

An Artificial Intelligence Approach to Financial Forecasting  
using Improved Data Representation, Multi-objective  
Optimization, and Text Mining.

by

Matthew Butler

Submitted in partial fulfillment of the requirements  
for the degree of Masters of Electronic Commerce

at

Dalhousie University  
Halifax, Nova Scotia  
August 2009

© Copyright by Matthew Butler, 2009

## Table of Contents

List of Tables .....	vii
List of Figures .....	viii
Abstract .....	ix
Acknowledgements .....	x
Chapter 1 - Introduction .....	1
1.1 Investment Theory .....	1
1.1.1 Efficient Market Hypothesis .....	2
1.1.2 Random Walk Hypothesis .....	3
1.2 Trading Strategies .....	3
1.2.1 Short-selling .....	4
1.3 Prediction Methodology .....	5
1.4 Research Objectives and Hypotheses .....	6
1.4.1 - Pseudo Financial Factor Modeling .....	7
1.4.2 – Multi-objective Optimization .....	7
1.4.3 – N-gram Analysis and Readability Scores of Annual Reports .....	8
1.5 - Organization of Thesis .....	8
Chapter 2 - Background and Related Work .....	9
2.1 Support Vector Machines .....	9
2.2 Genetic Programming .....	9
2.3 Evolutionary Neural Networks .....	10
2.4 Text Mining .....	11
Chapter 3 - Optimizing a Pseudo Financial Factor Model with Support Vector Machines and Genetic Programming .....	13
3.1 Introduction .....	13
3.2 Financial Factor Modeling .....	13
3.3 Data Description and Pre-processing .....	15
3.4 Methodology and Software Utilization .....	17
3.4.1 Genetic Programming Approach .....	17
3.4.2 Support Vector Machine Approach .....	19
3.5 Trading Strategy .....	23
3.6 Testing Results .....	23
3.7 Discussion and Conclusions .....	26
Chapter 4 - Multi-objective Optimization with an Evolutionary Artificial Neural Network for Financial Forecasting .....	27
4.1 Introduction .....	27

4.2 Algorithm.....	28
4.2.1 Parameter and Data-structure description .....	29
4.2.2 Initial Population.....	30
4.2.3 Crossover Operation .....	30
4.2.4 Mutation.....	31
4.2.5 Speciation.....	32
4.2.6 Fitness of an individual .....	33
4.3 Semi-active Trading System .....	33
4.3.1 Trading Strategy.....	33
4.3.2 Data Description .....	33
4.4 Multi-objective Optimization.....	34
4.5 Experiments and Results .....	35
4.5.1 Training Phase .....	35
4.5.2 Testing Phase .....	38
4.5.3 Results for Backpropagation.....	39
4.6 Discussion and Conclusions .....	40
Chapter 5 - Financial Forecasting Using Character N-gram Analysis and Readability Scores on Annual Reports .....	42
5.1 Introduction.....	42
5.2 Data Pre-processing .....	44
5.2.1 Data Collection .....	44
5.2.2 Data Labelling.....	44
5.2.3 Generating N-Gram Profiles .....	44
5.2.4 Generating Readability Scores.....	45
5.3 CNG Classification of N-Gram Profiles .....	46
5.4 SVM Classification with Readability Scores.....	46
5.4 Experiment Results .....	47
5.6 Model Results Comparisons .....	50
5.7 Discussion and Conclusions .....	51
5.7.1 N-Grams with CNG Classification .....	52
5.7.2 Readability Scores with SVM.....	52
5.7.3 Combined Models.....	53
5.7.4 The 2008 Investment Anomaly.....	53
5.8 Drawbacks and Limitations .....	54
Chapter 6 - Results Analysis and Discussion.....	56

6.1 – Pseudo financial factor modeling.....	56
6.2 – Multi-objective optimization of stock markets returns .....	57
6.3 – NLP based models .....	59
Chapter 7 - Conclusion .....	62
References.....	64

