) \tag{18}$$

$Correlation: 0.8731$

GEP attempts to derive the best set of symbols for a genome with the highest fitness. The fitness function in the run aims to generate the highest number of results that occur while traversing the gene tree and generating values that are closest to the correlation generated by the equations shown in (17) and (18). Hence, the fitness function in this case is the summation of correlation (corr) difference and subtraction it with a large number as follows:

$$fitness = \sum_{i=1}^{100} abs[(corr_{(n)} + corr_{(n+1)}) - (corr_{(obs-n)} + corr_{(obs-(n+1))})] \tag{19}$$

Therefore the minimum the difference of correlation between a gene-built equation and the equations (17) and (18) the higher the fitness of the respective solution will be.

4.10.2 Generation of initial population

The methodology uses an ensemble of programming trees that follow the recursive construction processing from root to leaf via the guidelines described by Potvin *et al.* (2004).

- The tree root selection is made from Boolean functions and operators.
- After the root has been selected, its descendants (leaves) can be selected from Boolean constants and functions and Boolean or relational operators.
- If a relational operator is selected, its descendants are selected from either real functions or terminals.

4.11 Selection of next generation chromosomes

The selection process in this work adopts three main methods, namely Elistism, rank-based selection and the roulette-wheel sampling .The Elitism-based methodology was used to select at-least on best individual with the highest fitness value in the subsequent generations. The main advantage presented by the Elitism-based method is that it helps minimise the chances of best programmes recombining with weaker candidate. However, while preventing this recombination leads to highly fit candidates, the risk of best solutions loss during the genetic runs, based on the underlying mutation probability that was set at 0.01 (1%), exists. Accordingly, to address these issues the proposed FAMR used

In Rank-based sampling, during any genetic run, all the programmes (chromosomes/solutions) are ranked based on their best-to-worst raw fitness values. For each chromosome, a new fitness value is then assigned.

In roulette-wheel (RW) selection, each programme is assigned a slice of the roulette wheel based upon its fitness. Therefore, the individuals with highest fitness have more chances of getting selected in subsequent generations.

4.11.1 Composition of evolutionary operators

The most common pitfall of conventional GA is its complete randomness during the mutation process, which is equally likely to mutate (and hence destroy) the fittest chromosome as well as low-fitness ones. Additionally, low fitness chromosomes are less likely to generate good chromosomes due to the lack of essential building blocks to create better chromosomes (solutions). Hence, a balance must be kept where low-fitness chromosomes are to be mutated to get better solutions, while making it less likely for high-fitness solutions to get mutated in order to continue with the regular fitness improvement under normal crossover probability conditions (Jiang et al., 2008, Lei et al., 2007, Limin et al., 2008, Bautu et al., 2007). The techniques primarily focus on reducing mutation rate for entire generations or individual, low fitness chromosomes. This method, though, improves the survival of high fitness candidates, ultimately increases the chances of low-fitness solution survival which lead to low diversity duration in the later stages of the genetic run. Therefore, the approaches still present a significant risk of getting the genetic process stuck into local minima based on the fact that a large number of very low fitness functions with an extremely high mutation probability are likely to transform into a completely random process.

4.11.2 Chromosomal composition as encoded real-numbered constants

Automatically Defined Functions (ADFs) were originally introduced by Koza (1994) which derived their logic from the manner in which human programmers as reusable components. Based on the real-numbered time-series nature of the algorithms, random numerical constants were incorporated in the Automatically Defined Functions (ADFs).
The proposed approach used two ADF-derived function sets deriving from the following sets:

- Simple: (+, -, *, /),
- Extended: (+, -, *, /,exp, sqrt)

4.11.3 Mutation via fractionally dynamic fitness-based adaptation

In conventional GEP, a fixed probability is assigned which provides equal opportunity for each chromosome to be selected. However, this provides equal chances for best chromosomes to be mutated and hence lose fitness as well. Hence, in the proposed methodology a dynamic fitness-based adaptation was induced in the mutation process. The proposed approaches report on fitness-based mutation rate adjustment which only permits mutation rate adjustment for a set of top 'n%' fitness chromosomes while keeping the mutation rate constant for the rest Table 3.

$\mathbb{R} = \{x/0 \leq x \leq 1\}$

Sort generation $G_i \in \mathbb{G}$ as follows (where $1 \leq \mathbb{G} < Total\ generations$)

Let $C = (x_i)_{N=1}^{n}$ be a sequence of chromosomes and their fitness satisfying:

$\forall\ 1 \leq \mathbb{G} < Total\ generation|\ \{i \in N|x_i = x_k\}| = |\{i \in N|y_i = y_k\}$

$Select\ \mathbb{R}_X = max(x\ such\ that\ \#\{s \in \mathbb{R}\ |\ s \geq r\} = X)$

$\mathbb{R}^X = \{r \in \mathbb{R}|r \geq \mathbb{R}_X\}$

where 'X' are the top 'n' fittest individuals

Calculate average probability $R_{p(c)}$as follows:

$R_{p(c)} = \left(f_y/f_{max}\right)\times (p_{max} - p_{min})$

Where $f_y \in \mathbb{R}^X$ is an individual whose fitness is to be calculate adaptively and

$f_{max} = \frac{1}{K}\sum_{i=1}^{y_k} y_i$ is the average maximum fitness of the top X fittest chromosomes and

p_{max} and p_{min} are the minimum and maximum probabilities of the entire group G_i

Table 3: Proposed adaptive mutation algorithm for fractional probability adjustment based on top-n fitness members.(FAMR)

4.11.4 Transposition operator

Transposition operator is used in evolutionary algorithm to exchange sections of genes from within a single parent (asexually) or between two parents (sexually). Regardless of the case, in GEP, the inserted sequence can appear anywhere in tree but the insertion can only be performed at the heads of genes. The operator in the proposed approach was used to asexually change single chromosomes as shown in Table 4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Chromosome	/	-	b	/	a	b	B	a	a	*	a	-	/	a	b
Transposed Chromosome	/	-	a	*	a	-	/	a	b	B	/	a	b	b	a

Table 4: Simple transposition operation used to exchange genetic material in GEP function trees.

4.11.5 Recombination

Recombination operation generally involves two parent chromosomes, which are combined via single/multipoint crossover. The resulting children chromosome is deemed syntactically correct if the resultant size is same and the resultant fragments are homologous. The proposed technique used a single-point crossover mechanism for recombination as shown in Table 5. The technique probabilistically selected two chromosomes from a generation and swapped their genetic composition over a single crossover point. The crossover point for the case shown in Table 5 is on the immediate right of the red-coloured cell.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Chromosome 1	/	-	b	/	a	b	b	a	a	*	a	-	/	a	b
Chromosome 2	-	B	b	a	b	*	a	b	/	a	b	B	-	a	c
Chromosome 1	/	-	b	/	a	*	a	b	/	a	b	B	-	a	c
Chromosome 2	-	B	b	a	b	b	b	a	a	*	a	-	/	a	b

Table 5: Sample single-point crossover operation used to diversify GEP function trees.

In conclusion, this chapter discussed a detailed methodology of the evolutionary optimisation algorithm that is used to improve the time-series-based prediction accuracy of the stock price forecasting system. The selection process in this research adopts three main methods that include Elistism, rank-based selection and the roulette-wheel sampling. In Rank-based sampling, all the programmes (chromosomes) are ranked based on their best-to-worst raw fitness values during any genetic run. In RW (roulette-wheel) selection, each programme is assigned a slice of the roulette wheel based on its fitness. Therefore, the individuals with highest fitness have more chances of being selected in the subsequent generations. Lastly, the Elitism-based methodology is used to select at-least on best individual with the highest fitness value in the subsequent generations. This minimises the chances of best programmes recombining with weaker candidate resulting in best solutions getting lost during the genetic runs. However, this strategy presented another challenge in the overall running where the fittest individual was often lost based on the underlying mutation probability.

The next chapter presents the design and analysis of this research

5 Chapter5: System Design.

This chapter presents the design and analysis of this research. Using GA, the research aims to develop software that will help in forecasting financial markets. Firstly, the development methodology of software is provided and various possible system design approaches (top-down, bottom-up & middle-out) explored. In the next section, system implementation via the Object-oriented paradigm is presented, in which a baseline model transformed is provided. In addition, various industrial development standards including DSDM and SCRUM and their possible role in the software developed are also discussed in the section. In the next part, class and system context and data flow diagrams of the research's developed software, employed prediction algorithms and STMgS project are presented. The developed system's programming and implementation are provided in the last section of this chapter. The entire software system was developed in Visual Studio 2012/2013 under .Net Framework 4+.

5.1 System Development Methodology

A system development methodology of a software system is termed as a design framework developed to control the process of software solution/system development (Highsmith, 2013). Since the inception of modern-day computing in the late 20th century, a wide range of software system development techniques have evolved. Fenton and Bieman (2014) note that each of these techniques bears its own benefits and disadvantages, as influenced by involved projects. In practice, one single software development methodology may not be feasible for all projects. For instance, as suggested by Balaji and Murugaiyan (2012), for a time-driven project, with a well-defined submission deadline, a waterfall model is deemed the most feasible. On the other hand, for a project (or research venture) where the final objectives are not well-defined, and the stakeholders are more inclined towards frequent system builds, a more iterative application development model is deemed more suitable (Partsch, 2012).

According to Liu (2013), Structured Analysis (SA) is a conventional systems design and development technique containing a series of phases similar to a waterfall model. These phases comprise of the inception, analysis, design, implementation and maintenance stages. A complete iteration of these stages is commonly regarded as the systems development lifecycle (SDLC) (Shelly and Rosenblatt 2010, p. 22). However, as discussed earlier, in modern, time-driven and complex software systems that require routine requirement changes and product evolution, the use of a full SDLC model may not be an efficient solution.

Alternatively, agile methods were recently launched with the advent of the object-oriented design and development paradigm (Ambler & Lines, 2012). As noted by Gorans and Kruchten (2014), these methods/methodologies are considered to be the newest class of system development methodologies which use a spiral (or iterative) model of development. The model represents a sequence of iterations or revisions, which may occur because of changes in the requirements of clients or focus on a more result-concerted approach, which necessitates modifications in the functional specifications of a project. Generally, because of this model's nature, each iteration includes sub-phases of planning, risk analysis, engineering and evaluation, which may change continuously during the entire project lifecycle. These repetitive iterations develop a number of prototypes which, based on client-developer discussions, evolve into the final system build (Shelly and Rosenblatt 2010, p. 26). A number of standard approaches in this domain are the RAD, joint application development (JAD) and software prototyping. The core objective of these techniques is to bypass many of the SDLC phases in order to improve the overall software development process. As these methods emerged to substantially reduce the time spent in the analysis phase of the SDLC, the development efforts in these techniques directly jump from planning to the implementation phases thereby leading to an improved and robust requirement analysis phase (Highsmith, 2013). RAD methodology works by compressing the SDLC phases to planning, design and development. The repetitive phases are reduced or eliminated, which enables the realisation of a highly result-oriented system. The elimination not only allows the evaluation of a project's performance within shorter time intervals, but also facilitates faster requirement change integration, which in the case of a Waterfall model would only have been possible at the end of the development and testing phases.

The RAD methodology involves the employment of a prototyping tool that permits an expedited development of the physical software model (Salvendy and Institute of Industrial 2001, p. 105).

The computer aided software engineering (CASE) methodology can also be applied to reduce system development time. However, as opposed to the RAD and JAD techniques, CASE methodology aims to keep the complete SDLC intact. This leads to CASE tools containing phases of analysis, design, project management and maintenance (Gould 2005, pp. 189-190). Three examples of CASE tools in use in professional and industrial software development settings are EXCELERATOR, IEW and TurboAnalyst. CASE tools strengthen SDLC by inducing consistency evaluation between data flow diagrams (DFDs), entity relationship diagrams (ERDs) and the data dictionary (Muller, Norman & Slonim, 2012); as suggested by Encarnacao, Lindner and Schlechtendahl (2012), this is done by checking if proper naming conventions and definition rules are followed for attributes and data structures.

Based on the scope of usage, there are three different categories of CASE tools. “Upper CASE” contains planning, scheduling, DFD generation and data dictionary synchronising tools whereas “Lower CASE” provides support for the lateral part of the SDLC of maintenance and implementation. The “cross lifecycle CASE” tools provide a comprehensive support from planning to support.

The prototyping techniques discussed above reduce development time and costs, and increase user involvement. However, these techniques have disadvantages such as analysis insufficiency, developer misunderstanding of client’s objectives, unnecessary attachment to prototype, excessive time required in prototype development and the underlying development expense. Nonetheless, none of these limitations has any major adverse impact on the software; rather, factors such as client objectives misunderstanding and insufficient analysis involve the developer and customer and can be readily solved by consistent prototype reviews. Therefore, these prototyping techniques will be employed in this research; examples of the techniques used include DFDs and Class Diagrams.

5.2 System Design Strategies

A strategic system design is obtained by breaking down the design objective into a number of smaller design objectives to allow each of them to be assigned specific design problems or satisfy specific requirements. It is the underlying nature of these activities that assist in the decision of the design method.

The following system design strategies can be applied to any system analysis and design task methodology (Olderog and Steffen 1999, p. 3).

5.2.1 Top-down approach

The approach is predominantly goal driven. As discussed before, the technique involves the development of modules or sub-systems first. The designer starts building the system by typically developing a hierarchical layout of the framework defining the links between the system modules, modules or functions. The elements at one level are generally developed before starting development at the subsequent lower level (Davis & Yen 1999, p. 518).

5.2.2 Bottom-up approach

This technique is data driven as the work starts at the lowest level first. Once the modules at the lower level are complete, the development moves to the next higher level. This approach is suitable for projects with massive database sizes (Davis & Yen 1999, p. 518). This research uses the bottom-up approach. The approach is suitable because object-oriented programming, whose development begins from the lowest levels to the highest ones, has been employed for software development.

5.2.3 Middle-out approach

The strategy starts developing a system at the middle of its hierarchy. For instance, for a student file system, the system first implements a customer file in an ecommerce system. Eventually, lower-level files to hold order-specific data and higher-level files for customer shipping details are designed (Davis & Yen 1999, p. 518).

However, efficient system design strategies do involve a pre-assessment of the type of real-world system involved. Michael Porter's Five Forces Model is generally used to assess the competitive structure of an industry (Porter, 1980, p. 4). Porter's Five Forces assess the power organisations hold within their industry of operation. The forces include threat of product substitutes and new entrants, supplier and consumer bargaining powers, and industry rivalry (Huggins & Izushi, 2011). When determined, the strength in each force helps organisations formulate the best strategies against the involved industry player.

A synergistic system design strategy involves the verification and validation of more than one aspect of the system's design. The underlying system design model is characterised by its structural and behavioural aspects. The synergistic strategy therefore integrates both the quantitative and qualitative techniques for system evaluation (Debbabi et al., 2010, p. 96). The synergistic design strategy of a system can further be elaborated as the measure to which a system can attain better operation by its modules being specific to each other.

Based upon the above-mentioned design principles, system design strategies can be divided into the following categories (Davis & Yen 1999, p. 583):

- Step-wise, evolutionary design: Start with smaller modules and incrementally add better or newer sub-systems.
- Incremental design: Attain a system design by incrementally integrating developed modules in a sequential manner.
- Goal-driven methodology: The system is openly developed with an objective to achieve a specific goal. The goal can be attained by means of any system development techniques.

For instance, in the case of software systems development, design patterns are used in designing individual classes. Various types of these patterns can be used in conjunction with other patterns, like structural and behaviour, to define the design and conduct of the classes via any of the above-mentioned system design strategies.

Based on the nature of this project the software development process in this work adopted a RAD-based iterative development approach. The approach facilitated speedy changes to the overall programming paradigm based upon reviewer feedback and also ensured defects were addressed in proper time. The design details of this approach are further elaborated in the next section.

5.3 System Implementation via the Object-Oriented Paradigm

The object oriented design and analysis (OOAD) principles were used via the RAD principles elaborated above to achieve the required application development speed. The underlying OOAD software design is paramount in providing an overall technical understanding of the system both from the developer as well as the stakeholder perspective. The software development lifecycle (SDLC) of a project is divided into six main phases under the Waterfall Model as follows. In Figure 12, various stages of RAD paradigm were induced by turning the seemingly sequential model into an iterative paradigm at the design and implementation stages. The system testing phase was based on established AI regression evaluation methodologies including the Jack-Knife and k-Fold testing. The maintenance stage was primarily based upon developing the capability of the software to be extensible to any future updates due to extending research.

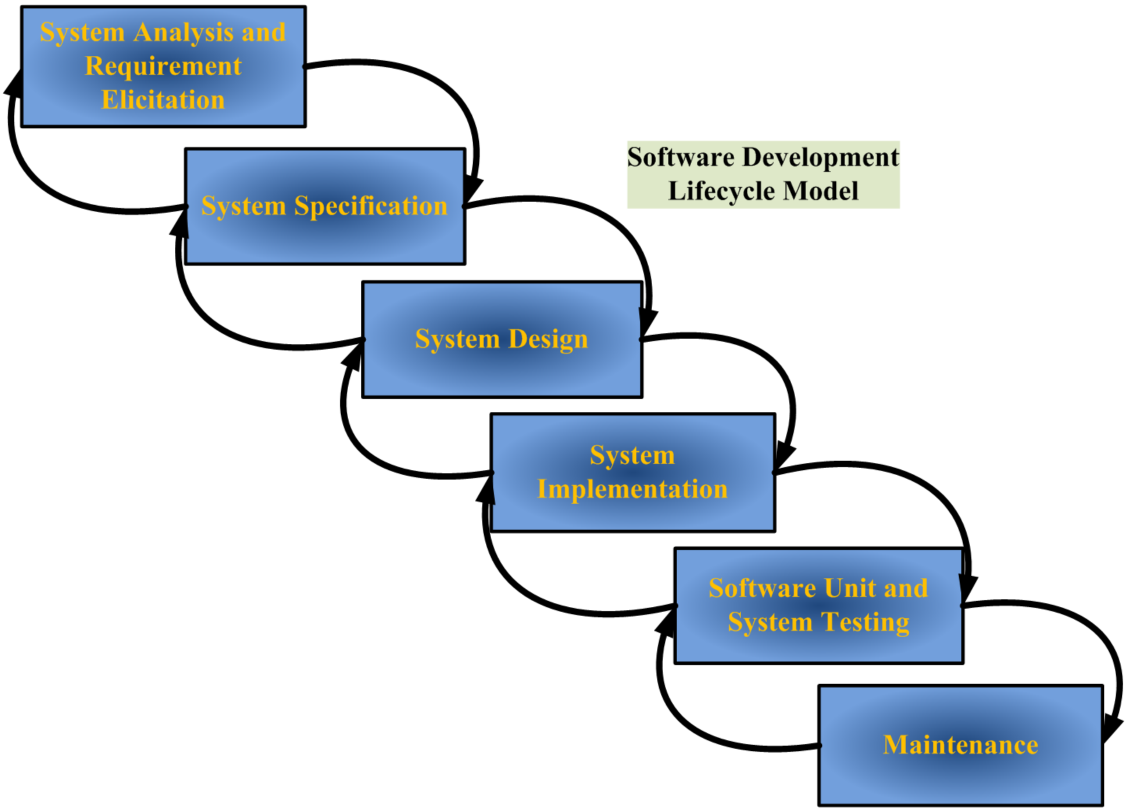

System Analysis and Requirement Elicitation
System Specification
Software Development Lifecycle Model
System Design
System Implementation
Software Unit and System Testing
Maintenance

Figure 12: The baseline model which was transformed into a RAD model by inducing repetitive development during the design and implementation stages of the SDLC.

5.3.1 Class Diagrams

Figure 13 and 14 below are the class diagrams of the evolutionary AI software developed as part of the system and the ANN prediction algorithm including the testing/validation techniques, respectively.