## List of Tables

<b>Table 1 - Input Attributes for the PFFM .....</b>	<b>16</b>
<b>Table 2 – Input Attribute Rankings for the PFFM .....</b>	<b>17</b>
<b>Table 3 – GP Parameters .....</b>	<b>19</b>
<b>Table 4 - SVM Parameters .....</b>	<b>22</b>
<b>Table 5 - Testing Results for PFFM. ....</b>	<b>24</b>
<b>Table 6 – EANN Parameter Settings .....</b>	<b>36</b>
<b>Table 7 – EANN Training Results .....</b>	<b>37</b>
<b>Table 8 – EANN Testing Results .....</b>	<b>38</b>
<b>Table 9 – Detailed Experiment Results for NLP Based Models .....</b>	<b>48</b>
<b>Table 10 – P-values for the pseudo financial factor models .....</b>	<b>57</b>
<b>Table 11 – EANN Training Results for Statistical Analysis.....</b>	<b>58</b>
<b>Table 12 – EANN Testing Results for Statistical Analysis.. ....</b>	<b>58</b>
<b>Table 13 - P-values for multi-objective optimization .....</b>	<b>59</b>
<b>Table 14 – NLP Based Models Symbol Legend.....</b>	<b>60</b>
<b>Table 15 - P-values for NLP based models. ....</b>	<b>60</b>

## List of Figures

Figure 1 - Comparing stock returns to random walk.....	3
Figure 2 – Trading strategy developed from moving averages. ....	4
Figure 3 - Flowchart of the process of short-selling .....	5
Figure 4 - Index values.....	6
Figure 5 – Financial Factor Model Example. ....	14
Figure 6 – Example GP Produced Expression Tree .....	20
Figure 7 - SVM for classification .....	21
Figure 8 – Factor Model Cumulative Investment Returns. ....	25
Figure 9 – Non-Factor Model Cumulative Investment Returns .....	25
Figure 10- Charting Magnitude of Index Returns. ....	35
Figure 11 – MOO and SOO Training Data Investments Comparisons .....	38
Figure 12 - MOO and SOO Testing Data Investments Comparisons .....	39
Figure 13 – NLP Based Models Year over year accuracy .....	50
Figure 14 – NLP Based Models Year over year percentage returns.....	50
Figure 15 – NLP Based Models Cumulative Investment Returns .....	51
Figure 16 - S & P 500 Index Returns.....	54

## **Abstract**

This thesis presents Artificial Intelligence (AI) approaches to creating investment models. A novel data representation to optimize forecasting models created with a Support Vector Machine (SVM) and Genetic Programming. The representation is a pseudo financial factor model (PFFM). The results show that both algorithms were able to achieve superior investment returns with the aid of the PFFM.

Next is a multi-objective approach for making predictions of a market index with the aid of an Evolutionary Artificial Neural Network (EANN). The fitness function promoted EANNs that could identify behaviour in the market that predicated direction and magnitude. The results indicated that an EANN trained for multiple objectives was superior to models created using a single-objective optimization.

Finally, text mining techniques for analyzing annual reports, the first is based on n-gram profiles and CNG classification. The second approach combines readability scores and performance measures. Both methods and their combination outperformed the benchmark.

## **Acknowledgements**

I would like to thank Vanessa my loving wife to be, for all her patience and support while I was writing my thesis. Thank you to my supervisor Dr. Vlado Kešelj for all his guidance and support throughout my masters program. I would also like to thank several of my collaborators and colleges, in no particular order, Dr. Malcolm Heywood, Ali Daniyal and Dr. Vladimir Lucic.



## **Chapter 1**

### **Introduction**

This thesis is focused on research into financial modeling with the aid of Artificial Intelligence the benefits of such research can lead to superior forecasting models and improved risk-adjusted investment returns. The financial markets are regarded as a leading indicator to the economy and when the markets begin to contract the population braces for a slowdown in the economy or possibly a recession. Conversely, as the markets begin to expand the business cycle shifts and the economy starts to build forward momentum. Being that the stock market is a leading indicator, making accurate predictions on its movement becomes a difficult task. The forces which affect stock markets are abundant, highly correlated and difficult to accurately measure. Given the large amounts of data available from various sources and what are potentially very complex relationships to model the application of Artificial Intelligence techniques are appropriate. The first two techniques introduced in this thesis (pseudo financial factor modeling and multi-objective optimization) are intended for short-term position traders whom would typically hold investments for 30–90 days. These techniques would be utilized on a monthly basis but monitored more frequently. The third technique (text mining of annual reports) is intended for portfolio management and longer-term investing of at least one year. The technique would be utilized as new annual reports are released to the public, which could be several or very few depending on the month. The remainder of this chapter will introduce traditional investment theory and the specific research objectives for each approach.