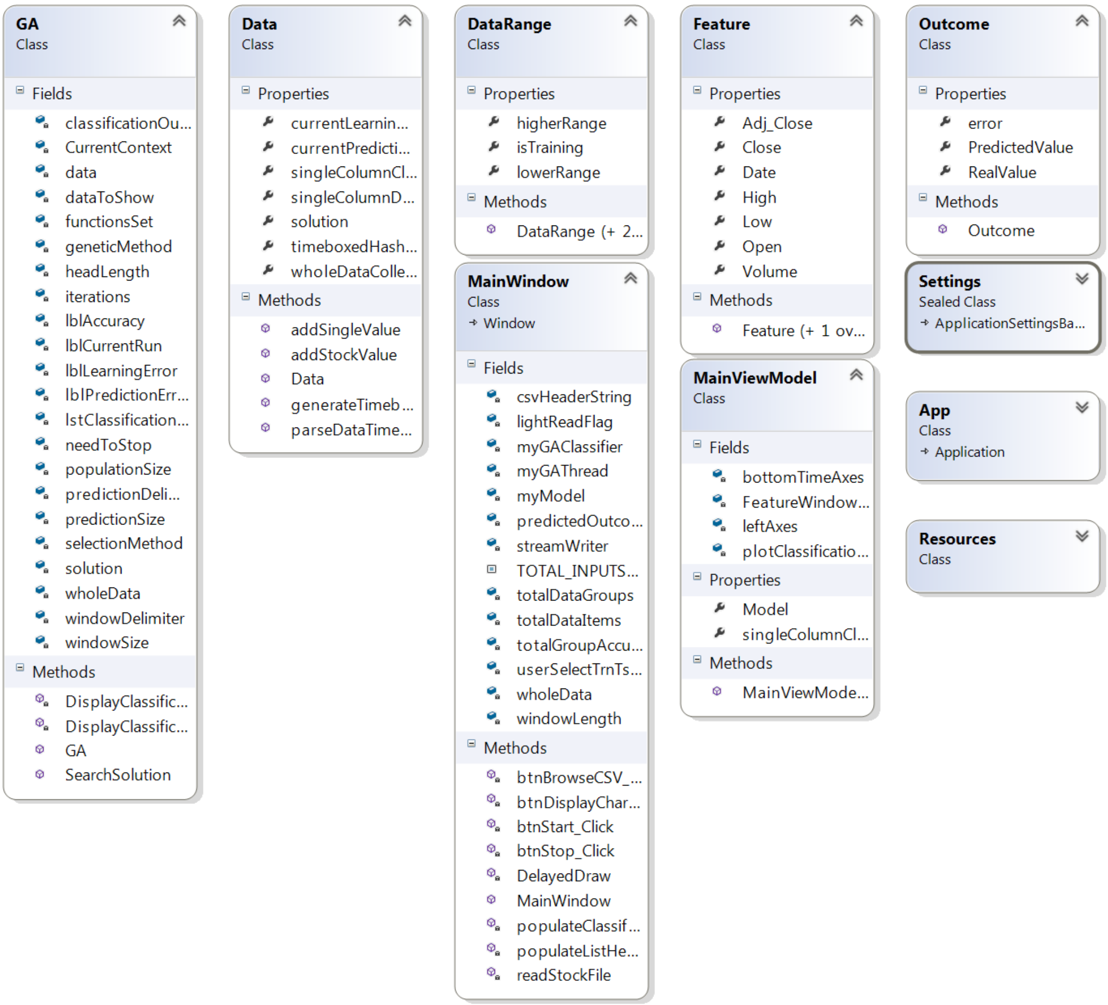


Figure 5: A class diagram of the evolutionary AI software developed as part of the system

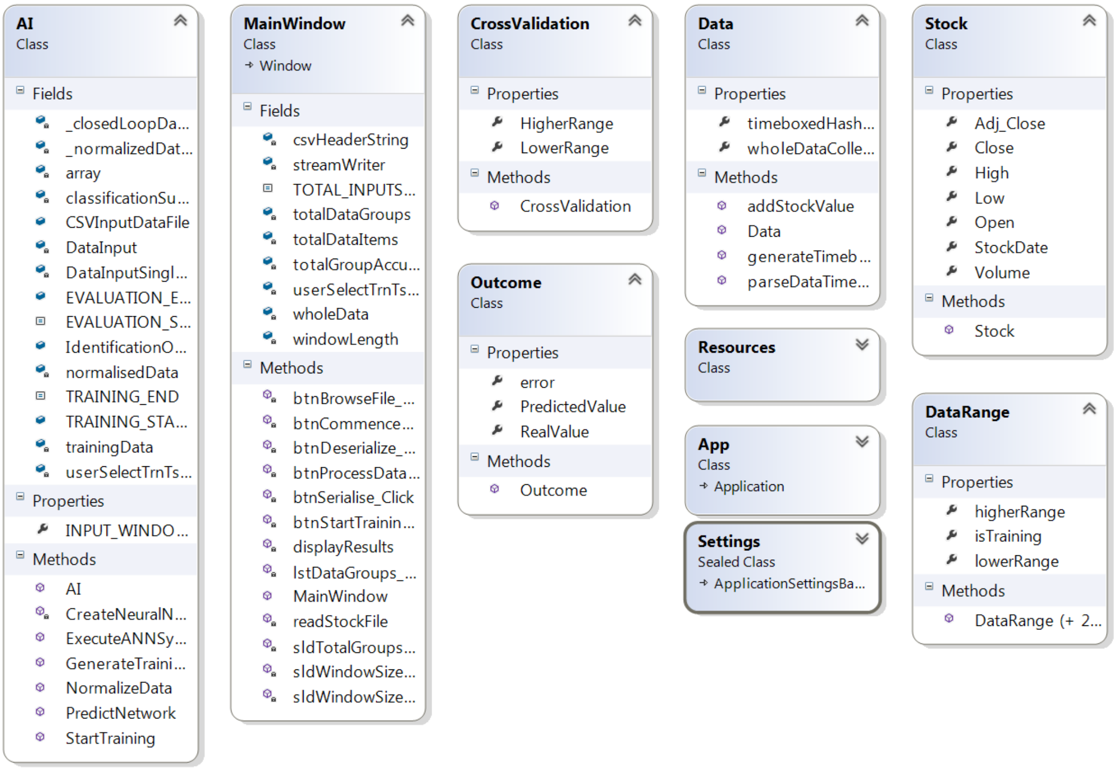


Figure 14: A class diagram of the ANN prediction algorithm including the testing/validation

5.3.2 System Context Diagram (SCD)

As noted by Weilkiens et al. (2015), a System Context Diagram provides a first-level view of the overall system. The diagram shows the flow of information or data between various processes and is the first data flow diagram (DFDs) in a software development model. SCDs do not have data stores are not present generally and are explained at the later, more refined representations. The SCD of the detailed system is shown in Figure 15.

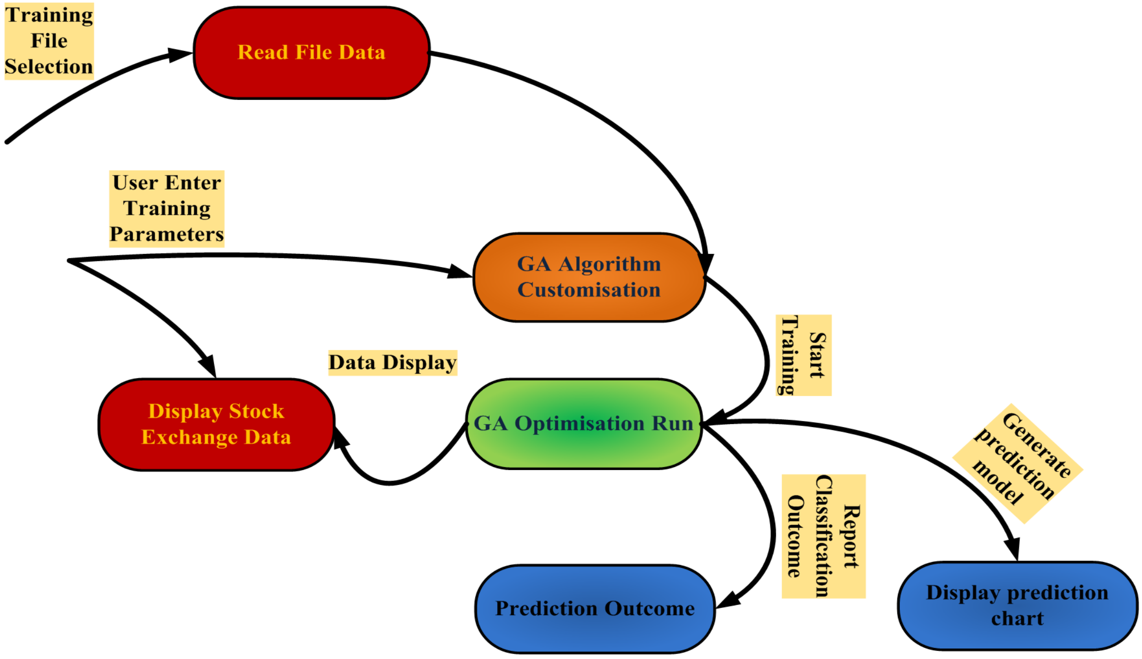

Training File Selection
Read File Data
User Enter Training Parameters
GA Algorithm Customisation
Start Training
Data Display
Display Stock Exchange Data
GA Optimisation Run
Generate prediction model
Report Classification Outcome
Prediction Outcome
Display prediction chart

Figure 15: The System Context Diagram representing various contexts of the STMgS project.

5.3.3 System Data Flow Diagrams (DFDs)

Data Flow Diagrams are systematic representation of flow of information between various entities of a system. Opposed to the SCD, DFDs are composed of multiple levels where each increasing level provides more details than the previous one. A standard DFD comprise of three main elements as shown in Figure 16.

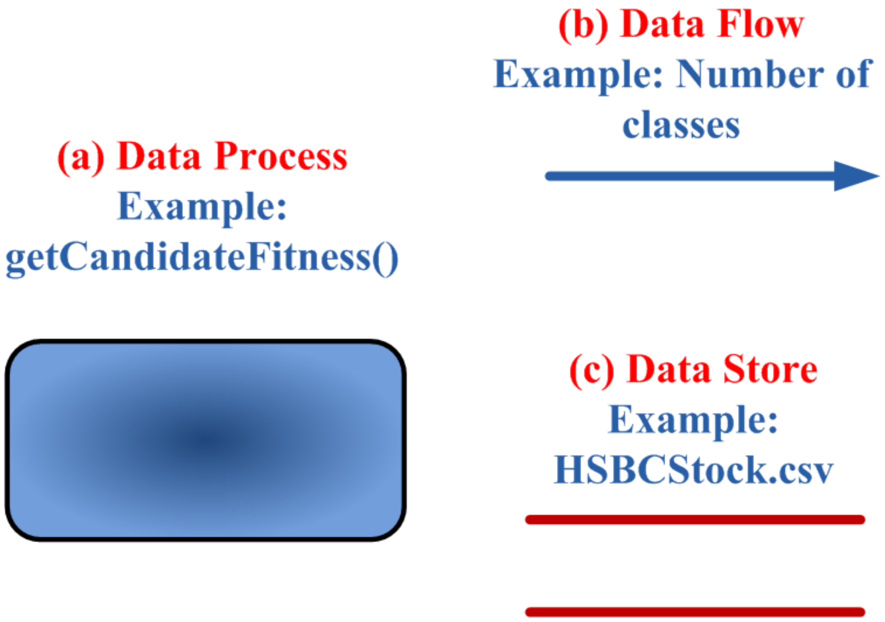


Figure 16: Thee main elements of a standard DF diagram representing the data flow to and from the data stores. (a) Representation of a data process, (b) representation of a sample information/data flow and (c) a data store which can be anything from a file to a database table.

DF diagrams are divided into various levels starting from "*Level 0*" and with each level increment, complexity/detail also increases.

The Level-0 DFD as shown in Figure 17 represents the processes involved in the first customisation interface screen of the user interface.

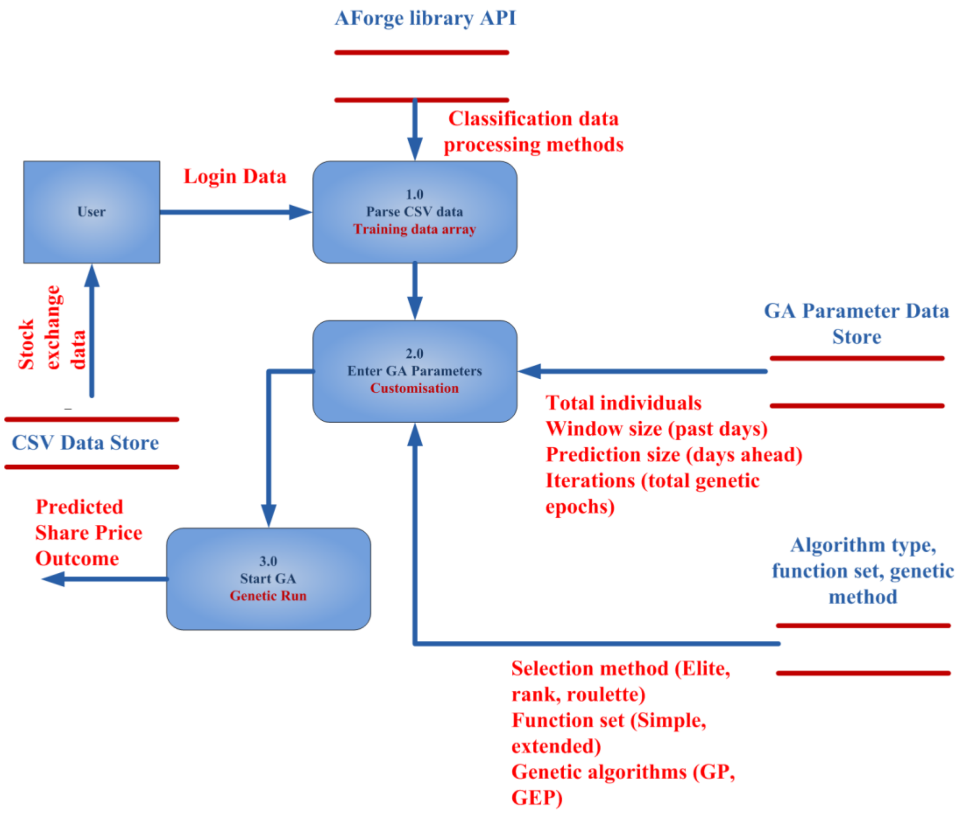

AForge library API
Classification data processing methods
User
Login Data
1.0
Parse CSV data
Training data array
Stock exchange data
CSV Data Store
2.0
Enter GA Parameters
Customisation
GA Parameter Data Store
Total individuals
Window size (past days)
Prediction size (days ahead)
Iterations (total genetic epochs)
Predicted Share Price Outcome
3.0
Start GA
Genetic Run
Algorithm type, function set, genetic method
Selection method (Elite, rank, roulette)
Function set (Simple, extended)
Genetic algorithms (GP, GEP)

Figure 17: A Level-0 dataflow diagram representing the algorithm customisation of the user interface.

The Level-1 DFDs further dissect processes to provider finer details into the operation of the underlying system. Generally, there should be Level-1 DFD diagrams equal to the processes present in the Level-0 DFD. These sub-modules are also known as children and are represented in '#.#' format. These processes are then further detailed for all the children (e.g. 2.1) with both levels balanced properly. In the proposed case, only the AI regression process bears crucial information regarding the operational details of the proposed algorithm and therefore it is shown in Figure18.

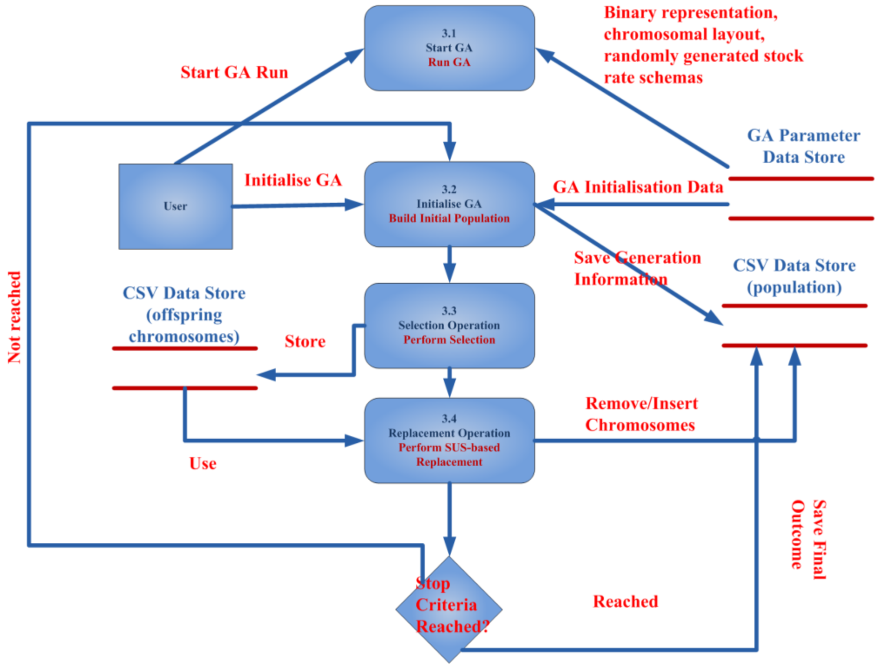
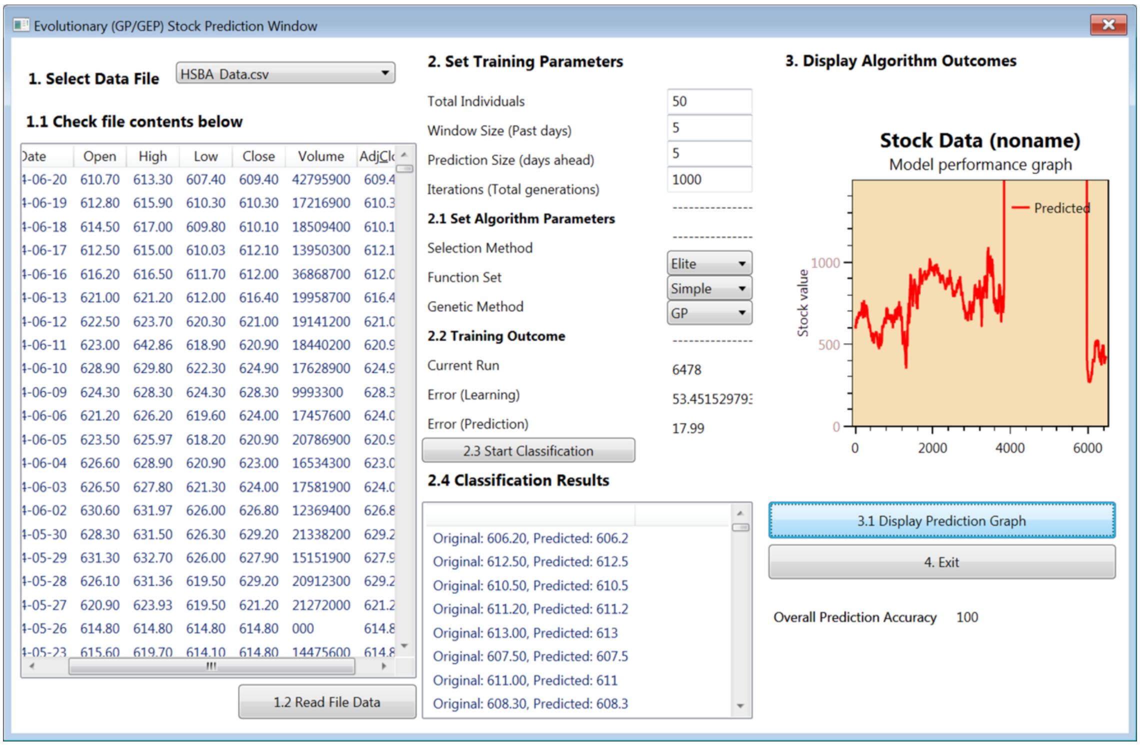


Figure 18: The Level-1 DFD for the classification process shown as process 3.0 in Figure 17.

5.3.4 System programming and implementation

Based on the SCD and DFD representations of the system, the system was implemented in C#.Net with its initial AI programming done in Matlab. The main user interfaces are shown in Figure 19 and 20.