#### **1.1 Investment Theory**

One of the most interesting questions in investment theory is how to build robust models for making accurate and reliable predictions of the stock market and its related assets. There are two bodies of analysis that garner the most attention: fundamental analysis and technical analysis. Fundamental analysis is the process of analyzing a company based on its financial statements (balance sheet, income statement, statement of cash flows),

internal stability, quality of senior management and its competitiveness within its industry. This type of analysis relies heavily on quarterly and annual reports released by publicly-traded companies. Technical analysis concerns analyzing historical price data to develop rules for entering and exiting the market. Technical analysts believe that market behaviour has a discernable pattern which can be learned and exploited to create excess returns after transaction costs are incurred.

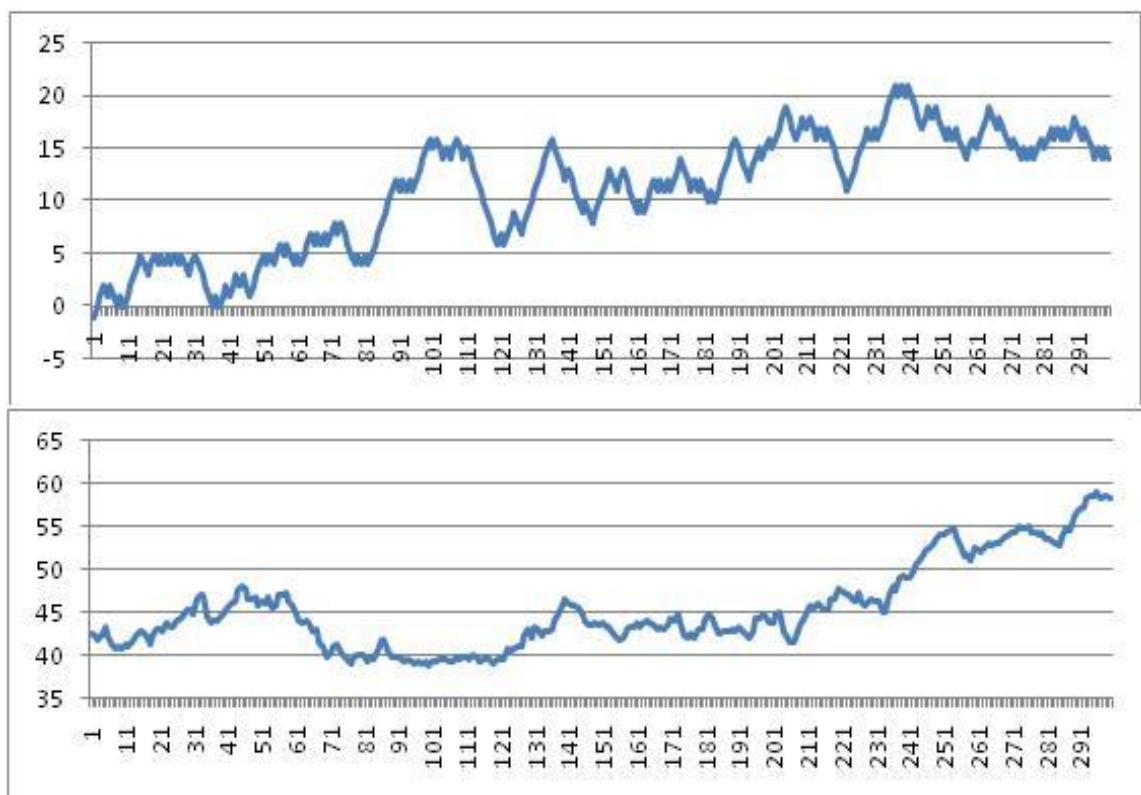
Investment theory concerns a decision making process where one intends to make investments to satisfy certain goals with regards to risk and return. It mainly concerns four related areas, the capital asset pricing model (CAPM), portfolio theory, arbitrage pricing theory and the efficient market hypothesis (EMH). The CAPM, arbitrage pricing theory and portfolio theory are not utilized in the research and therefore are out of the scope of this thesis.

### **1.1.1 Efficient Market Hypothesis**

The efficient market hypothesis was developed by Eugene Fama [1] and has three forms; weak, semi-strong and strong efficiency. Under the weak efficiency a sequence of past changes contains no information about future changes and so stocks follow a kind random walk where an investor cannot produce excess returns to that of the market using technical analysis, meaning that trading rules generated from historical prices will not be able to accurately predict market behaviour. However under weak efficiency, fundamental analysis may reveal information not already incorporated into the market prices and therefore create excess returns. The semi-strong form states that when new information is released it will be quickly incorporated into the share price so that neither fundamental nor technical analysis could utilize it to produce excess returns. Finally in the strong form of efficiency the hypothesis states that all information pertaining to a stock price including public and private information is included in the stock price which implies that no one could produce excess returns including company insiders (President, CEO, COO, etc.). Despite the popularity of the EMH there is a significant portion of investors whom do not believe in it in any form. Several studies provide evidence that technical analysis can consistently produce excess returns and that the stock market is not completely efficient [2, 3, 4, 5, 6] nor do stock prices follow a random walk [7].

### 1.1.2 Random Walk Hypothesis

The random walk hypothesis states that stock prices follow a random walk and do not follow any patterns or trends. It is initially theorized by a French economist Louis Bachelier in 1900, however it came back into main stream economics in 1965 in Eugene Fama's article "Random Walks in Stock Market Prices" [8]. Figure 1 replicates the coin tossing experiment used by Burton Malkiel, author of the book "A Random Walk Down Wall Street". If the coin lands heads up the stock moves up one point if the coin lands tails up the stock moves down one point. Such experiments mimic the nature of stock price movements and are often cited as evidence for the random walk theory.

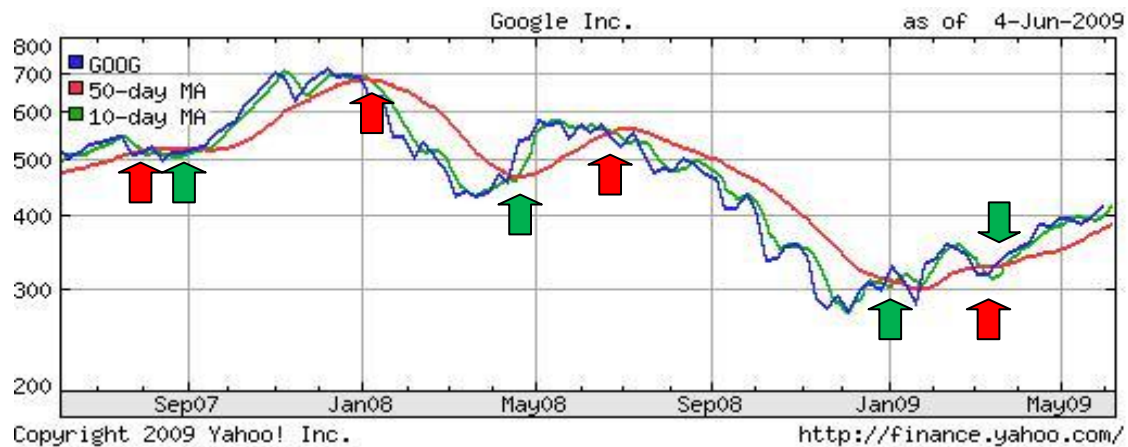


**Figure 1 - Comparing stock returns to random walk - (Top) shows the results from 300 trials of a coin tossing experiment where the value of the fictitious stock moves up one point for tails and down one point for heads. (Bottom) the historical price chart for Alcoa showing the closing price for 300 days of trading.**

## 1.2 Trading Strategies

Investors whom are looking to govern their decisions by a set of rules, thereby eliminating any trades induced from an emotional response or "gut feeling" will employ a trading strategy which lays out a set of guidelines on when and if to enter or exit a trade.

The specific set of rules may be something very simple that follows interactions between moving averages or very complex with sort-selling and where several criteria need to be satisfied before a decision is made. In either case these systems are usually back tested to check the validity of the trading rules. Back testing does not guarantee future performance but will provide confidence if the system tests well. An example of using simple moving averages to trade Google Inc. is shown in Figure 2.