1. Select Data File: HSBA Data.csv

1.1 Check file contents below

Date	Open	High	Low	Close	Volume	AdjClo
4-06-20	610.70	613.30	607.40	609.40	42795900	609.4
4-06-19	612.80	615.90	610.30	610.30	17216900	610.3
4-06-18	614.50	617.00	609.80	610.10	18509400	610.1
4-06-17	612.50	615.00	610.03	612.10	13950300	612.1
4-06-16	616.20	616.50	611.70	612.00	36868700	612.0
4-06-13	621.00	621.20	612.00	616.40	19958700	616.4
4-06-12	622.50	623.70	620.30	621.00	19141200	621.0
4-06-11	623.00	642.86	618.90	620.90	18440200	620.9
4-06-10	628.90	629.80	622.30	624.90	17628900	624.9
4-06-09	624.30	628.30	624.30	628.30	9993300	628.3
4-06-06	621.20	626.20	619.60	624.00	17457600	624.0
4-06-05	623.50	625.97	618.20	620.90	20786900	620.9
4-06-04	626.60	628.90	620.90	623.00	16534300	623.0
4-06-03	626.50	627.80	621.30	624.00	17581900	624.0
4-06-02	630.60	631.97	626.00	626.80	12369400	626.8
4-05-30	628.30	631.50	626.30	629.20	21338200	629.2
4-05-29	631.30	632.70	626.00	627.90	15151900	627.9
4-05-28	626.10	631.36	619.50	629.20	20912300	629.2
4-05-27	620.90	623.93	619.50	621.20	21272000	621.2
4-05-26	614.80	614.80	614.80	614.80	000	614.8
4-05-23	615.60	619.70	614.10	614.80	14475600	614.8

1.2 Read File Data

2. Set Training Parameters

Total Individuals: 50
Window Size (Past days): 5
Prediction Size (days ahead): 5
Iterations (Total generations): 1000

2.1 Set Algorithm Parameters

Selection Method: Elite
Function Set: Simple
Genetic Method: GP

2.2 Training Outcome

Current Run: 6478
Error (Learning): 53.45152979
Error (Prediction): 17.99

2.3 Start Classification

2.4 Classification Results

Original: 606.20, Predicted: 606.2
Original: 612.50, Predicted: 612.5
Original: 610.50, Predicted: 610.5
Original: 611.20, Predicted: 611.2
Original: 613.00, Predicted: 613
Original: 607.50, Predicted: 607.5
Original: 611.00, Predicted: 611
Original: 608.30, Predicted: 608.3

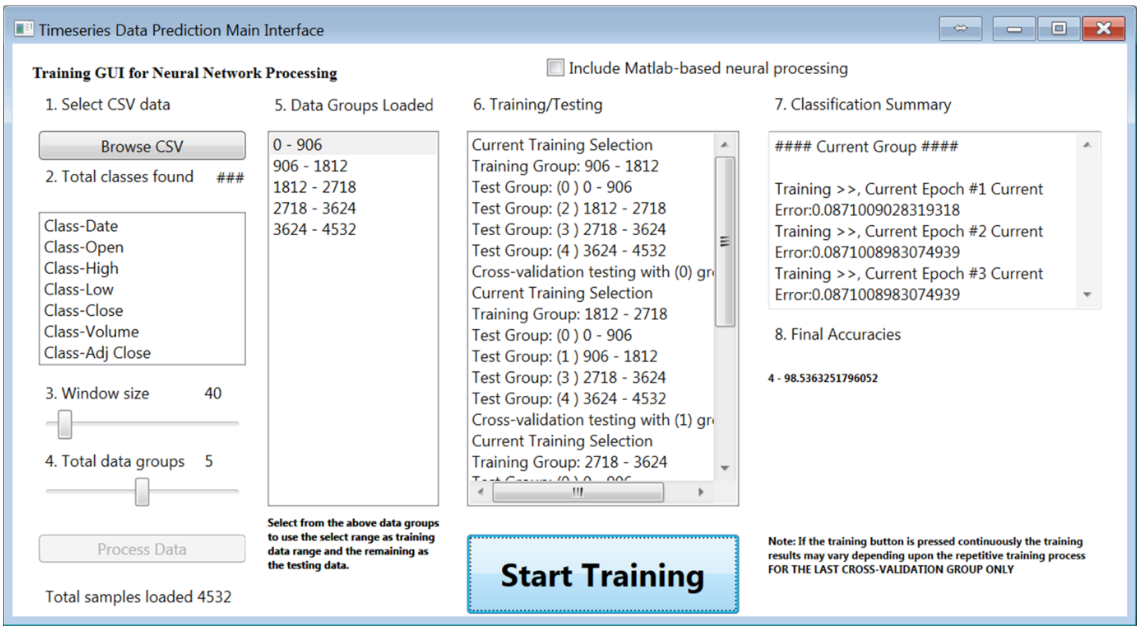
3. Display Algorithm Outcomes

3.1 Display Prediction Graph

4. Exit

Overall Prediction Accuracy 100

Figure 19: Main interface for the GP/GEP classification interface


Timeseries Data Prediction Main Interface

Training GUI for Neural Network Processing

□ Include Matlab-based neural processing

1. Select CSV data

Browse CSV

2. Total classes found ###

Class-Date
Class-Open
Class-High
Class-Low
Class-Close
Class-Volume
Class-Adj Close

3. Window size 40

4. Total data groups 5

Process Data

Total samples loaded 4532

5. Data Groups Loaded

```
0 - 906
906 - 1812
1812 - 2718
2718 - 3624
3624 - 4532
```

Select from the above data groups to use the select range as training data range and the remaining as the testing data.

6. Training/Testing

```
Current Training Selection
Training Group: 906 - 1812
Test Group: (0 ) 0 - 906
Test Group: (2 ) 1812 - 2718
Test Group: (3 ) 2718 - 3624
Test Group: (4 ) 3624 - 4532
Cross-validation testing with (0) gr
Current Training Selection
Training Group: 1812 - 2718
Test Group: (0 ) 0 - 906
Test Group: (1 ) 906 - 1812
Test Group: (3 ) 2718 - 3624
Test Group: (4 ) 3624 - 4532
Cross-validation testing with (1) gr
Current Training Selection
Training Group: 2718 - 3624
```

Start Training

7. Classification Summary

```
#### Current Group ####

Training >>, Current Epoch #1 Current
Error:0.0871009028319318
Training >>, Current Epoch #2 Current
Error:0.0871008983074939
Training >>, Current Epoch #3 Current
Error:0.0871008983074939
```

8. Final Accuracies

4 - 98.5363251796052

Note: If the training button is pressed continuously the training results may vary depending upon the repetitive training process FOR THE LAST CROSS-VALIDATION GROUP ONLY

Figure 20: Main interface for the ANN classification interface including the CrossValidation testing implementation

In summary, the chapter discusses design techniques and stages involved in the design, development and analysis of the underlying software system. Various software system development techniques have evolved since the inception of modern-day computing in late twentieth century. Each of these techniques has benefits and disadvantages depending on the scope of each project. As a result, one single software development methodology cannot be feasible for all projects. The SA (Structured Analysis) has been identified as the conventional systems design and development technique. This technique contains a series of stages similar to a waterfall model which include inception, analysis, design, implementation and maintenance phases. A complete iteration of these stages is considered as the SDLC However, in modern, complex and time-driven software systems that involve constant changes, the use of a full SDLC model is not an efficient solution. To meet the modern-day demands, agile methods have recently been launched with the advent of the object-oriented design and development paradigm. These techniques are a new class of system development methodologies that use an iterative model of development. The model represents a sequence of iterations, which may occur in response to the client’s requirement or as a way of improving overall results of a project. The RAD, JAD and software prototyping are examples of standard approaches under this model. The main aim of these techniques is to improve the overall software development process by bypass many of the SDLC phases. This strategy not provides an opportunity to assess a project's performance within short time intervals, but also facilitates faster change integration which is difficult to achieve in the conventional model. Various industrial development standards and algorithm stages have also been explored. For instance, the chapter explored Data Flow Diagrams in which each level provides more details than the previous one in the flow of information between various entities of a system. Notably, a standard DFD comprises of three main elements which include data process, data flow and data store.

The next chapter primarily covers the algorithmic outcome of the two methodologies proposed earlier and therefore, helps achieve Objective 5.

6 Chapter6: Results and Critical Analysis

6.1 Outcome of Sliding window-based FIS system to identify forecast anomalies

The first module of the system uses a financial domain analyst's expert knowledge to build a fuzzy inference system. However, as discussed earlier, due to high complexity of the actual system, the initial "expert-tuned" system was only limited to a two-input and single output prediction system. The final "AI-trained" system uses novel, unseen financial forecasting data to present its outcomes based on a previous set of inputs. The outcome of this model broadly entails a manually tuned FIS which is eventually able to predict time-series-based data to predict future forecasts. The features extracted for this model are based on a data clustering methodology (discussed earlier-on) via a balanced sliding window kurtosis value taken on both the sides.

The resultant statistical trend is aimed to assist experts in the assessment of subtle data anomalies within complex time-series forecasts. This is achieved by the statistical ability of the Kurtosis measure to detect abrupt variations in time-series data. That is, for a normally operating stock market, a sudden share price change can be efficiently detected if a sliding-window Kurtosis measure spanning previous historic values is iteratively measured. The sample stock data to explain the underlying concept was downloaded from Yahoo! Finance (2014). The data contains a daily trading volume of stock volume and prices from 12/04/1996 to 31/08/2014 containing the following five parameters:

- Open (share price)
- High (share price)
- Low (share price)
- End-of-day Close (share price)
- Volume (trade volume in US$)

The data is extracted for adaptive neuro-fuzzy training based on a sliding-window operation. Based on the single-step (one-day) sliding window operation, a feature vector containing a set of input vectors and the output (closing value) will be obtained in a row-wise fashion. Each row represents a single day prediction based on the previous ‘n’ number of days.

This study uses experimental data from Yahoo Finance (YHOO) to evaluate the performance of the proposed methodology. The data has been extracted from 12/04/1996 to 11/04/2014 comprising a total of 4532 days. For the entire duration of data, the pattern represents substantial fluctuations in stock prices and the overall trade volume (Figure 21 (b) and (a), respectively). An example can be seen in Figure 21 (b) where a clear depression starting from value 2146 can be seen which an indication of the start of global recession which started around year 2006.

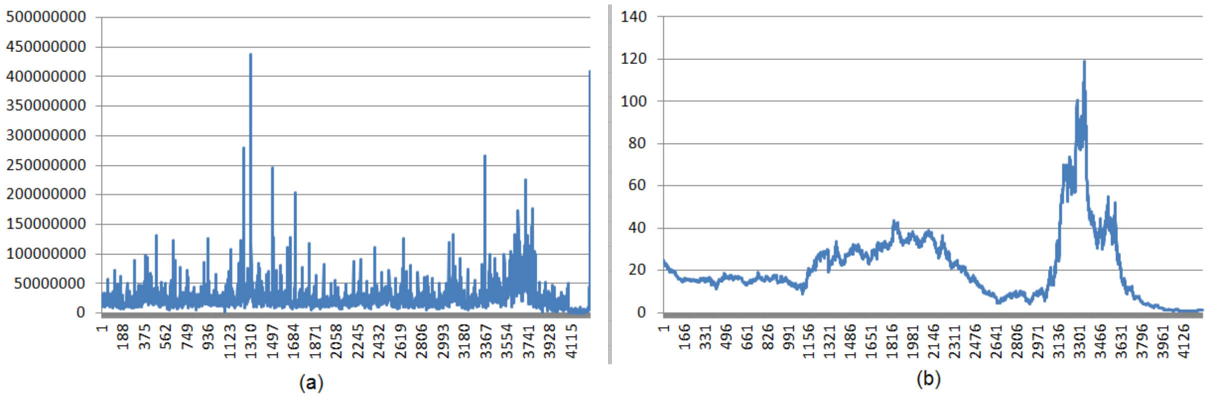


Figure 21: (a) Stock prices and (b) trading volume during the entire 18-year duration of the stock market of yahoo data

The closing, low and high stock values for the entire duration are shown in Figure 22, which shows substantial fluctuations in stock market values during the daily operating hours. This measures a significant justification for the utilisation of all the four (i.e. close, low, high and adj close) values in regression training in addition to the trading volume measure. The justification lies in the fact that the opening stock price of a share may substantially change by the end of the trading day and may therefore change the closing stock price drastically.

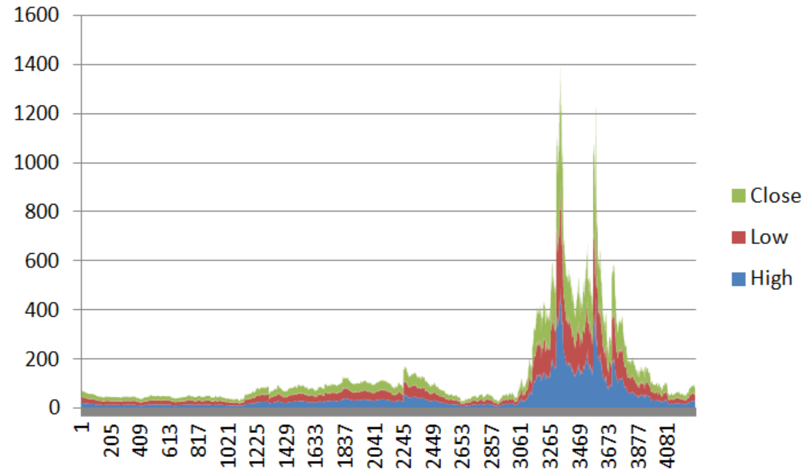


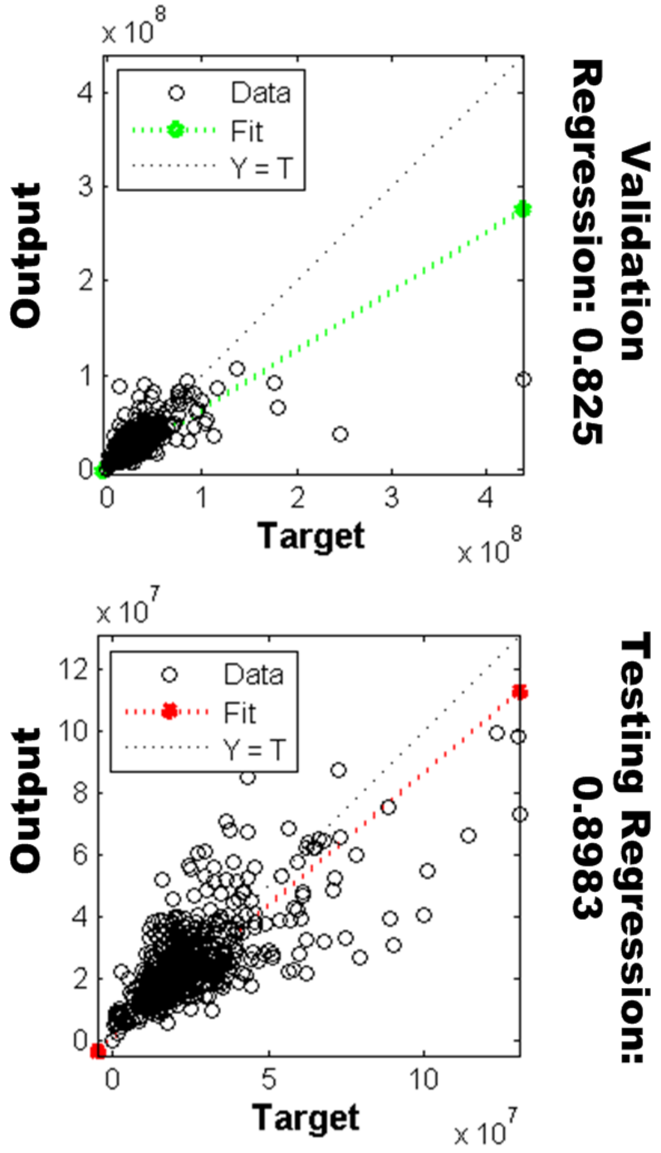
Figure 22: Closing, low and high share value limits during the entire 18-year duration of the stock market data

6.1.1 System Implementation

The system was trained against the AForce.Neuro neural computation library with extensions made to the AForge.Fuzzy computations library for the hybridised implementation of the ANFIS framework. The training was based on a 10-day-delay with 10 neurons via a nonlinear autoregressive classification (Meng & Koch 2009; Yusuf *et al.* 2013). The data was divided into three randomly selected groups with training, testing and validation data selected at 75, 15 and 15% respectively. The 75-15-15 is a standard machine learning training practice used in research which was adopted from standard Matlab ANN toolbox (Matlab, 2014a). It must be noted that validation data group was only used to measure network generalisation where the training was halt if the generalisation stopped improving for at-least 5 consecutive epochs. An epoch in ANN terminology is the completion of a single training iteration which leads either to the termination of the training sequence or the start of the next iteration based upon the criteria set in the initialisation stage of the training process. The data division left 17980 target time steps of data for training, and 3853 days each for validation and testing purposes. The non-linear auto-regression for this training is described by the equation given below, where d = 10 days (Mathworks, 2014c):

$$y(t) = f(y(t-1), y(t-2), \ldots, y(t-d)) \tag{20}$$

The equation shown in (20) shows a sliding window operation based upon previous d=10 values to predict share prices on the 10th day. The algorithm was run over a range of randomly selected data combinations and generated promising regression outcomes particularly over test and validation data as shown in Figure 23. A regression value closer to 1 means a close regression relationship between outputs and targets whereas a value closer to zero shows a poor correlation and therefore a poorly trained system. The validation performance plotted during the 20-epoc training cycle generated a low mean-square-error (MSE) pattern which also demonstrates an optimally converged network.



Validation Regression: 0.825

Testing Regression: 0.8983

Figure 23: High regression closure values depicting a robustly trained ANN regression.

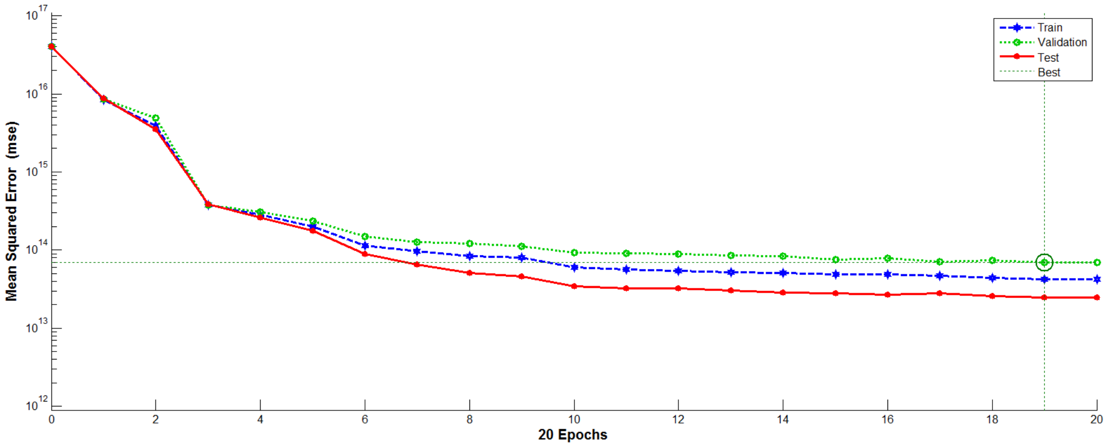
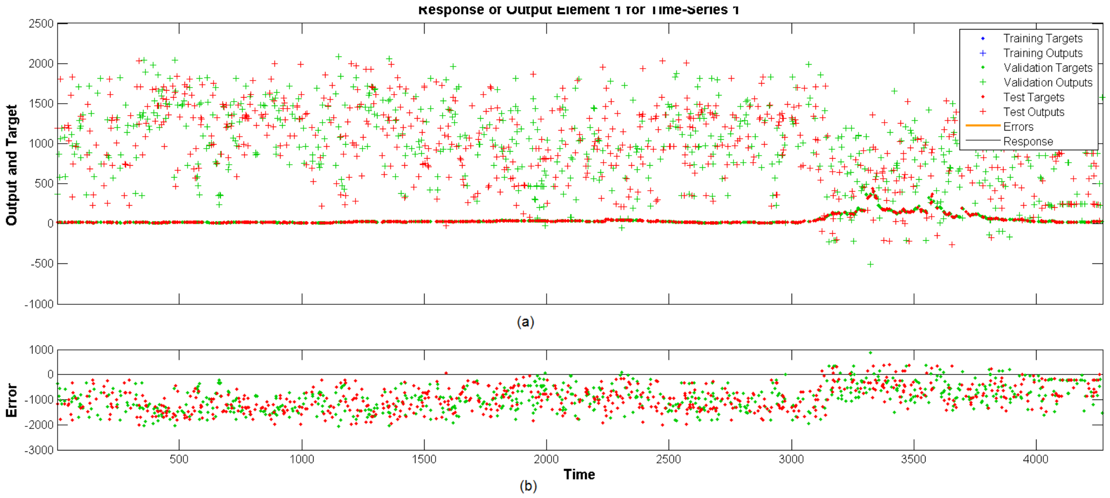


Figure 24: Validation MSE performance during network training.

Figure 24 demonstrates the ability of the underlying training sequence to have improved the overall actual-to-predicted mean square error (MSE). The best prediction outcome was shown to be from training data which is apparent due to the fact that training sequences are already used and known to the system which is a clear indication why the overall training error is lower when compared to validation error. The highest validation MSE is attributed mainly to the fact that it is obtained when the trained regression is used against unseen data. On top of it, validation is also used to terminate the training sequence when it sees 5 consecutive MSE increments in continuous epochs. The test MSE is lower than the remaining two datasets which may be attributed to the fact that test sequences generally see a trained regression and do not tend to see an uncertain regression which is being trained.



(a)

(b)

Figure 25: (a) Output and target plot of testing (red markers) and validation data (blue markers) and (b) the respective error plot.

The overall system outcome presents outstanding accuracy as evident from Figure 25 The markers for both '.' and '*' represent the target and output comparison for both validation (blue) and test data (red). In Figure 25, the majority of error values can be seen during the 2006 global recession time (see right-most part of Figure 25(a)). Nonetheless, the majority of correct classifications are shown as test values which demonstrate the viability of this regression to predict stock data. A sparse spread shows outstanding neural regression accuracy. A sparse error basically indicates a better trained regression which is expected to demonstrate higher prediction accuracy when subjected to unseen data sequences.

6.1.2 Validation Techniques:

Cross validation is a validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. In a prediction problem, a model is usually given a dataset of known data on which training is run (training dataset), and a dataset of unknown data (or first seen data) against which the model is tested (testing dataset) (Refaeilzadeh *et al.* 2009)

The overall accuracy of the system was evaluated against two standard testing methodologies of k-fold and jack-knife-based techniques with k = 5. The overall accuracy of these measures is shown in Table 6. The k-fold validation randomly divided unseen data into 5 unique sets out of which 4 sets were used for localised training, testing and validation. Once trained, the trained regression was then used against an unseen (5th) dataset with the prediction outcome recorded. In the next cycle, "group 2" was used as a baseline group against a regression trained on Group 1, 3, 4, 5. The overall accuracy was thus reported as shown in Table 6.

In the jack-knife-based technique , the entire dataset was sliced at a random location and the system trained over the data on the left-hand-side of the slice. The procedure was repeated three times with the outcome recorded. The average accuracy in this case was relatively lower than the k-fold validation with 86.72% which may be attributed to the possibility of a very narrow jack-knife slice resulting in the majority of outliers being left on the test side instead of the training side. Nonetheless, the overall system accuracy provides a promising venue for the underlying system to be further improved and extended.

Group	5-fold validation (%)	Jack-knife (%)
1	92.71	-
2	85.71	-
3	92.87	-
4	89.54	-
5	88.95	-
Slice – 1	-	93.23
Slice – 2	-	78.17
Slice – 3	-	88.76
	89.956	86.72

Table 6: Overall ANFIS prediction accuracy based on 5-fold cross-validation and jack-knife-based testing.

6.2 Outcome and analysis of the GEP prediction methodology

As elaborated in section 4.1, this research aims to critically analyse the GEP sliding window algorithm against a diverse range of datasets and varying prediction scenarios ranging from input window containing 2+ month trailing stock data to a week's data. The data was selected for five well-known stock companies, namely Yahoo, British Petroleum, GSK, HSBC and the RBS. In order to understand the difference of these companies' performance, it is necessary to look into the nature of performance they showed during the past 20+ years. The GEP prediction model for Yahoo compared with the previously reported ANFIS model. All the dataset described below were trained and evaluated over two window sizes of 5 days (short-term) and 56 days (medium-term).

Figure 26 represents stock data from Yahoo and British Petroleum where the later at present shows a closing price of 523.9, a dropping trend of 6.28 at 1.2%.

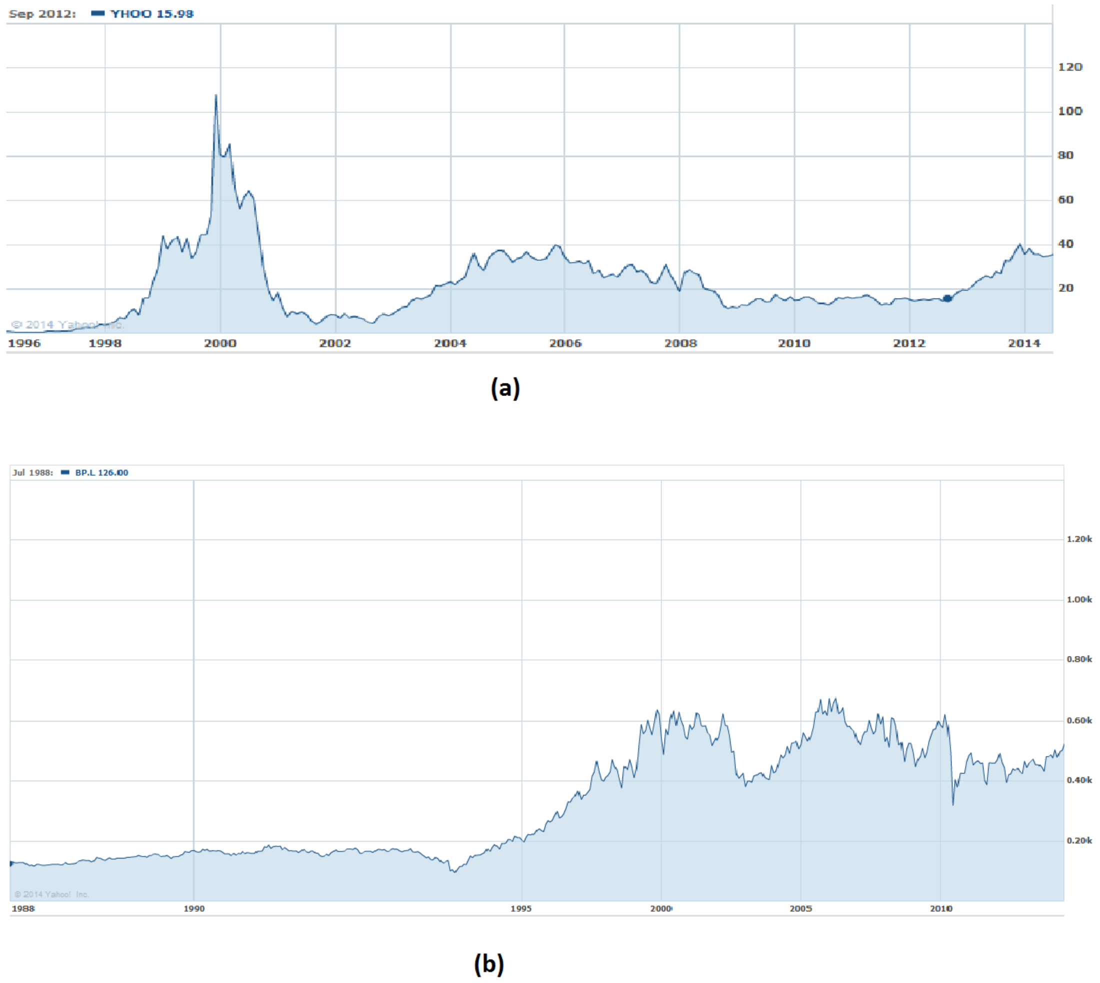


Figure 26: Baseline Yahoo (Yahoo, 2014) and British Petroleum stock data.

From Figure 26, in which BP's performance since the 1980s is depicted, the company experienced a spontaneous and gradual increase in stock prices that was also marked by abrupt patterns during the post-oil crisis era of Middle East's war and political turmoil.

Glaxo Smith Kline (GSK) was selected because of its unusually spontaneous stock turmoil during early 1990s that was caused by increasing competitions from Wellcome PLC, which launched a large number of pharmaceutical drugs in the same period. The company's stock currently stands at a falling pattern of 14 (0.89%) at the stock price of 1562.95 as shown in Figure 27 below.

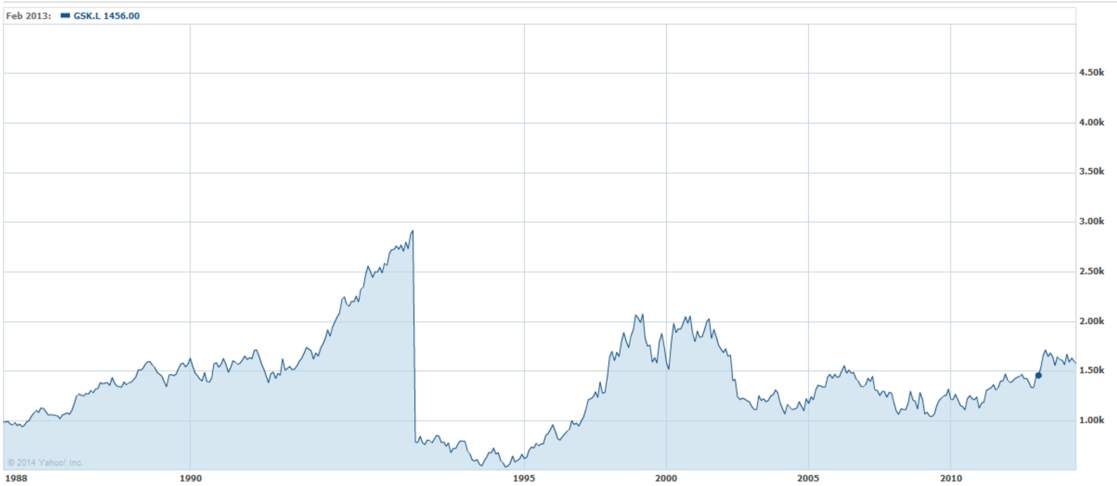


Figure 27: GSK stock data showing an abrupt fall during the early 1990s due to increased competition (GSK, 2014).

The two banks involved in these tests were HSBA and RBS, which showed a phenomenally large drop in shares during the past 5 recession years (see Figure 28). The shares continue to fall at respective prices of 11.02 (1.96% fall) and 604.5 (0.15%) for RBS and HSBA PLCs.

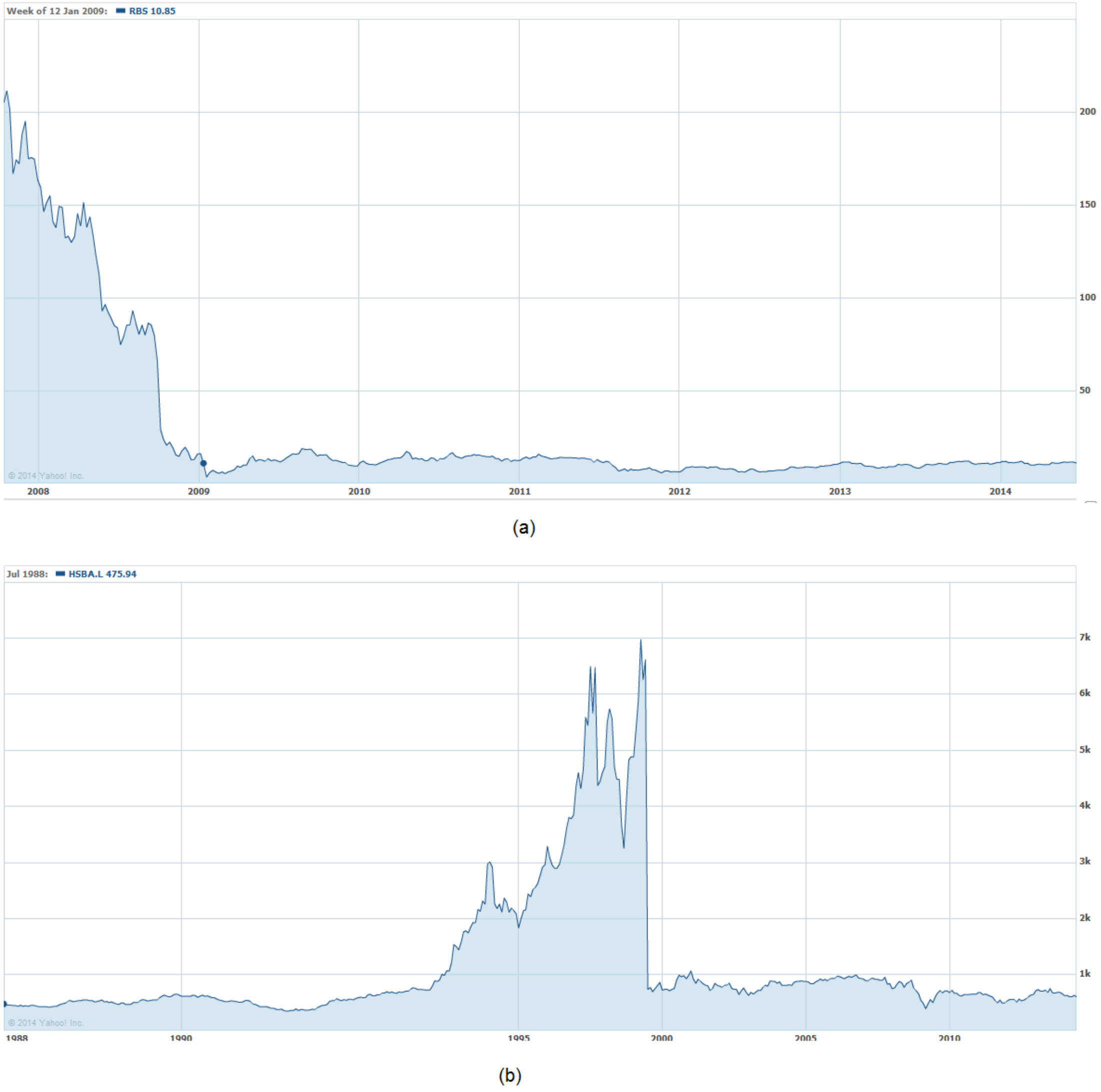


Figure 28: HSBA/RBS stock data showing a huge fall during the global recession (HSBA, 2014; RBS, 2014).

In the above four cases, it must be noted that both medium-term and short-term predictions bear important insight for traders to buy or sell certain shares. For instance, for long-term investors, an accurate and reliable prediction algorithm representing a correct trend (increasing or decreasing) will support in the decision making process.

Time limit	5 days
Dataset	6761 days starting from 1988
Currency	US$
Function set (basic)	+, -, *, /
Population size (generations)	100
Number of genes	4
Selection operator	Elitism, Rank , RW
Mutation rate	Adaptive (non-constant) FAMR
Crossover rate	60%

Table 7: Experimental parameters used for data presented in Figure 36, 37 and 38

Results presented in Figure 29, 30 and 31 represent the outcome of GEP based on parameters defined in Table 7. Green and blue plots in the figures represent actual and predicted stock rates whereas the red points show the error in prediction. The overall dataset contained a total of 6761 days of stock data where the initial 70% was used for training whereas the remaining 30% was used for testing where the data started from March, 2006. It must be noted that the 30% data used for testing contained a high number of outliers from the recent recession of 2006 – 2011 which made it particularly challenging for a 5-day prediction system. The high error values can particularly be seen where there are sudden and unpredicted change in stock values.

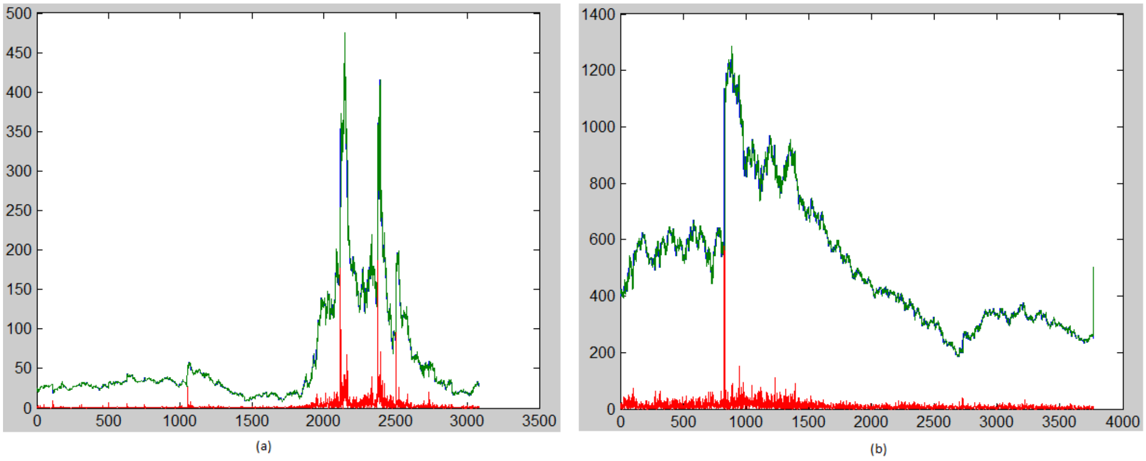


Figure 29: Actual (Green) and classified (Blue) values against error rate (Red) obtained via GEP-FAMR algorithm for (a) Yahoo and (b) British Petroleum stock data (BP, 2014).

Glaxo Smith Kline (GSK) was selected on its unusually spontaneous stock turmoil during early 1990s due to increasing competitions from Wellcome PLC which launched a large number of new pharmaceutical drugs during 1980s and early 1990s. The stock currently stand at a falling pattern of 14 (0.89%) at the stock price of 1562.95.

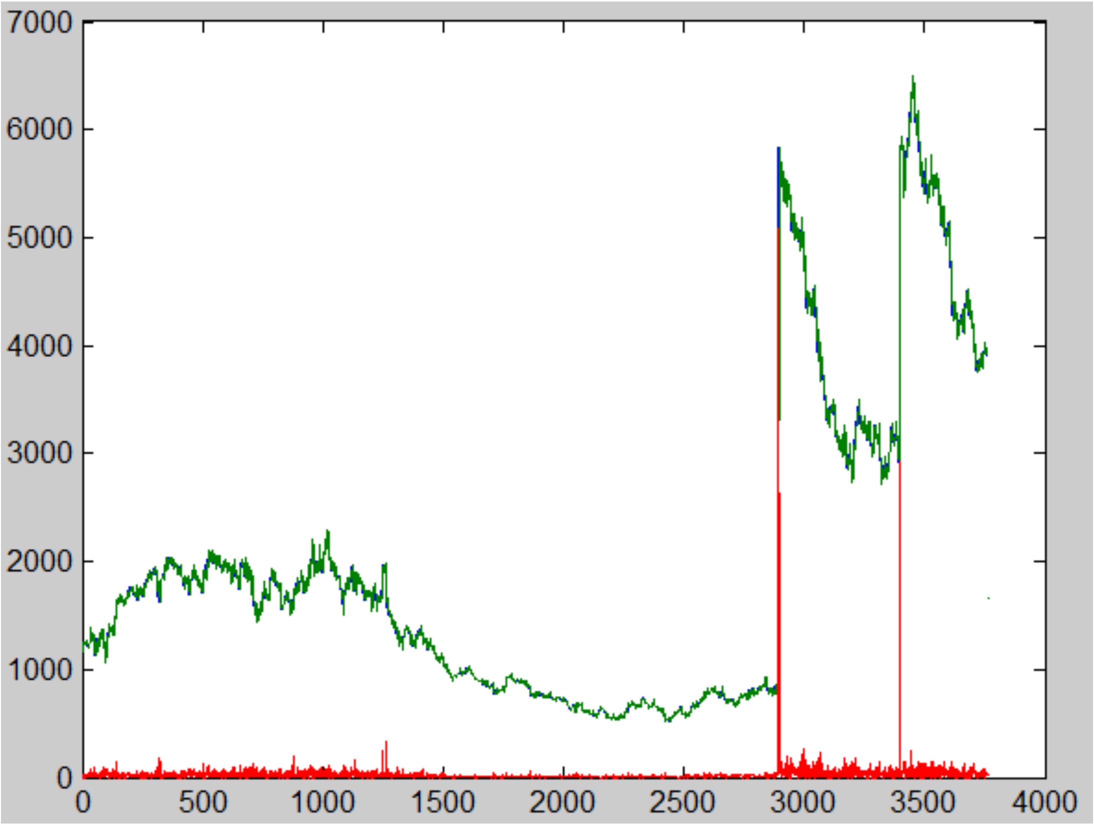


Figure 30: Actual (Green) and classified (Blue) values against error rate (Red) obtained via GEP-FAMR algorithm for GSK stock data (GSK, 2014).

The two banks involved in these tests were HSBA and RBS data which showed a phenomenally large drop in shares during the past 5 recession years (see Figure 31). The shares continue to fall at a respective prices of 11.02 (1.96% fall) and 604.5 (0.15%) for RBS and HSBA PLCs.

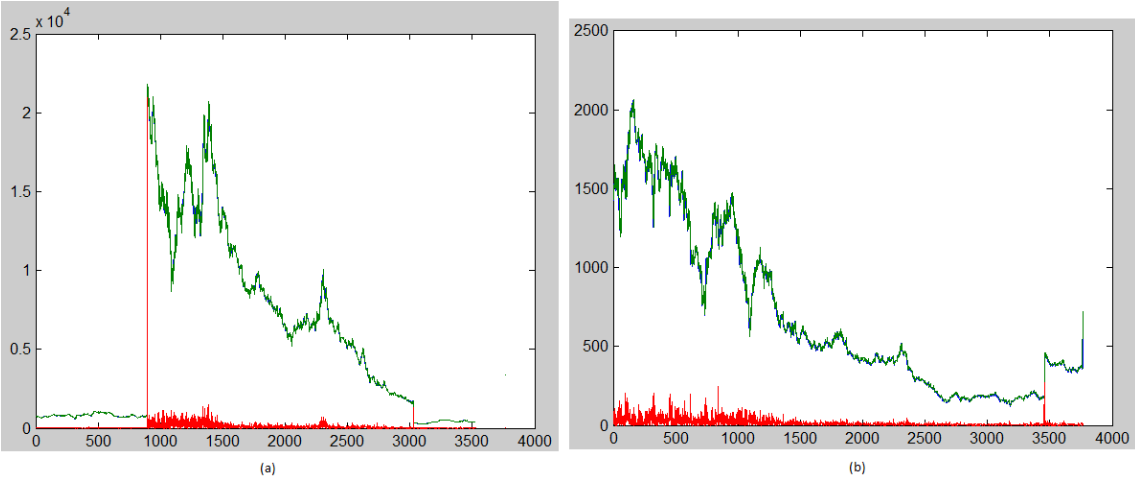


Figure 31: Actual (Green) and classified (Blue) values against error rate (Red) obtained via GEP-FAMR algorithm for HSBA/RBS stock data (HSBA, 2014; RBS, 2014)

In the above four cases, it must be noted that both medium-term and short-term predictions bear important insight for traders to buy or sell certain shares. For instance, for long-term investors, an accurate and reliable prediction algorithm representing a correct trend (increasing or decreasing) will support in the decision making process. However, the figures (Figures 29, 30 and 31) report the errors rates on a short-term regression only.