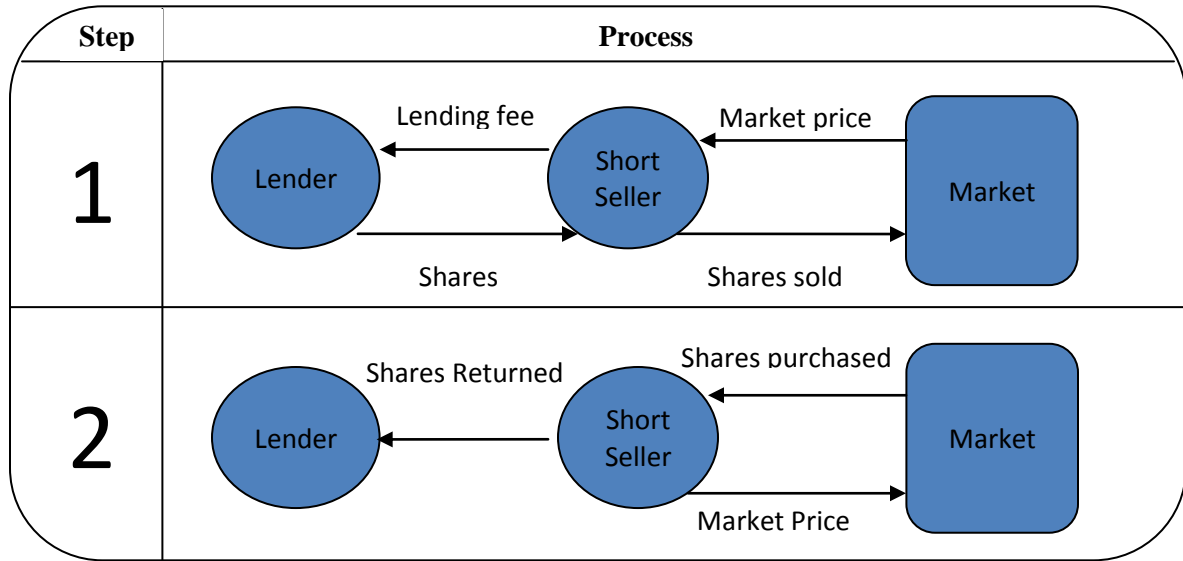


**Figure 2 – Trading strategy developed from moving averages - 2 year historical price chart for Google Inc. showing the interactions between 10 and 50 day simple moving averages. Buy signals are created when the 10 day MA crosses above the 50 day MA (green arrows). A sell signal is created when the 10-day MA crosses below the 50-day MA (red arrows).**

### 1.2.1 Short-selling

Short-selling a stock is a trading action where the seller of an asset does not own the asset at the time of the transaction with the intention of buying the asset back at a lower price sometime in the future. In this case the agent making the sale is expecting the value of the underlying asset to decrease. To facilitate such a transaction the seller borrows the asset from a lender for a fee, and then sells it at the current market price. Then at a non-predetermined time in the future the short-seller reimburses the lender of the asset once the position is closed out and the asset is bought back. The act of short-selling has received criticisms because some believe it causes speculators to devalue assets below their fair market price and has even been blamed for compounding the negative effects of recent market turmoil [9]. However proponents of short selling sight that it allows for an investor to profit from both contractions and expansions in the market and therefore