The performance of the proposed sliding window GEP algorithm was evaluated under the conditions that prediction of the existing pattern (increasing/decreasing) would be based on short and medium-term training and next-day stock prices as would be forecasted as accurately as possible.

6.2.1 Initial Selection Operator Evaluation

The algorithm's performance was initially evaluated over a fixed number of chromosomes with a total of 50 individuals (chromosomes) per generation and 1000 genetic runs with each performing/applying genetic crossover and mutation to induce diversity in the subsequent generation. The selection criteria were initially based upon Elitism, Rank-based and Roulette-wheel algorithms. As shown in Table 8, the proposed GEP-FAMR approach showed significant improvements for the cases of BP and GSK and RBS. The algorithm however showed average improvement when compared to RW selection and a marginal improvement in the case of Yahoo. Prediction errors were not focused on at this stage because of possible improvement later on once the remaining parametric combination (i.e. window-size, chromosome/generation, etc.) were evaluated.

Company	BP			GSK			HSBC/A			RBS			Yahoo		
Chromosomes /Generation	50			50			50			50			50		
Window Size/Category	5			5			5			5			5		
Genetic Iterations	1000			1000			1000			1000			1000		
Selection Method	*FAMR*	Rank	RW	*FAMR*	Rank	RW	*FAMR*	Rank	RW	*FAMR*	Rank	RW	*FAMR*	Rank	RW
Learning Error	*10.61*	14.87	18.89	*20.06*	37.09	29.74	*53.07*	120.00	53.45	*15.31*	23.00	28.89	*1.62*	1.63	1.632
Prediction Error	*6.32*	13.00	11.00	*25.11*	40.00	46.99	*16.50*	0.00	17.99	*5.76*	8.00	9.00	*0.89*	1.00	1.00

Legend:

RW: Roulette-wheel

Note: Prediction accuracies rounded to the nearest whole number and represented in US$

Table 8: Comparison of learning and prediction error rate for three selection criteria of Elitism, Rank-based and Roulette-Wheel (RW) selection.

6.2.2 Evaluation over short and medium-term data spans

Based on the abovementioned performance, the five datasets were tested against training performed under two window sizes of short 5-days and 56-day-lengths. Both combinations were repeatedly evaluated against simple and extended function sets, and genetic programming and GEP.

Table 9 presents a mixed response to the change of gene functions where simple arithmetic function and arguments do not seem to have an overall impact on the prediction accuracy. It must be noted that the values shown in Table 10 are the difference of stock value by which a prediction has given an erroneous value. The extended gene function class represents arithmetic functions $\mathcal{F} = \{\Re, *, /, -, +\}$ and some common arithmetic functions inclusive of the set $\{sin, cos, ln, exp, sqrt\}$. As discussed in the methodology section, this methodology is used by the chromosomes to build arbitrary expression via genetic operators.

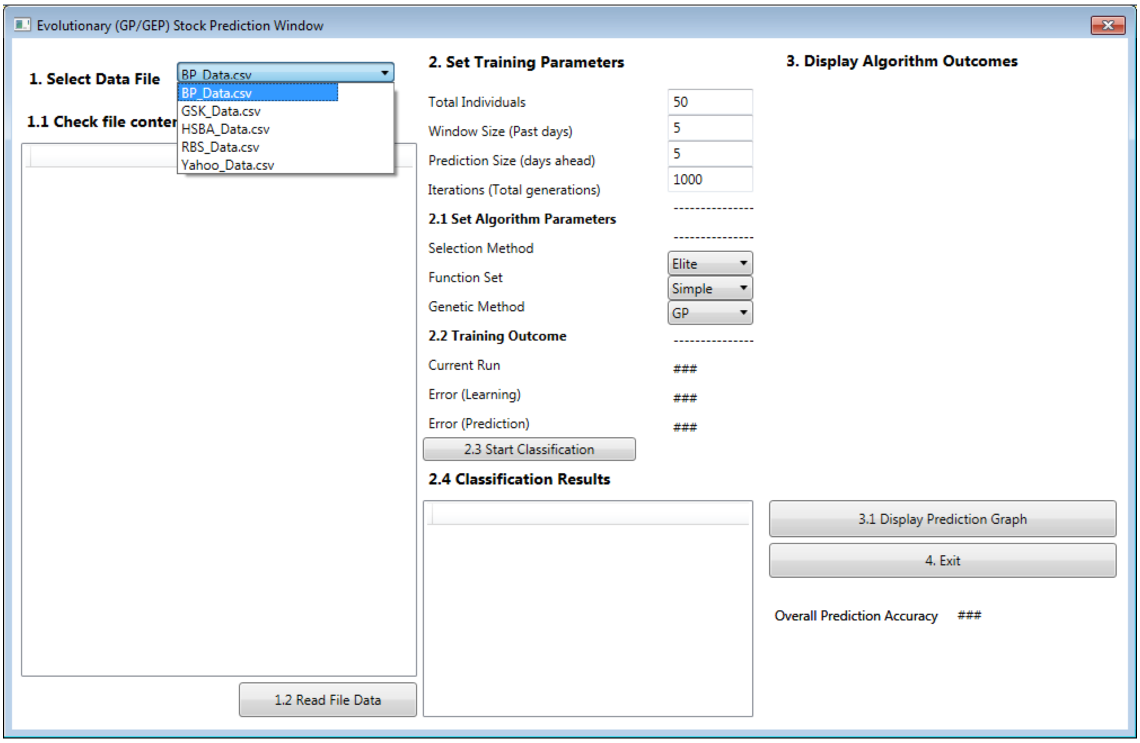
Short-ahead prediction: Prediction of the next day						
Gene Operations	Function/Arithmetic	BP	GSK	HSBA	RBS	Yahoo
Medium-term (56 days)	Simple	11	40	0	3	4
	Extended	4	24	9	8	1
Short-term	Simple	0	48	18	3	3
	Extended	13	23	0	8	2

Table 9: Comparison of short-ahead next-day-prediction based on medium-term and short-term look-back periods.

A critical view of the table however illustrates the fact that the extended functions do not substantially improve the overall prediction accuracy of stock trading systems at both the medium-term and short-term scales.

6.2.3 Software development environment and the GUI

The algorithms were implemented via AForge .Net software development kit over a Core i7 2.4GHz system operating with an 8GB of RAM. The implementation used C# 5 with .Net Framework 4.5 with Windows Presentation Foundation and OxyPlot Charting API to implement the final graphical user interface (GUI) shown Figure 32 below:



Evolutionary (GP/GEP) Stock Prediction Window

1. Select Data File BP_Data.csv

1.1 Check file conter

2. Set Training Parameters

Total Individuals 50

Window Size (Past days) 5

Prediction Size (days ahead) 5

Iterations (Total generations) 1000

2.1 Set Algorithm Parameters

Selection Method Elite

Function Set Simple

Genetic Method GP

2.2 Training Outcome

Current Run ###

Error (Learning) ###

Error (Prediction) ###

2.4 Classification Results

3. Display Algorithm Outcomes

Overall Prediction Accuracy ###

Figure 32: Main GEP/GP data selection, algorithm/parameter selection, training and display software with British Petroleum data currently selected

The C# API was primarily used because of its ability to develop extensible and industrial grade software systems that can be extended to multiple platforms with minimal restructuring effort. The average time required to run a genetic run comprising of 1000 generations with 50 chromosomes per generation and a sliding window size ranging from 5 (days) to 56 (days) was 11.33 seconds. This duration may increase or reduce based upon the underlying machine used. Moreover, the system extensively utilised parallel computing architecture which made use of the 4 processor cores in parallel which further increased the system responsiveness in asynchronous processing.

The interface window first requires the user to select from the four companies given in the top-left drop down box which populates the data for training. In the developed GUI shown in Figure 32, based on the parameters defined in Step 2, the classification performs an iterative genetic run whose outcome is eventually shown at Step 2.4. Once the button at Step 3.1 is clicked, the stock prediction outcome is displayed in the graph shown at top right corner of the screen. This graphical outcome can be cross-matched with the original data of the three companies provided in Figure 36, 37 and 38. Alternatively, single-values predictions can be evaluated under the classification results section.

Table 10 below demonstrates a comparison between GEP-FAMR with both the GA variants of the conventional genetic programming (GP) and the gene expression programming (GEP) –based classification. The comparison shows GEP-FAMR to perform substantially better for both the cases of short and medium-term prediction apart from the GSK dataset where GEP with fixed mutation rate performed better. This anomaly may be attributed to the initial random conditions on which the genetic run for the particular GEP-FAMR started or the fact that a low mutation rate may still result in the loss of fitter individuals in the case of GEP-FAMR which does not happen in constant-mutation-rate GEP where Elitism still retains the best performing individual anyway. Nonetheless, the overall success rate in prediction with other datasets indicates the suitability of GEP-FAMR on other methodologies.

	MT			ST		
	GEP-FAMR	GP	GEP	GEP-FAMR	GP	GEP
BP	91.14	89.39	87.01	96.32	96.23	89.76
GSK	98.45	76.34	100	99.12	76.65	100
HSBA	92.11	82.09	91.87	88.56	82.1	84.09
RBS	95.85	97.12	89.54	96.64	95	100
Yahoo	99.2	98.75	96.75	99.16	97.64	96.11
Overall accuracy	**95.35**	88.738	93.034	**95.96**	89.524	93.992

Table 10: Short and medium-term prediction accuracy based on GEP sliding-window operation (for 5-day look-ahead only).

6.2.4 Comparative analysis of ANFIS with GEP for noisy/outlier values

Based on the abovementioned outcomes for five datasets, the GEP-FAMR approach was found to have a high performance compared to standard Rank and RW approaches. However, the approach showed its vulnerability particularly against abrupt data variations under short-duration data value changes. These variations are highly likely in volatile datasets such as those encountered in global stock markets. In order to assess the suitability of the approach, it was evaluated against a short, 42-day dataset extracted from the BP dataset. Figure 33 shows GEP-FAMR to have shown a better recovery against such a change compared to the ANFIS algorithm. An extended comparison(see Table 11) of the 30% dataset extracted for GEP-FAMR testing against ANFIS also showed a substantial accuracy for the former methodology compared to ANFIS. Figure 33 below provides a comparative analysis of a 5-day prediction outcome error recovery capability of ANFIS and GEP-FAMR algorithms against sudden value changes in stock data profile of BP.

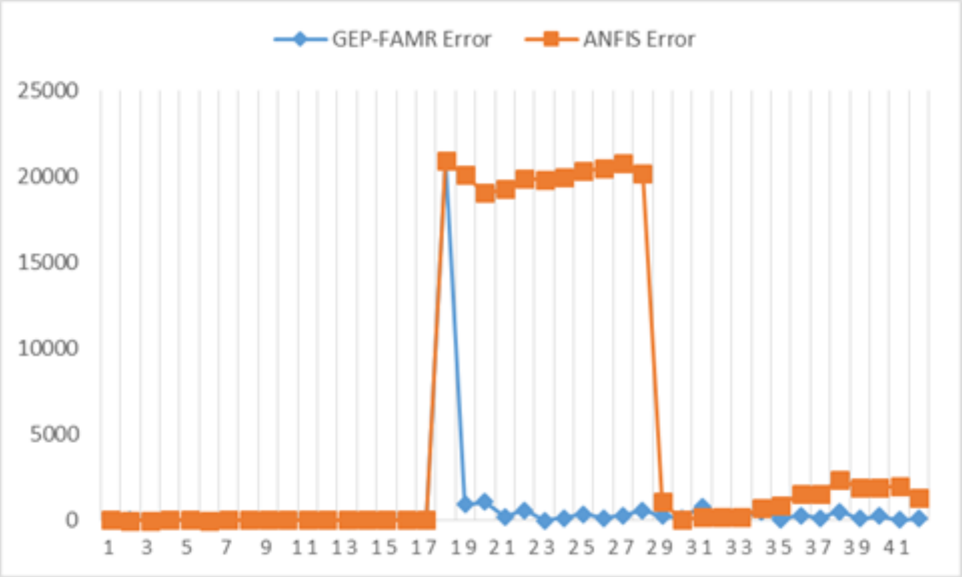


Figure 33: ANFIS and GEP-FAMR recovery vs sudden value changes in BP's stock data profile.

	BP	GSK	HSBC	RBS	Yahoo
GEP	97.99	98.83	98.7	96.74	97.49
ANFIS	96.63	94.21	97.56	94.19	95.78

Table 11: Comparison of 30% GEP-FAMR prediction accuracy and performance comparison of BP data with the ANFIS algorithm.

6.2.5 Benchmarking and Critical Analysis with Existing Research

The Yahoo dataset, along with other stock data including the HSBA, BP, GSK and RBS data is regarded as a standard stock dataset widely used as a benchmark to evaluate a wide range of machine learning algorithms as reported by Wang and Niu (2009) and Wang *et al.* (2009) and Bourdino *et al.* (2014).

ANN and evolutionary systems have lately been used in a wide range of research literature with each report different measures of improvements on the existing state of knowledge. A cross-analysis of these methodologies is provided in Table 12:

Author	Algorithm/Technique	Prediction Accuracy (MT/ST)	Regression Error (Yahoo data)	Year of Publication
Proposed technique(GEP-FAMR)	GEP	**95.96/95.35**	98.368	2017
Kumar & Murugan	ANN	N/A	97.4	2013
Devi *et al.*	ANN/Levenberg Marquardt	N/A	98	2013
Shen & Zing	BPNN/GA	97.48%	N/A	2009
Du *et al.*	Basic RBF	93.33%	N/A	2010
Skolpadungket *et al.*	GA/ANN	79-96%	N/A	2009

Table 12: A comparison of the best performing GEP methodology with the existing state of knowledge and research.

Comparison was made between recent outcomes from the literature from research focusing mainly on the application of ANN and evolutionary paradigm to the prediction of stock values. Based on the regression and classification error accuracies the following three achievements were accomplished:

- This was the first-ever attempt to use genetic evolutionary programming for the prediction of stock market data, and employ FAMR in improving the overall forecasting accuracy
- The algorithm reported a higher regression outcome of 98.368 when compared to the one proposed by Kumar and Murugan (2013) and Devi et al. (2009). Accordingly, a high regression outcome implies that present and past data can be used to predict financial forecasting systems with minimal errors. In fact, high regression implies that the developed system's prediction error is considerably low. The regression outcome obtained in this research (98.368) is different from the one proposed by Kumar and Murugan (2013) and Devi et al. (2009) (89.9)

for many reasons. Kumar and Murugan (2013) and Devi et al. (2009) base the figure on wide-ranging literature as opposed to empirical research. However, the main factor causing the obtained high regression figure is the use of the GEP algorithm, in which the gaps pointed out in the investigated ANFIS methodology were addressed through employment of various selection, optimisation and operation techniques.

- Shen and Zing (2009) reported a higher prediction accuracy of 97.48% compared to the proposed approach. However, there were no comparisons made with regard to the measure of computational efficiency gained as the methodology focused on pure ANN-based approach.

Based on the comparative analysis, the proposed algorithm presents a promising outcome when compared to existing literature. If the algorithm is further extended to various combinations of selection criteria, it is highly likely that the prediction outcomes can be further improved.

6.2.6 Systematic Analysis of GEP against standard datasets

The fractionally adaptive mutation technique was introduced in this work to improve the manner in which fittest individuals were retained in subsequent genetic runs. The underlying objective was to improve the overall Elitism-based selection mechanism on the fact that regardless of retaining the best-performing chromosome via Elitism, there was still a possibility to lose it if it was probabilistically selected by the system to be mutated. The data used for this analysis was taken from the standard BP dataset.

6.2.7 Comparison of fractionally adaptive GEP against standard optimisation methodologies

In order to compare the suitability of the proposed selection mechanism, the technique was compared against the following two existing standards of Elitism with random mutation and constant mutation (first two):

- Elitism with high/uncontrolled mutation rate (100% mutation probability);
- Elitism with constant mutation rate (1% mutation probability); and
- Proposed: Elitism with fractional mutation rate.

Based on the theoretical nature of GA a mutation rate of 100% effectively changes a GA run into a random search routine.

In the random search based on 100% mutation probability, the overall search lost 100% of its fittest chromosomes (subject to mutation) which were then supported by better individuals again in the subsequent generation that were again lost subsequently as shown in Figure 34 below:

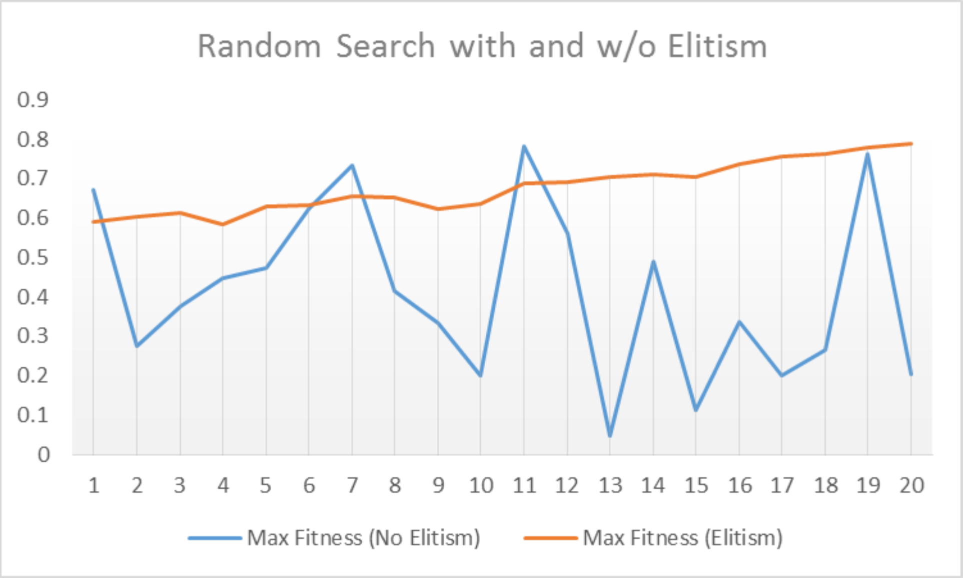


Figure 34: Representation of a genetic run with 100% mutation (random search: RS) and the resultant worst-to-best Elitism outcome.

The purpose of this elaboration was to show the impact of high mutation rate on seemingly better genetic runs. However, a simple single high fitness chromosome survival leads to an organised fitness improvement of 0.7875.

In Figure 35 below, a standard 1% mutation rate was applied and a both the approaches lead to a gradually improving fitness rate.

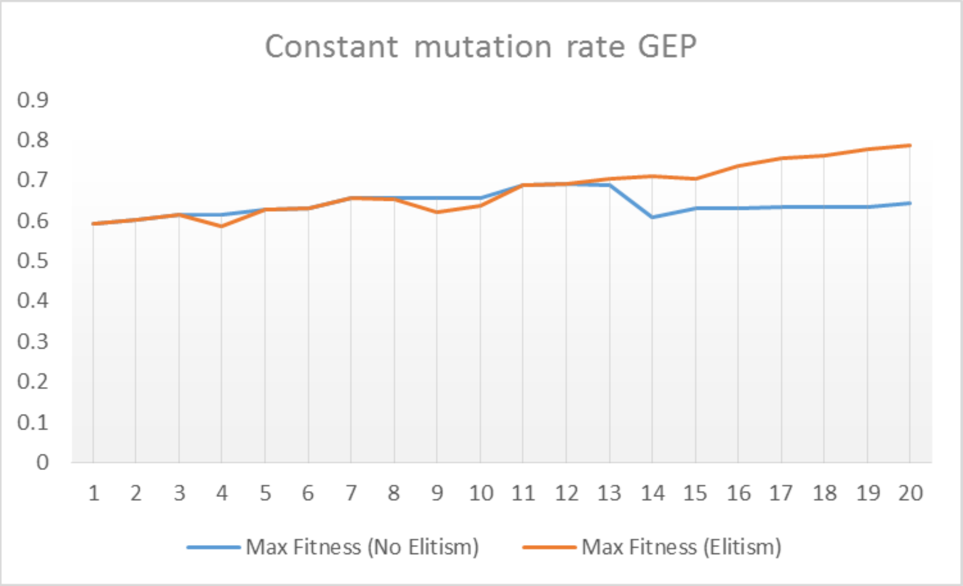


Figure 35: Representation of a genetic run with 1% constant mutation rate (CMR) and its impact on the best-fitness chromosome at generation 14.

However, due to an equal (1%) mutation rate, the fittest individual was itself mutated at generation 13 and since no Elitism was applied, the resultant fitness can be seen to have taken a considerable amount of time to recover and finally achieve a fitness score of 0.6444.

On applying the proposed adaptive mutation rate, the outcome shown in Figure 36 below was obtained.

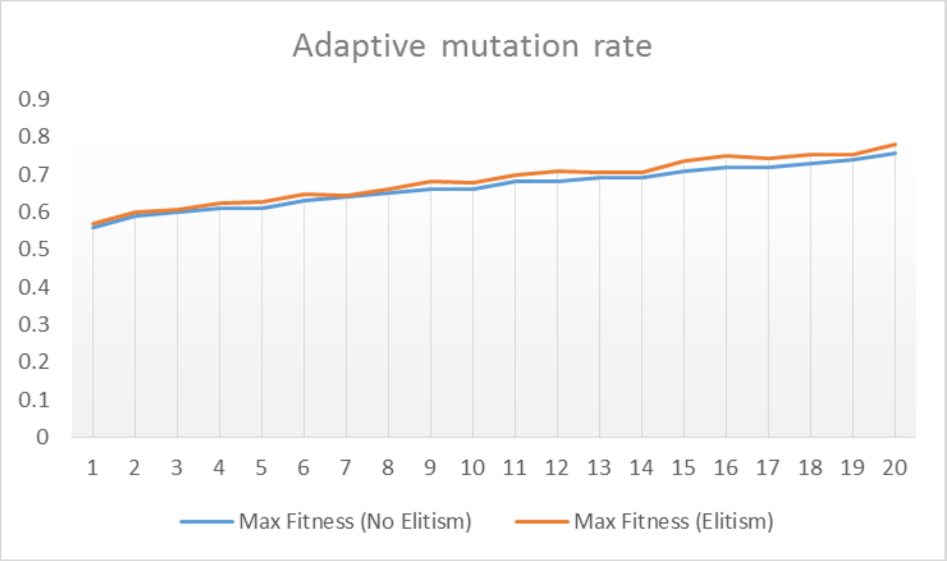


Figure 36: Representation of a proposed, GEP-FAMR fractionally adaptive mutation rate mechanism.

(The rate of change of fitness for all the fitness scores can be seen in Table 13.

Algorithm used	Elitism	No elitism
RS	0.73198	0.20655
CMR	0.665501	0.971033
AMR	1.057687	0.985493

Table 13: Rate of improvement of fitness for various search heuristics

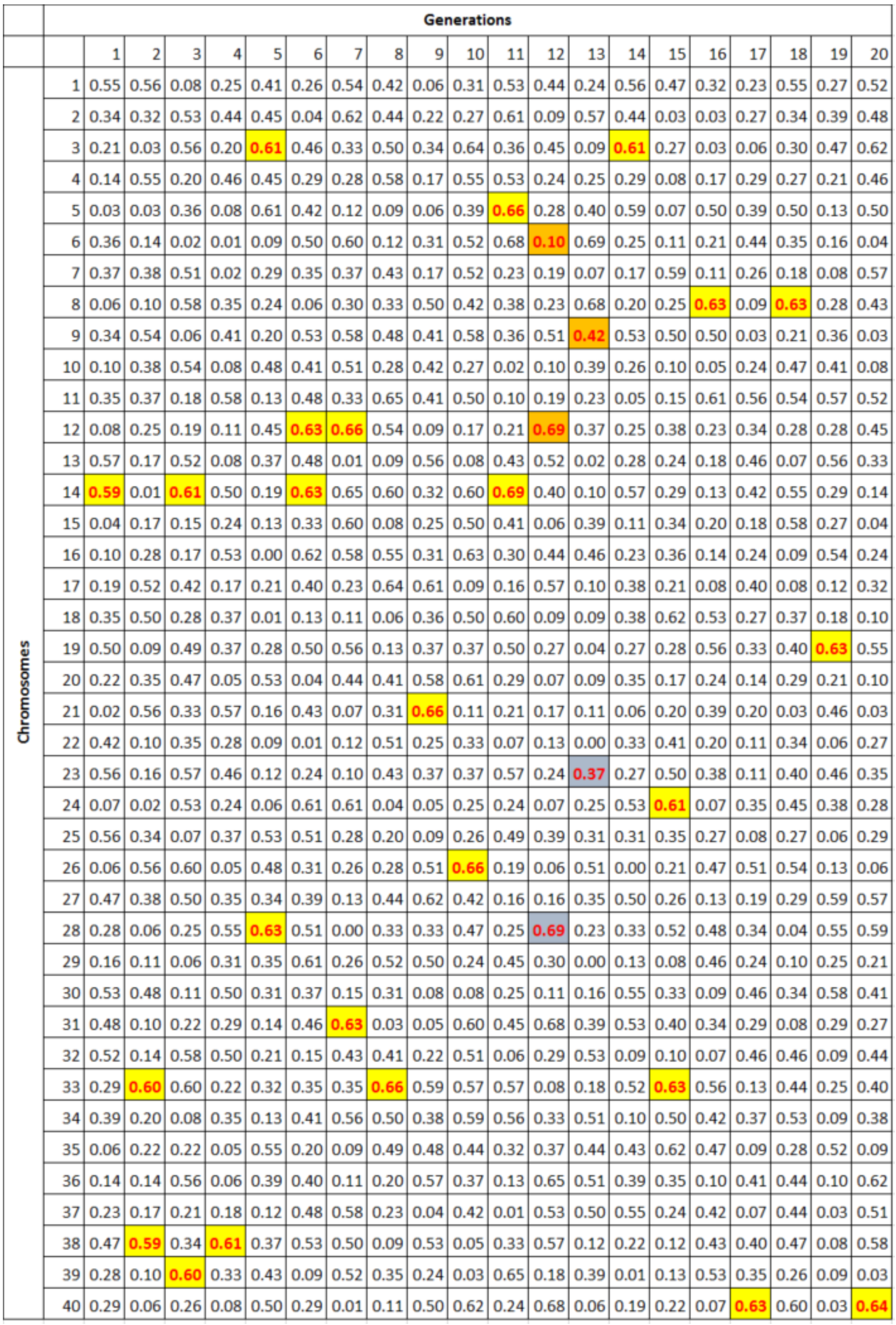
The low rate of fitness of constant mutation rate compared to random search must not be confused with a better performance of RS algorithm. Based on the inherent nature of the stage of randomisation, CMR is still more likely to perform better than RS in the absence of Elitism as evident from the average fitness achieved by RS under no Elitism. A detailed comparative analysis of the two techniques RS/CMR with the proposed GEP-FAMR is performed in Table 14. The rationale behind Elitism-based GEP-FAMR gaining better accuracy can still be attributed to the fact that with subsequent runs, the technique increases the total number of fitter solutions in the gene pool.

RS	CMR	GEP-FAMR
No Elitism: Each generation’s all chromosomes are mutated thereby changing even the highest fitness solutions. Therefore, there is no way of recovering a good solution even via Elitism.	**No Elitism**: There is a 1% chance the fittest chromosome will get lost. However, there is still likelihood that other closer-fitness chromosomes will live or crossover with others to move to other generation and generate better fitness individuals.	**No Elitism impact on better fitness chromosomes**: The mutation probability of high fitness individuals reduce which causing them to stay in subsequent generation, crossing with others and improving the chances of even better chromosomes. **No Elitism impact on lower fitness chromosomes**: The low-fitness solutions still stay in the pool and survive to crossover with other chromosomes to induce diversity in the genetic run. This reduces the chances of the search ending-up in a local maxima.
Elitism: The technique does retain the best	**Elitism**: The best individual will still move to the next generation but	**Elitism impact on better fitness chromosomes**:

individual but it still gets mutation (and lost) in the subsequent generation.	may get selected there for crossover/mutation/recombination with any other chromosomes to generate better (or worse) candidates. This situation may lead to a lot of similar-fitness solutions leading to fitness improvement getting stuck in a local maxima.	The best fitness solution still replaces the lowest-fitness solution. Despite other fitter solutions moving to the next population. **Elitism impact on lower fitness chromosomes**: Despite low-fitness solution having better chance to get to the next generation, the lowest one is still replaced by the best. This may marginally reduce the diversity of the overall population.

Table 14: Functional comparison of RS, CMS and AMR techniques.

The overall improvement in the rate of adaptive mutation with other models proves the hypothetical argument at the start of this research that fractional improvement is more likely to retain better individuals. Moreover, the per-generation improvement rate with Elitism presents a direct evidence that the technique will cope well with any other category of datasets. An entire genetic run Table 15 can be further analysed from where a simultaneous mutation and crossover of two best fitness chromosomes (both with a fitness of 0.69) are crossover into a child chromosome of fitness 0.42. The second 0.69 fitness chromosome is mutated resulting into a fitness of 0.37 providing clear evidence of the need of the proposed fractional adaptive mutation technique. A lower mutation rate at Generation 13 would have minimised the chances of 0.69 fitness chromosome getting lost as a 0.37-fitness candidate. The underlying rationale therefore is to improve the overall fitness of the chromosomes by reducing the probability by which good candidates are selected for mutation.



		Generations																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Chromosomes	1	0.55	0.56	0.08	0.25	0.41	0.26	0.54	0.42	0.06	0.31	0.53	0.44	0.24	0.56	0.47	0.32	0.23	0.55	0.27	0.52
	2	0.34	0.32	0.53	0.44	0.45	0.04	0.62	0.44	0.22	0.27	0.61	0.09	0.57	0.44	0.03	0.03	0.27	0.34	0.39	0.48
	3	0.21	0.03	0.56	0.20	0.61	0.46	0.33	0.50	0.34	0.64	0.36	0.45	0.09	0.61	0.27	0.03	0.06	0.30	0.47	0.62
	4	0.14	0.55	0.20	0.46	0.45	0.29	0.28	0.58	0.17	0.55	0.53	0.24	0.25	0.29	0.08	0.17	0.29	0.27	0.21	0.46
	5	0.03	0.03	0.36	0.08	0.61	0.42	0.12	0.09	0.06	0.39	0.66	0.28	0.40	0.59	0.07	0.50	0.39	0.50	0.13	0.50
	6	0.36	0.14	0.02	0.01	0.09	0.50	0.60	0.12	0.31	0.52	0.68	0.10	0.69	0.25	0.11	0.21	0.44	0.35	0.16	0.04
	7	0.37	0.38	0.51	0.02	0.29	0.35	0.37	0.43	0.17	0.52	0.23	0.19	0.07	0.17	0.59	0.11	0.26	0.18	0.08	0.57
	8	0.06	0.10	0.58	0.35	0.24	0.06	0.30	0.33	0.50	0.42	0.38	0.23	0.68	0.20	0.25	0.63	0.09	0.63	0.28	0.43
	9	0.34	0.54	0.06	0.41	0.20	0.53	0.58	0.48	0.41	0.58	0.36	0.51	0.42	0.53	0.50	0.50	0.03	0.21	0.36	0.03
	10	0.10	0.38	0.54	0.08	0.48	0.41	0.51	0.28	0.42	0.27	0.02	0.10	0.39	0.26	0.10	0.05	0.24	0.47	0.41	0.08
	11	0.35	0.37	0.18	0.58	0.13	0.48	0.33	0.65	0.41	0.50	0.10	0.19	0.23	0.05	0.15	0.61	0.56	0.54	0.57	0.52
	12	0.08	0.25	0.19	0.11	0.45	0.63	0.66	0.54	0.09	0.17	0.21	0.69	0.37	0.25	0.38	0.23	0.34	0.28	0.28	0.45
	13	0.57	0.17	0.52	0.08	0.37	0.48	0.01	0.09	0.56	0.08	0.43	0.52	0.02	0.28	0.24	0.18	0.46	0.07	0.56	0.33
	14	0.59	0.01	0.61	0.50	0.19	0.63	0.65	0.60	0.32	0.60	0.69	0.40	0.10	0.57	0.29	0.13	0.42	0.55	0.29	0.14
	15	0.04	0.17	0.15	0.24	0.13	0.33	0.60	0.08	0.25	0.50	0.41	0.06	0.39	0.11	0.34	0.20	0.18	0.58	0.27	0.04
	16	0.10	0.28	0.17	0.53	0.00	0.62	0.58	0.55	0.31	0.63	0.30	0.44	0.46	0.23	0.36	0.14	0.24	0.09	0.54	0.24
	17	0.19	0.52	0.42	0.17	0.21	0.40	0.23	0.64	0.61	0.09	0.16	0.57	0.10	0.38	0.21	0.08	0.40	0.08	0.12	0.32
	18	0.35	0.50	0.28	0.37	0.01	0.13	0.11	0.06	0.36	0.50	0.60	0.09	0.09	0.38	0.62	0.53	0.27	0.37	0.18	0.10
	19	0.50	0.09	0.49	0.37	0.28	0.50	0.56	0.13	0.37	0.37	0.50	0.27	0.04	0.27	0.28	0.56	0.33	0.40	0.63	0.55
	20	0.22	0.35	0.47	0.05	0.53	0.04	0.44	0.41	0.58	0.61	0.29	0.07	0.09	0.35	0.17	0.24	0.14	0.29	0.21	0.10
	21	0.02	0.56	0.33	0.57	0.16	0.43	0.07	0.31	0.66	0.11	0.21	0.17	0.11	0.06	0.20	0.39	0.20	0.03	0.46	0.03
	22	0.42	0.10	0.35	0.28	0.09	0.01	0.12	0.51	0.25	0.33	0.07	0.13	0.00	0.33	0.41	0.20	0.11	0.34	0.06	0.27
	23	0.56	0.16	0.57	0.46	0.12	0.24	0.10	0.43	0.37	0.37	0.57	0.24	0.37	0.27	0.50	0.38	0.11	0.40	0.46	0.35
	24	0.07	0.02	0.53	0.24	0.06	0.61	0.61	0.04	0.05	0.25	0.24	0.07	0.25	0.53	0.61	0.07	0.35	0.45	0.38	0.28
	25	0.56	0.34	0.07	0.37	0.53	0.51	0.28	0.20	0.09	0.26	0.49	0.39	0.31	0.31	0.35	0.27	0.08	0.27	0.06	0.29
	26	0.06	0.56	0.60	0.05	0.48	0.31	0.26	0.28	0.51	0.66	0.19	0.06	0.51	0.00	0.21	0.47	0.51	0.54	0.13	0.06
	27	0.47	0.38	0.50	0.35	0.34	0.39	0.13	0.44	0.62	0.42	0.16	0.16	0.35	0.50	0.26	0.13	0.19	0.29	0.59	0.57
	28	0.28	0.06	0.25	0.55	0.63	0.51	0.00	0.33	0.33	0.47	0.25	0.69	0.23	0.33	0.52	0.48	0.34	0.04	0.55	0.59
	29	0.16	0.11	0.06	0.31	0.35	0.61	0.26	0.52	0.50	0.24	0.45	0.30	0.00	0.13	0.08	0.46	0.24	0.10	0.25	0.21
	30	0.53	0.48	0.11	0.50	0.31	0.37	0.15	0.31	0.08	0.08	0.25	0.11	0.16	0.55	0.33	0.09	0.46	0.34	0.58	0.41
	31	0.48	0.10	0.22	0.29	0.14	0.46	0.63	0.03	0.05	0.60	0.45	0.68	0.39	0.53	0.40	0.34	0.29	0.08	0.29	0.27
	32	0.52	0.14	0.58	0.50	0.21	0.15	0.43	0.41	0.22	0.51	0.06	0.29	0.53	0.09	0.10	0.07	0.46	0.46	0.09	0.44
	33	0.29	0.60	0.60	0.22	0.32	0.35	0.35	0.66	0.59	0.57	0.57	0.08	0.18	0.52	0.63	0.56	0.13	0.44	0.25	0.40
	34	0.39	0.20	0.08	0.35	0.13	0.41	0.56	0.50	0.38	0.59	0.56	0.33	0.51	0.10	0.50	0.42	0.37	0.53	0.09	0.38
	35	0.06	0.22	0.22	0.05	0.55	0.20	0.09	0.49	0.48	0.44	0.32	0.37	0.44	0.43	0.62	0.47	0.09	0.28	0.52	0.09
	36	0.14	0.14	0.56	0.06	0.39	0.40	0.11	0.20	0.57	0.37	0.13	0.65	0.51	0.39	0.35	0.10	0.41	0.44	0.10	0.62
	37	0.23	0.17	0.21	0.18	0.12	0.48	0.58	0.23	0.04	0.42	0.01	0.53	0.50	0.55	0.24	0.42	0.07	0.44	0.03	0.51
	38	0.47	0.59	0.34	0.61	0.37	0.53	0.50	0.09	0.53	0.05	0.33	0.57	0.12	0.22	0.12	0.43	0.40	0.47	0.08	0.58
	39	0.28	0.10	0.60	0.33	0.43	0.09	0.52	0.35	0.24	0.03	0.65	0.18	0.39	0.01	0.13	0.53	0.35	0.26	0.09	0.03
	40	0.29	0.06	0.26	0.08	0.50	0.29	0.01	0.11	0.50	0.62	0.24	0.68	0.06	0.19	0.22	0.07	0.63	0.60	0.03	0.64

Table 15: Impact of constant mutation rate on fittest chromosomes (See generation 13)

6.3 Statistical analysis of datasets via non-parametric Two-sample Kolmogorov-Smirnov test

The Two-sample Kolmogorov-Smirnov test is used to non-parametrically measure the difference between two one-dimensional probability distribution. The test was used to compare if the actual and predicted values for the four datasets came from the same probability distribution and can be statistically defined as:

$$D_{n,n'} = \overset{sup}{x} \left|F_{1,n}(x) - F_{2,n'}(x)\right| \qquad (21)$$

In Equation (21), $F_{1,n}(x)$ and $F_{2,n'}(x)$ can be defined as respective distributive functions of the actual and predicted samples from the classification model and sup is the supremum function. The null hypothesis (of not originating from the same probability distribution) can therefore be rejected at level alpha if:

$$D_{n,n'} > c(\alpha)\sqrt{\frac{n+n'}{nn'}} \qquad (22)$$

In Table 16, small D-stat values indicate the samples to belong to distributions following similar probabilistic distribution. However, the p-value for HSBC and RBS indicate substantial difference in actual and predicted outcomes which can be attributed to the high level of noise encountered in the dataset.

	BP	GSK	HSBC	RBS	Yahoo
Alpha	0.05	0.05	0.05	0.05	0.05
D-stat	0.000801	0.000255	0.000962	0.000962	0.00096
p-value	0.22171	0.859958	2.74E-09	2.74E-09	1
D-crit	0.001037	0.000573	0.000409	0.000409	0.006174
size 1	3427915	11225730	22054895	22054895	96270
size 2	3428664	11223390	22055079	22055079	97077

Table 16: Two-sample Kolmogorov-Smirnov test for the four stock datasets to measure the measure of one-dimensional probability distribution of data.

To conclude, the general improvement in the rate of adaptive mutation with other models proves the hypothetical argument at the beginning of this research that fractional improvement is more likely to retain better individuals. Furthermore, the per-generation improvement rate with Elitism proves that the technique will work well with any other category of datasets. Therefore, the underlying rationale is to improve the overall fitness of the chromosomes by reducing the probability through which good candidates are selected for mutation.

7 Chapter7: Conclusions and Future Directions

The dissertation covered an in-depth review of statistical and AI techniques and critically analysed the current state-of-the-art in the financial and particularly stock market sector. Based on the review and existing research gap in time-series analysis of uncertain stock market shares, the work presented a novel GEP methodology which uses a direct way of optimisation (forecasting) with no need to integrate with other AI models, and obtained even better results comparing to the integrated ANFIS model presented earlier The evolutionary computing technique eventually gave way to a GEP-based genetic computing algorithm (GEP-FAMR) whose outcome was analysed in the previous chapter.

7.1 Conclusions

7.1.1 Application of the Neuro-Fuzzy paradigm in the financial time-series domain

The research covered under this thesis primarily analysed the FIS paradigm which is commonly used to integrate knowledge via expert engineers to model a domain whose boundaries are not clearly known by means of a set of manually designed rules. The subsequent tests revealed the usage of baseline FIS due to its inherent reliance on human-guidance which gets tedious with the increasing complexity of systems similar to stock market prediction. That is, for massively complex models such as time-series-based prediction systems, manual calibration of a fuzzy rule-base was found to be an immensely tedious task. This challenge was addressed by using sample input-output datasets to the membership functions via ANN-based training generally termed as Adaptive Network-based Fuzzy Inference System (ANFIS).

7.1.2 Research Contributions to the Neuro-Fuzzy Paradigm Application

This stage presented a critical analysis into various machine learning methodologies against time-series-based forecasting systems. This sub-model particularly evaluated the most commonly employed soft-computing paradigms that including fuzzy logic and neural networks. An in-depth analysis of the current state-of-the-art introduced significant potential in the utilisation of hybridised classification systems.

The proposed approach utilised the generalisation capabilities of neural networks to improve the automated rule-generation capability of the ANFIS framework. The approach utilised data from YHOO stock data to train a 10-day-delay back-propagation algorithm that converged with a very promising value of greater than 0.8. The large dataset generated by YHOO contained a total of 4281 days comprising of an estimated 11 years. In order to evaluate the overall consistency of reporting, the proposed technique employed two data evaluation techniques which presented a rounded identification accuracy of 90 and 86% with k-fold and Jack-knife validation respectively.

7.1.3 Role of Evolutionary GEP and Extensibility

The results of this phase presented promising outcomes specifically its ability to withstand high noise levels.

This research presented a novel evolutionary methodology with modified selection operators for the prediction of stock exchange data via a specialised extension to the conventional evolutionary GA. The extension dynamically adapts the mutation rate of different fitness groups in each generation to ensure a diversification balance between high and low fitness solutions.

7.1.4 Study Contributions on the Role of Evolutionary GEP and Extensibility

The proposed GEP-FAMR approach was adopted in comparison to Neural and Fuzzy approaches to address well-reported problems of over-fitting, algorithmic black-boxing, and data-snooping issues via GP and GEP algorithms. Another issue analysed in this case was the suitability of the algorithms to predict daily and weekly stock market patterns based upon medium-to-short-term stock history. The outcomes presented an outstanding GEP accuracy compared to GP in stock prediction via simple, non-arithmetic algebraic expressions with the best performance reported at short-term forecasting of 95.992%. On the other side, the worst performance of 88.23% was reported on a GP algorithm at medium-term prediction.

7.2 Limitations and Areas of Further Research

7.2.1 Limitations and Areas of Further Research(ANFIS)

Despite the promising prediction outcome, the accuracy achieved by using the proposed technique is not optimal. Further research should be conducted to determine ways of improving the technique such as through a varied number of neurons, activation functions, training algorithm types, number of neurons and the induced training delay. It was envisaged that an improvement in these values can be brought-in via a number of existing optimisation techniques. As discussed in the literature review, genetic algorithms, particle swarm optimisation, tabu-search and other similar optimisation algorithms can be employed to induce an automated, hill-climbing heuristic for the methodology to further improve the system outcome.

7.2.2 Limitations and Areas of Further Research(GEP)

This stage was also found to be reliant on the amount of stock market data available for training which was substantially limited. This compromised the ability of conventional ANN systems to generalise time-series on their own. At this stage, the research subsequently focussed on addressing this noise problem by optimising the underlying neuro-fuzzy training process by means of an evolutionary genetic algorithm. Furthermore, the worst performance mentioned above can be attributed to the fact the at 65-day-training-lengths, the algorithms routinely found data subjected to non-deterministic human-level changes because of global financial uncertainties such as the Middle East crisis and the global recession. These changes add to the study's limitation. However, GEP can particularly be used to increase the overall confidence level of trading markets if it is properly evaluated for the period of a few months before being implemented in risky stock markets for behaviour prediction. Furthermore, the GEP technique is reliant on the amount of stock market data available for training; which was substantially limited. Future research should be conducted to investigate ways of improving the number of neurons, activation functions, training algorithm types, and the induced training delay. An improvement in these values and the system outcome can be achieved through employing genetic algorithms, particle swarm optimisation, tabu-search among other optimisation algorithms.

8 Chapter8: References

Abdullah S. and Turabieh, H. (2008) "Generating University Course Timetable Using Genetic Algorithms and Local Search," in *Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on*, 2008, pp. 254-260.

Ahn, H., Ahn, J.J., Oh, K.J. and Kim, D.H. (2011) Facilitating cross-selling in a mobile telecom market to develop customer classification model based on hybrid data mining techniques. *Expert Systems with Applications*, **38**(5), pp. 5005-5012.

Akaike, H. (1986) Use of statistical models for time series analysis. Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86., 1986, Apr 1986. pp. 3147-3155.

Aladag C. H., Yolcu U., Egrioglu E., Bas E. (2014) Fuzzy lagged variable selection in fuzzy time series with genetic algorithms, Applied Soft Computing, Available online 29 April 2014, ISSN 1568-4946, http://dx.doi.org/10.1016/j.asoc.2014.03.028.

Al Tarawneh H.Y. and Ayob M. (2011) "Using Tabu search with multi-neighborhood structures to solve University Course Timetable UKM case study (faculty of engineering)," in *Data Mining and Optimization (DMO), 2011 3rd Conference on*, 2011, pp. 208-212.

Aladag C.H., Yolcu U., Egrioglu E., and Bas E. (2014) Fuzzy lagged variable selection in fuzzy time series with genetic algorithms, Applied Soft Computing, Available online 29 April 2014, ISSN 1568-4946, http://dx.doi.org/10.1016/j.asoc.2014.03.028.

Ambler, S.W. and Lines, M. (2012). *Disciplined agile delivery: A practitioner's guide to agile software delivery in the enterprise*. IBM Press.

ANFIS (2013) [Online] Adaptive Network Based Fuzzy Inference System, Available at http://www.bindichen.co.uk/post/AI/adaptive-neuro-fuzzy-inference-systems.html [Accessed 10/09/2014]

Ang, K.K. and Quek, C. (2006) Stock trading using RSPOP: A novel rough set-based neuro-fuzzy approach. *Ieee Transactions on Neural Networks*, 17(5), pp. 1301-1315.

Applegate, D.L. (2006) *The traveling salesman problem : a computational study*. Princeton ; Oxford: Princeton University Press, 2006.

Araújo R.A. and Ferreira T.A.E. (2009) An intelligent hybrid morphological-rank-linear method for financial time series prediction, Neurocomputing, Volume 72, Issues 10–12, June 2009, Pages 2507-2524, ISSN 0925-2312, http://dx.doi.org/10.1016/j.neucom.2008.11.008.

Atsalakis, G.S. and Valavanis, K.P. (2009) Forecasting stock market short-term trends using a neuro-fuzzy based methodology. *Expert Systems with Applications*, 36(7), pp. 10696-10707.

Baker, J.E. (1985) Adaptive selection methods for genetic algorithms In: Proceedings of the First International Conference on Genetic Algorithms and their Applications. Hillsdale, NJ: Lawrence Erlbaum, 1985. pp. 101–111.

Balaji, S. and Murugaiyan, M.S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management*, *2*(1), 26-30.

Balakrishnan, N. (2010) Methods and applications of statistics in business, finance, and management science. Hoboken, N.J.: Wiley.

Baragona, R., Battaglia, F. and Calzini, C. (2001) Genetic algorithms for the identification of additive and innovation outliers in time series. *Computational Statistics & Data Analysis*, **37**(1).

Barbulescu A.; Bautu E. (2012) A Hybrid Approach for Modeling Financial Time Series, The International Arab Journal of Information Technology, Vol. 9, No. 4, July 2012

Bautu, E., Bautu, A., Luchian H. (2010) Evolving Gene Expression Programming Regressions for Ensemble Prediction of Movements on the Stock Market

Ben G.M. and Wang, P.P. (2000) Intelligent system to support judgmental business forecasting: the case of estimating hotel room demand. **Fuzzy Systems, IEEE Transactions on,** v. 8, n. 4, p. 380-397, 2000. ISSN 1063-6706.

Bhatt V. and Sahajpal R. (2004) "Lecture timetabling using hybrid genetic algorithms," in *Intelligent Sensing and Information Processing, 2004. Proceedings of International Conference on*, 2004, pp. 29-34.

Bisoi, R., Dash, P.K., Padhee, V., and Naeem, M.H. (2011) "Mining of electricity prices in energy markets using a computationally efficient neural network," *Energy, Automation, and Signal (ICEAS), 2011 International Conference on* , vol., no., pp. 1,5, 28-30 Dec. 2011doi: 10.1109/ICEAS.2011.6147178

Blandis, E. and Simutis, R. (2002) Using Principal Component Analysis and Neural Network for Forecasting of Stock Market Index. Bizinesa augstskola Turiba SIA, Riga, 2002, 31-35.

Boo, J. (2007) "Stock Price forecasting using PSO-trained neural networks," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, 2007, pp. 2879-2885.

Bordino, I., Kourtellis, N., Laptev, N., and Billawala, Y. (2014) "Stock trade volume prediction with Yahoo Finance user browsing behavior," *Data Engineering (ICDE), 2014 IEEE 30th International Conference on* , vol., no., pp. 1168,1173, March 31 2014-April 4 2014

Bouktif, S. and Awad, M.A. (2013) "Ant colony based approach to predict stock market movement from mood collected on Twitter," in *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, 2013, pp. 837-845.

BP (2014) British Petroleum Plc [Online] Available at <https://uk.finance.yahoo.com/q/hp?a=&b=&c=&d=6&e=3&f=2014&g=d&s=BP&ql=1> [Accessed: 20/06/2014]

Burke, E.K. and Newall, J.P. (1999), "A multistage evolutionary algorithm for the timetable problem," *Evolutionary Computation, IEEE Transactions on,* vol. 3, pp. 63-74, 1999.

Cai, Q., Zhang, D., Wu, B., and Leung, S.C.H. (2013) A Novel Stock Forecasting Model based on Fuzzy Time Series and Genetic Algorithm, Procedia Computer Science, Volume 18, 2013, Pages 1155-1162, ISSN 1877-0509, http://dx.doi.org/10.1016/j.procs.2013.05.281.

Carlos, J., Garcia, F., and Mendez, J.J.S. (2007) "A Comparison of ANFIS, ANN and DBR systems on volatile Time Series Identification," Fuzzy Information Processing Society, 2007. NAFIPS '07. Annual Meeting of the North American, vol., no., pp. 319,324, 24-27 June, doi: 10.1109/NAFIPS.2007.383858

Castillo, O. and Melin, P. (2002) Hybrid intelligent systems for time series prediction using neural networks, fuzzy logic, and fractal theory. *Ieee Transactions on Neural Networks*, **13**(6), pp. 1395-1408.

Chang, P.-C., Liu, C.-H. and Fan, C.-Y. (2009) Data clustering and fuzzy neural network for sales forecasting: A case study in printed circuit board industry. *Knowledge-Based Systems*, **22**(5), pp. 344-355.

Chen, C.-F., Lai, M.-C. and Yeh, C.-C. (2012) Forecasting tourism demand based on empirical mode decomposition and neural network. *Knowledge-Based Systems*, **26**, pp. 281-287.

Chen, F.L. and Ou, T.Y. (2009) Gray relation analysis and multilayer functional link network sales forecasting model for perishable food in convenience store. *Expert Systems with Applications*, **36**(3), pp. 7054-7063.

Chen, F.L. and Ou, T.Y. (2011) Sales forecasting system based on Gray extreme learning machine with Taguchi method in retail industry. *Expert Systems with Applications*, **38**(3), pp. 1336-1345.

Chen, S.M. and Chung, N.Y. (2006) Forecasting enrollments using high-order fuzzy time series and genetic algorithms. *International Journal of Intelligent Systems*, **21**(5).

Chen, Y.H., Yang, B. and Dong, J.W. (2006) Time-series prediction using a local linear wavelet neural network. *Neurocomputing*, **69**(4-6), pp. 449-465.

Chen, Y.-S. (2012) Classifying credit ratings for Asian banks using integrating feature selection and the CPDA-based rough sets approach. *Knowledge-Based Systems*, **26**, pp. 259-270.

Cheng, C., Chen, T., and Wei, L. (2010) A hybrid model based on rough sets theory and genetic algorithms for stock price forecasting, Information Sciences, Volume 180, Issue 9, 1 May 2010, Pages 1610-1629, ISSN 0020-0255, http://dx.doi.org/10.1016/j.ins.2010.01.014.

Cheng, M. and Andreas, F.V.R. (2011) Evolutionary fuzzy decision model for cash flow prediction using time-dependent support vector machines, International Journal of Project Management, Volume 29, Issue 1, January 2011, Pages 56-65, ISSN 0263-7863.

Chiu, S. (1994) A cluster extension method with extension to fuzzy model identification. Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on, 1994, 26-29 Jun 1994. pp. 1240-1245 vol.2.

Chuang, L.-Y., Hsiao, C.-J. and Yang, C.-H. (2011) Chaotic particle swarm optimization for data clustering. *Expert Systems with Applications*, **38**(12), pp. 14555-14563.

Connor, J.T., Martin, R.D. and Atlas, L.E. (1994) Recurrent neural networks and robust time-series prediction. IEEE Transactions on Neural Networks, 5(2), pp. 240-254.

Daníelsson, J. (2011). *Financial risk forecasting: the theory and practice of forecasting market risk with implementation in R and Matlab* (Vol. 588). John Wiley & Sons.

Darwin, C. (2003) *On the origin of species by means of natural selection, or, The preservation of favoured races in the struggle for life*. Holicong, PA: Wildside Press, 2003.

Davis, W.S. and Yen, D.C. (1999) *The information system consultant's handbook : systems analysis and design.* Boca Raton, Fla. ; London: CRC Press.

Dazhuo, Z., Jinxia, L., and Wenxiu, M. (2009) Clustering Based on LLE For Financial Multivariate Time Series. Management and Service Science, 2009. MASS '09. International Conference on, 2009, 20-22 Sept. 2009. pp. 1-4.

Debbabi, M., Hassaine, F., Jarraya, Y., Soeanu, A., and Alawneh, L. (2010).*Verification and validation in systems engineering: assessing UML/SysML design models*. Berlin: Springer Science & Business Media.

Devi, B.U., Sundar, D., and Alli, P., (2013) "An optimized approach to predict the stock market behavior and investment decision making using benchmark algorithms for Naïve investors," *Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on* , vol., no., pp. 1,5, 26-28 Dec. 2013

Dipti, S., Tian Hou, S., and Jian-Xin, X. (2002) "Automated time table generation using multiple context reasoning for university modules," in *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, 2002, pp. 1751-1756.

Duan, L., Tang, C., Gou, C., Jiang, M., and Zuo, J. (2011). Mining good sliding window for positive pathogens prediction in pathogenic spectrum analysis. In *Advanced Data Mining and Applications* (pp. 152-165). Springer Berlin Heidelberg.

Eads, D., Hill, D., Davis, S., Perkins, S., Ma, J.S., Porter, R. and Theiler, J. (2002) Genetic algorithms and support vector machines for time series classification. *Conference on Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation V*, **4787**. Available from <<Go to ISI>://WOS:000180353600008>. ISSN 0277-786X 0-8194-4554-1.

Encarnacao, J.L., Lindner, R., and Schlechtendahl, E.G. (2012). *Computer aided design: fundamentals and system architectures*. Springer Science & Business Media.

Esfahanipour, A. and Aghamiri, W. (2010) Adapted Neuro-Fuzzy Inference System on indirect approach TSK fuzzy rule base for stock market analysis. *Expert Systems with Applications*, 37(7), pp. 4742-4748.

Fenton, N. and Bieman, J. (2014). *Software metrics: a rigorous and practical approach*. CRC Press.

Ferreira. C. (2001) Gene Expression Programming: A New Adaptive Algorithm for Solving Problems, Complex Systems, Vol. 13 Issue 2.

Fox, J. (2011). *The myth of the rational market: a history of risk, reward, and delusion on Wall Street*. Harriman House Limited.

Gao, W. (2008), "New Neural Network Based on Ant Colony Algorithm for Financial Data Forecasting," in *Intelligent Information Hiding and Multimedia Signal Processing, 2008. IIHMSP '08 International Conference on*, 2008, pp. 1437-1440.

Garg, S. Sriram, K. Tai (2013) Empirical Analysis of Model Selection Criteria for Genetic Programming in Modeling of Time Series System.

Gen, M. and Cheng, R. (2000) *Genetic algorithms and engineering optimization*. New York ; Chichester: Wiley, 2000.

Genetic Algorithms Overview . Available: http://geneticalgorithms.ai-depot.com/Tutorial/Overview.html [Accessed: 8/2/2017].

Ghaemi, S., Vakili, M.T., and Aghagolzadeh, A. (2007) "Using a genetic algorithm optimizer tool to solve University timetable scheduling problem," in *Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on*, 2007, pp. 1-4.

Gibson, R. (2013). *Asset Allocation: Balancing Financial Risk: Balancing Financial Risk*. McGraw Hill Professional.

Goh, C.-K., Tan, K.C. and Springerlink (2009) *Evolutionary multi-objective optimization in uncertain environments [electronic resource] : issues and algorithms.* Berlin: Springer-Verlag.

Gorans, P. and Kruchten, P. (2014). *A guide to critical success factors in agile delivery.* IBM Center for the Business of Government.

Gordini, N. (2014) A genetic algorithm approach for SMEs bankruptcy prediction: Empirical evidence from Italy, Expert Systems with Applications, Volume 41, Issue 14, 15 October 2014, Pages 6433-6445, ISSN 0957-4174,

http://dx.doi.org/10.1016/j.eswa.2014.04.026.
(http://www.sciencedirect.com/science/article/pii/S0957417414002486)

Gould, D.A.D. and Sciencedirect (2005) *Complete Maya programming. : an in-depth guide to 3D fundamentals, geometry, and modeling.* San Francisco, CA: Morgan Kaufmann Publishers.

GSK (2014) Glaxo Smith Klien Plc [Online] Available at < https://uk.finance.yahoo.com/q/hp?a=&b=&c=&d=6&e=3&f=2014&g=d&s=gsk&ql=1 > [Accessed: 20/06/2014]

Grosan C, Abraham A. (2006) Crina Grosan1 and Ajith Abraham2 () Stock Market Modeling Using Genetic Programming Ensembles, Genetic Systems Programming: Theory and Experiences, pp. 131-146.

Han, M., Xi, J.H., Xu, S.G. and Yin, F.L. (2004) Prediction of chaotic time series based on the recurrent predictor neural network. *Ieee Transactions on Signal Processing*, **52**(12), pp. 3409-3416.

Hewahi, N.M. (2015). Particle Swarm Optimization For Hidden Markov Model. *International Journal of Knowledge and Systems Science (IJKSS)*,*6*(2), 1-15.

Hiemstra, Y. (1994) A stock market forecasting support system based on fuzzy logic. System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on, 1994, 4-7 Jan. 1994. pp. 281-287.

Highsmith, J. (2013). *Adaptive software development: a collaborative approach to managing complex systems*. Addison-Wesley.

Hongbin, W., Liyi, Z., Haukui, W. (2010) The Research on Neural Network Prediction based on the GEP, Second International Workshop on Education Technology and Computer Science.

Holland, J.H. (1975) *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence.* Ann Arbor, Mich.: University of Michigan Press.

Hong, W., Dong, Y., Chen, L., and Wei, S. (2011) SVR with hybrid chaotic genetic algorithms for tourism demand forecasting, Applied Soft Computing, Volume 11, Issue 2, March 2011, Pages 1881-1890, ISSN 1568-4946, http://dx.doi.org/10.1016/j.asoc.2010.06.003.

HSBA (2014) HSBC Holdings Plc [Online] Available at https://uk.finance.yahoo.com/q/hp?a=&b=&c=&d=6&e=3&f=2014&g=d&s=hsba&ql=1 [Accessed: 12/06/2014]

Hsien-Kai, H., En-Jui, C., Chih-Hao, C., Shu-Yen, L. and An-Yeu, W. (2011) "Multi-Pheromone ACO-based routing in Network-on-Chip system inspired by economic phenomenon," in *SOC Conference (SOCC), 2011 IEEE International*, 2011, pp. 273-277.

Huang, K.Y. (2009) Application of VPRS model with enhanced threshold parameter selection mechanism to automatic stock market forecasting and portfolio selection. *Expert Systems with Applications*, **36**(9), pp. 11652-11661.

Huang, K.Y. and Jane, C.-J. (2009) A hybrid model for stock market forecasting and portfolio selection based on ARX, grey system and RS theories. *Expert Systems with Applications*, **36**(3), pp. 5387-5392.

Huang, K., Qi, Z., and Liu, B. (2009) Network Anomaly Detection Based on Statistical Approach and Time Series Analysis. Advanced Information Networking and Applications Workshops, 2009. WAINA '09. International Conference on, 2009, 26-29 May 2009. pp. 205-211.

Huggins, R. and Izushi, H. (Eds.). (2011). *Competition, competitive advantage, and clusters: the ideas of Michael Porter*. OUP Oxford.

Hung, J.-C. (2008) A genetic algorithm approach to the spectral estimation of time series with noise and missed observations. *Information Sciences*, **178**(24).

Ishibuchi, H. and Nakashima, T. (1998) A study on generating fuzzy classification rules using histograms. Knowledge-Based Intelligent Electronic Systems, 1998. Proceedings

KES '98. 1998 Second International Conference on, 1998, 21-23 Apr 1998. pp. 132-140 vol.1.

JaeHung, Y. and Sethi, I.K. (1995) Design of radial basis function networks using decision trees. Neural Networks, 1995. Proceedings., IEEE International Conference on, 1995, Nov/Dec 1995. pp. 1269-1272 vol.3.

Karova, M. (2004) "Solving timetabling problems using genetic algorithms," in *Electronics Technology: Meeting the Challenges of Electronics Technology Progress, 2004. 27th International Spring Seminar on*, 2004, pp. 96-98 vol.1.

Keyes, J. (2000) *Financial services information systems.* Boca Raton: Auerbach.

Khang, N., Khon, T. and Nuong, T. (2010) "Automating a Real-World University Timetabling Problem with Tabu Search Algorithm," in *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2010 IEEE RIVF International Conference on*, 2010, pp. 1-6.

Kim, K. and Han, I. (2000) Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index, Expert Systems with Applications, Volume 19, Issue 2, August 2000, Pages 125-132, ISSN 0957-4174, http://dx.doi.org/10.1016/S0957-4174(00)00027-0.

Kingdon, J. (2012). *Intelligent systems and financial forecasting*. Berlin: Springer Science & Business Media.

Kocadağlı O., Aşıkgil B. (2014) Nonlinear time series forecasting with Bayesian neural networks, Expert Systems with Applications, Volume 41, Issue 15, 1 November 2014, Pages 6596-6610, ISSN 0957-4174, http://dx.doi.org/10.1016/j.eswa.2014.04.035. (http://www.sciencedirect.com/science/article/pii/S0957417414002589)

Koza, J. R. (1992) *Genetic programming : on the programming of computers by means of natural selection*. Cambridge, Mass. ; London: MIT, 1992.

Kozma, R. (1994) Anomaly detection by neural network models and statistical time series analysis. Neural Networks, 1994. IEEE World Congress on Computational

Intelligence., 1994 IEEE International Conference on, 1994, 27 Jun-2 Jul 1994. p.3207-3210 vol.5.

Kumar, D.A.; Murugan, S., (2013) "Performance analysis of Indian stock market index using neural network time series model," *Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on* , vol., no., pp.72,78, 21-22 Feb. 2013

Kumar, P. R. and Ravi, V. (2007) Bankruptcy prediction in banks and firms via statistical and intelligent techniques - A review. *European Journal of Operational Research*, **180**(1), pp.1-28.

Laha, D. and Mandal, P. (2008) *Handbook of computational intelligence in manufacturing and production management.* Hershey, Pa: Information Science Reference.

Laws and Statutes, E. (1999) *Data protection act 1998 : chapter 29*. Corrected reprint. ed. London: Stationery Office.

Lee, M. A.; Smith, M. H. (1995) Handling uncertainty in finance applications using soft computing. Uncertainty Modeling and Analysis, 1995, and Annual Conference of the North American Fuzzy Information Processing Society. Proceedings of ISUMA - NAFIPS '95., Third International Symposium on, 1995, 17-19 Sep 1995. p.384-389.

Leung, H., Lo, T. and Wang, S. C. (2001) Prediction of noisy chaotic time series using an optimal radial basis function neural network. *Ieee Transactions on Neural Networks*, **12**(5), pp.1163-1172.

Li, C., Zhou, J., Kou, P. and Xiao, J. (2012) A novel chaotic particle swarm optimization based fuzzy clustering algorithm. *Neurocomputing*, **83**, pp.98-109.

Li, Q.-Z.; Shi, W.-C. (2012) Research in financial risk prediction on biochemical industry of China listed companies. Management Science and Engineering (ICMSE), 2012 International Conference on, 2012, 20-22 Sept. 2012. p.1517-1521.

Li, X. H. (2005) Application of grey majorized model in tunnel surrounding rock displacement forecasting. **Advances in Natural Computation, Pt 2, Proceedings,** v. 3611, p. 584-591, 2005 2005. ISSN 0302-9743. Disponível em: < <Go to ISI>://WOS:000232222500083 >.

Liu, S. (2013). *Formal Engineering for Industrial Software Development: Using the SOFL Method*. Springer Science & Business Media.

Liu, S., Lin, Y. and Springerlink (2006) *Grey information [electronic resource] : theory and practical applications.* London: Springer.

Lopes, L. S., Portuguese Conference on Artificial, I., Lau, N., Rocha, L. M., Mariano, P. and Springerlink (2009) *Progress in Artificial Intelligence [electronic resource] : 14th Portuguese Conference on Artificial Intelligence, EPIA 2009, Aveiro, Portugal, October 12-15, 2009. Proceedings.* Berlin, Heidelberg: Springer Berlin Heidelberg.

Martinetz, T.M., Berkovich, S.G., and Schulten, K.J. (1993). Neural-gas' network for vector quantization and its application to time-series prediction. *Neural Networks, IEEE Transactions on*, *4*(4), 558-569.

Mathworks (2014a) Adaptive Neuro-Fuzzy Modelling, [Online] Available at http://www.mathworks.co.uk/help/fuzzy/what-is-sugeno-type-fuzzy-inference.html?nocookie=true [Accessed: 17/09/14]

Mathworks (2014b) Gaussian Mixture Models Clustering [Online] Available at http://www.mathworks.co.uk/help/stats/gmdistribution.cluster.html [Accessed: 17/09/14]

Mathworks (2014c) NARX Classification [Online] Available at < http://www.mathworks.co.uk/help/nnet/ug/design-time-series-narx-feedback-neural-networks.html> [Accessed: 20/10/2014]

Meng Tang; Koch, W.H. (2009) "An Intelligent Model Based on TS NARX for Process Prediction and Diagnosis Rule Extraction," Fuzzy Systems and Knowledge Discovery,

2009. FSKD '09. Sixth International Conference on , vol.6, no., pp.91,95, 14-16 Aug. 2009

Muller, H.A., Norman, R.J., and Slonim, J. (Eds.). (2012). *Computer Aided Software Engineering*. Springer Science & Business Media.

Neuhierl, A. and Schlusche, B. (2011). Data snooping and market-timing rule performance. *Journal of Financial Econometrics*, *9*(3), 550-587.

Nenortaite J. and Simutis R. (2005) "Adapting particle swarm optimization to stock markets," in *Intelligent Systems Design and Applications, 2005. ISDA '05. Proceedings. 5th International Conference on*, 2005, pp. 520-525.

Ni H., Wang Y. (2013) Stock index tracking by Pareto efficient genetic algorithm, Applied Soft Computing, Volume 13, Issue 12, December 2013, Pages 4519-4535, ISSN 1568-4946, http://dx.doi.org/10.1016/j.asoc.2013.08.012.

Oja, E.; Kiviluoto, K.; Malaroiu, S. (2000) Independent component analysis for financial time series. Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000, 2000, 2000. p.111-116.

Ollos, G.; Vida, R. (2011) Adaptive Event Forecasting in Wireless Sensor Networks. Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd, 2011, 15-18 May 2011. p.1-5.

Olderog, E. R. and Steffen, B. (1999) *Correct system design [electronic resource] : recent insights and advances.* New York: Springer.

Partsch, H.A. (2012). *Specification and transformation of programs: a formal approach to software development*. Springer Science & Business Media.

Pengying Du; Xiaoping Luo; Zhiming He; Liang Xie, (2010) "The application of Genetic Algorithm-Radial Basis Function (GA-RBF) Neural Network in stock forecasting," *Control and Decision Conference (CCDC), 2010 Chinese* , vol., no., pp.1745,1748, 26-28 May 2010

Potvin J., Soriano P., Vallée M. (2004) Generating trading rules on the stock markets with genetic programming, Computers & Operations Research, Volume 31, Issue 7, June 2004, Pages 1033-1047, ISSN 0305-0548, http://dx.doi.org/10.1016/S0305-0548(03)00063-7. (http://www.sciencedirect.com/science/article/pii/S0305054803000637)

Porter, M.E. (1980) Competitive Strategy, Free Press, New York, 1980.

Pulido M., Melin P., Castillo O. (2014) Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican Stock Exchange, Information Sciences, Volume 280, 1 October 2014, Pages 188-204, ISSN 0020-0255, http://dx.doi.org/10.1016/j.ins.2014.05.006.

Qiaolin, D.; Jing, T.; Jianxin, L. (2005) Application of new FCMAC neural network in power system marginal price forecasting. Power Engineering Conference, 2005. IPEC 2005. The 7th International, 2005, Nov. 29 2005-Dec. 2 2005. p.1-57.

Rachev, S. T. and Ebrary, I. (2008) *Bayesian methods in finance.* Hoboken, N.J.: Wiley.

RBS (2014) Royal Bank of Scotland [Online] Available at < https://uk.finance.yahoo.com/q/hp?a=&b=&c=&d=6&e=3&f=2014&g=d&s=rbs&ql=1 > [Accessed: 19/06/2014]

RegCal (2015) Regression Calculation, [Online] Article available online at <http://corr> Accessed 01/05/2015.

Refaeilzadeh, P., Tang, L. and Liu, H., 2009. Cross-validation. In *Encyclopedia of database systems* (pp. 532-538). Springer US.

Satchell, S., & Knight, J. (2011). *Forecasting volatility in the financial markets*. Butterworth-Heinemann.

Salvendy, G. and Institute of Industrial, E. (2001) *Handbook of industrial engineering : technology and operations management.* 3rd ed. New York ; Chichester: Wiley.

Seese, D. (2008) *Handbook on Information Technology in Finance [electronic resource].* Dordrecht: Springer-Verlag.

Shelly, G. B. and Rosenblatt, H. J. (2010) *Systems analysis and design.* Boston, Mass. ; United Kingdom: Course Technology.

Shinn-Ying, H. (2004) Design of accurate regressions with a compact fuzzy-rule base using an evolutionary scatter partition of feature space. **Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,** v. 34, n. 2, p. 1031-1044, 2004. ISSN 1083-4419.

Sigl B., Golub, M. and Mornar, V. (2003) "Solving timetable scheduling problem using genetic algorithms," in *Information Technology Interfaces, 2003. ITI 2003. Proceedings of the 25th International Conference on*, 2003, pp. 519-524.

Skolpadungket, P.; Dahal, K.; Harnpornchai, N., (2009) "Forecasting stock returns using variable selections with Genetic Algorithm and Artificial Neural-Networks," *Computational Intelligence and Industrial Applications, 2009. PACIIA 2009. Asia-Pacific Conference on* , vol.1, no., pp.186,189, 28-29 Nov. 2009
Smith C., Jin Y. (2014) Evolutionary Multi-Objective Generation of Recurrent Neural Network Ensembles for Time Series Prediction, Neurocomputing, Available online 12 June 2014, ISSN 0925-2312, http://dx.doi.org/10.1016/j.neucom.2014.05.062.

Song, Y.H. (Ed.). (2013). *Modern optimisation techniques in power systems* (Vol. 20). Springer Science & Business Media.

Soofi, A. S. and Cao, L. (2002) *Modelling and forecasting financial data : techniques of nonlinear dynamics.* Boston, Mass.: Kluwer Academic Pub.

Sourirajan K., Ozsen L., Uzsoy U. (2009), A genetic algorithm for a single product network design model with lead time and safety stock considerations, European Journal of Operational Research, Volume 197, Issue 2, 1 September 2009, Pages 599-608, ISSN 0377-2217, http://dx.doi.org/10.1016/j.ejor.2008.07.038.

Sovilj D., Sorjamaa A., Yu Q., Miche Y., Séverin E. (2010) OPELM and OPKNN in long-term prediction of time series using projected input data, Neurocomputing, Volume 73, Issues 10–12, June 2010, Pages 1976-1986, ISSN 0925-2312, http://dx.doi.org/10.1016/j.neucom.2009.11.033.
(http://www.sciencedirect.com/science/article/pii/S0925231210001086)

Stojanović M. B., Božić M. M., Stanković M. M., Stajić Z. P. (2014) A methodology for training set instance selection using mutual information in time series prediction, Neurocomputing, Volume 141, 2 October 2014, Pages 236-245, ISSN 0925-2312, http://dx.doi.org/10.1016/j.neucom.2014.03.006. (http://www.sciencedirect.com/science/article/pii/S092523121400410X)

Straßburg J., Gonzàlez-Martel C., Alexandrov V. (2012) Parallel genetic algorithms for stock market trading rules, Procedia Computer Science, Volume 9, 2012, Pages 1306-1313, ISSN 1877-0509, http://dx.doi.org/10.1016/j.procs.2012.04.143.

Szpiro, G. G. (1997) Forecasting chaotic time series with genetic algorithms. *Physical Review E*, **55**(3).

Tay, F. E. H.; Cao, L. J. (2001) Application of support vector machines in financial time series forecasting. **Omega-International Journal of Management Science,** v. 29, n. 4, p. 309-317, Aug 2001. ISSN 0305-0483. Disponível em: < <Go to ISI>://WOS:000169885000002 >.

Tang, T. C. and Chi, L. C. (2005) Predicting. multilateral trade credit risks: comparisons of Logit and Fuzzy Logic models using ROC curve analysis. *Expert Systems with Applications*, **28**(3), pp.547-556.

Tay, F. E. H., Shen, L. and Cao, L. (2003) *Ordinary shares, exotic methods : financial forecasting using data mining techniques.* River Edge, N.J.: World Scientific.

Tung, W. L., Quek, C. and Cheng, P. (2004) GenSo-EWS: a novel neural-fuzzy based early warning system for predicting bank failures. *Neural Networks*, **17**(4), pp.567-587.

Venkadesh S., Hoogenboom G. (2013) Walter Potter, Ronald McClendon, A genetic algorithm to refine input data selection for air temperature prediction using artificial neural networks, Applied Soft Computing, Volume 13, Issue 5, May 2013, Pages 2253-2260, ISSN 1568-4946, http://dx.doi.org/10.1016/j.asoc.2013.02.003.

Vasconcelos, J. A., Ramirez, J. A., Takahashi, R. H. C. and Saldanha R. R. (2001), "Improvements in genetic algorithms," *Magnetics, IEEE Transactions on,* vol. 37, pp. 3414-3417, 2001.

Wang, J., Zhu, S., Zhao, W. and Zhu, W. (2011) Optimal parameters estimation and input subset for grey model based on chaotic particle swarm optimization algorithm. *Expert Systems with Applications*, **38**(7), pp.8151-8158.

Wang, J.-J., Wang, J.-Z., Zhang, Z.-G. and Guo, S.-P. (2012) Stock index forecasting based on a hybrid model. *Omega-International Journal of Management Science*, **40**(6), pp.758-766.

Wang W. & Niu Z., (2009) "Time Series Analysis of NASDAQ Composite Based on Seasonal ARIMA Model," *Management and Service Science, 2009. MASS '09. International Conference on* , vol., no., pp.1,4, 20-22 Sept. 2009

Wang W., Guo Y.; Niu Z.; Cao Y., (2009) "Stock indices analysis based on ARMA-GARCH model," *Industrial Engineering and Engineering Management, 2009. IEEM 2009. IEEE International Conference on* , vol., no., pp.2143,2147, 8-11 Dec. 2009

Wei Shen; Mian Xing, (2009) "Stock Index Forecast with Back Propagation Neural Network Optimized by Genetic Algorithm," *Information and Computing Science, 2009. ICIC '09. Second International Conference on* , vol.2, no., pp.376,379, 21-22 May 2009

Wei, L.-Y., Chen, T.-L. and Ho, T.-H. (2011) A hybrid model based on adaptive-network-based fuzzy inference system to forecast Taiwan stock market. *Expert Systems with Applications*, **38**(11), pp.13625-13631.

Wei L. (2013) A GA-weighted ANFIS model based on multiple stock market volatility causality for TAIEX forecasting, Applied Soft Computing, Volume 13, Issue 2, February 2013, Pages 911-920, ISSN 1568-4946, http://dx.doi.org/10.1016/j.asoc.2012.08.048.

Weilkiens, T., Lamm, J., Roth, S., and Walker, M. (2015). *Model-Based System Architecture*. John Wiley & Sons.

Wong, B. K. and Selvi, Y. (1998) Neural network applications in finance: A review and analysis of literature (1990-1996). *Information & Management*, **34**(3), pp.129-139.

Whitley D. (1989) The GENITOR algorithm and selection pressure: why rank-basedallocation of reproductive trials is best. In: Proceedings of the Third International

Conference on Genetic Algorithms. San Mateo, CA: Morgan Kaufmann, 1989. p. 116–121.

Xiao-qin W. (2012) Research on Prediction Model of Time Series Based on Fuzzy Theory and Genetic Algorithm, Physics Procedia, Volume 33, 2012, Pages 1241-1247, ISSN 1875-3892, http://dx.doi.org/10.1016/j.phpro.2012.05.205.

Yadavalli, V. K., Dahule, R. K., Tambe, S. S. and Kulkarni, B. D. (1999) Obtaining functional form for chaotic time series evolution using genetic algorithm. *Chaos*, **9**(3).

Yahoo (2012) Historical Price Data [Online] Available at <http://finance.yahoo.com/q/hp?s=YHOO> [Accessed: 1st Sept, 2012]

Yahoo (2014) Yahoo Inc stock data YHOO [Online] Available at < https://uk.finance.yahoo.com/q/hp?s=YHOO> [Accessed: 20/06/2014]

Yan, H. and Ma, L. (2008) New Model of Price Index of Commodity Retail Forecasting System. *Recent Advance in Statistics Application and Related Areas, Pts 1 and 2*, pp.293-298.

Yang, C.-H., Hsiao, C.-J. and Chuang, L.-Y. (2009) Accelerated Chaotic Particle Swarm Optimization for Data Clustering. *Proceedings of 2009 International Conference on Machine Learning and Computing (Iacsit Icmlc 2009)*, pp.249-253.

Yan, W. and Clack, C.D. (2011). Evolving robust GP solutions for hedge fund stock selection in emerging markets. *Soft Computing*, *15*(1), 37-50.

Ye F., Mabu S., Wang L., Hirasawa K. (2009) Genetic Network Programming with General Individual Reconstruction ICROS-SICE International Joint Conference 2009 August 18-21, 2009, Fukuoka International Congress Center, Japan.

Yih-Jen, H. (2005) A new method for fuzzy information retrieval based on fuzzy hierarchical clustering and fuzzy inference techniques. **Fuzzy Systems, IEEE Transactions on,** v. 13, n. 2, p. 216-228, 2005. ISSN 1063-6706.

Yoshida, Y. (2003) The valuation of European options in uncertain environment. *European Journal of Operational Research*, **145**(1), pp.221-229.

Yuling, L.; Haixiang, G.; Jinglu, H. (2013) An SVM-based approach for stock market trend prediction. Neural Networks (IJCNN), The 2013 International Joint Conference on, 2013, 4-9 Aug. 2013. p.1-7.

Yusuf, S.A.; Brown, D.J.; Mackinnon, A.; Papanicolaou, R. (2013) "Application of dynamic neural networks with exogenous input to industrial conditional monitoring," Neural Networks (IJCNN), The 2013 International Joint Conference on , vol., no., pp.1,8, 4-9 Aug. 2013

Zadeh, L. A., Klir, G. J. and Yuan, B. (1996) *Fuzzy sets, fuzzy logic, and fuzzy systems : selected papers.* Singapore ; London: World Scientific.

Zeng, D. and Springerlink (2012) *Advances in information technology and industry applications [electronic resource].* Berlin: Springer.

Zhang X., Onieva E., Perallos A., Osaba E., Lee V.C.S. (2014) Hierarchical fuzzy rule-based system optimized with genetic algorithms for short term traffic congestion prediction, Transportation Research Part C: Emerging Technologies, Volume 43, Part 1, June 2014, Pages 127-142, ISSN 0968-090X, http://dx.doi.org/10.1016/j.trc.2014.02.013.

Zhang, G. P. and Berardi, V. L. (2001) Time series forecasting with neural network ensembles: an application for exchange rate prediction. *Journal of the Operational Research Society*, **52**(6), pp.652-664.

Zhu, D.; Wang, X.; Ren, R. (2010) A Heuristics R and D Projects Portfolio Selection Decision System Based on Data Mining and Fuzzy Logic. Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on, 2010, 11-12 May 2010. p.118-121.

Zhu, M. and Springerlink (2012) *Business, economics, financial sciences, and management [electronic resource].* International Conference on Business, Berlin ; New York: Springer.