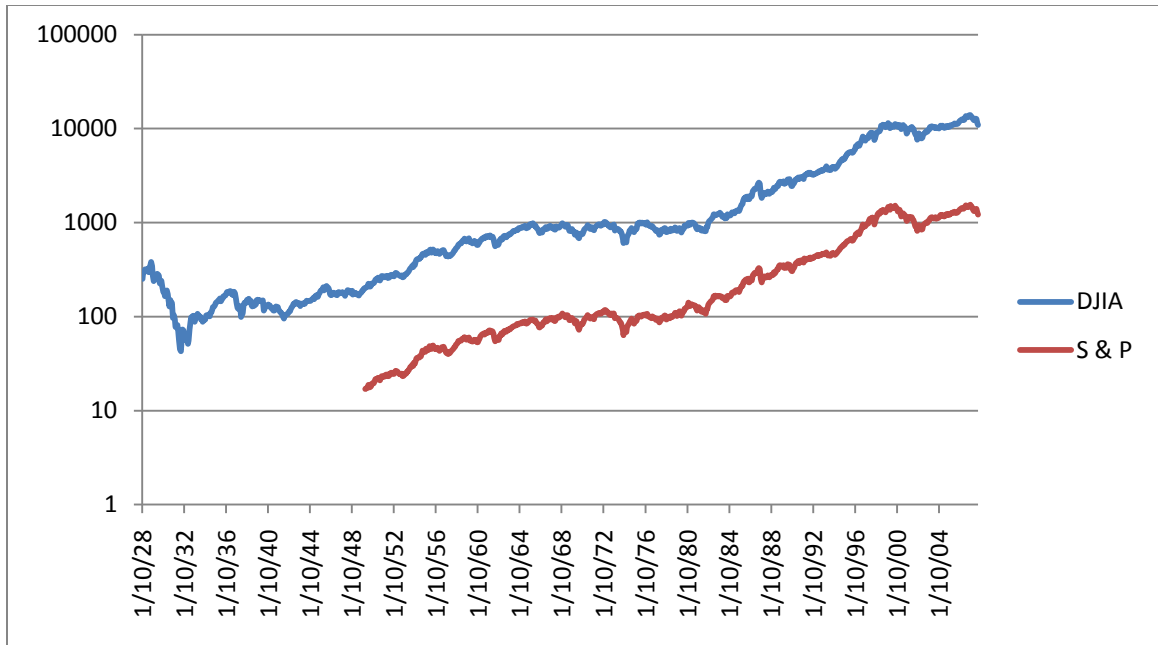
counteracting assets becoming overvalued due to speculations only working in one direction [10]. Figure 3 shows a flowchart of the process of short-selling.



**Figure 3 - Flowchart of the process of short-selling.** In step 1 the short seller borrows the shares from a lender for a lending fee and then sells the shares on the stock market at the current market price. In step 2, which occurs at sometime in the future, the short seller purchases the shares from the stock market at the new current market price and then returns the shares back to the lender.

### 1.3 Prediction Methodology

In each of the three approaches presented in this thesis the data mining task is classification. Two forms of two class classification are utilized; the first is classifying the stock market movement as either expanding or contracting over the next month. The second is classifying stocks as either over or under performing a relevant benchmark. Classification was chosen rather than regression or level estimation because research suggests that it is superior in terms of accuracy of the models produced and investment returns [11]. The second form of classification stems from an investors main objective to produce excess returns. As an investor, if you have a balanced portfolio with a long-term time horizon the wealth will increase at a higher rate than just depositing the money in a bank account. Since black Thursday, October 24th 1929, when the Dow Jones Industrial Average (DJIA) dropped over 50% in one day and was the beginning of the great depression, the markets have enjoyed a relatively steady increase. Figure 4 shows the long-term growth of the two major indexes in the United States.



**Figure 4 - Index values - shown on a logarithmic scale base 10, for the DJIA (1928 -2008) and the S&P 500 (from 1950).**

If the markets in the long-term have the natural tendency to rise than the only reason to conduct analysis would be to outperform the markets and therefore it is logical that the relative performance of a stock is more important than its value at some predetermined time in the future. For example if a stock is projected to gain 20% over the coming year, even if this estimate was accurate it would be unfavourable if the market was performing in excess to that return.

#### **1.4 Research Objectives and Hypotheses**

The main objective of this research is not to provide evidence for or against the efficient market hypothesis or random walk, rather it is under the assumption that markets are not efficient and aims to improve and introduce new techniques for trading and investing in the stock market. The assessment of each technique includes a comparison to a relevant benchmark and as a result may provide incidental evidence in support of either side of the argument. The three approaches are as follows:

1. Pseudo financial factor modeling of monthly stock market returns,
2. Evolutionary Artificial Neural Networks applied to modeling monthly stock market returns with a multi-objective approach, and
3. Character n-gram analysis and readability scores of annual reports.

### 1.4.1 - Pseudo Financial Factor Modeling

The goal of this chapter is to introduce a novel data representation based on a linear factor model and to test its effectiveness to create more robust trading models developed from complex learning algorithms, a support vector machine and tree-based genetic programming. The null hypothesis is expressed as follows:

$H_0$  – *Trading models developed from support vector machines and tree-based genetic programming with the aid of the PFFM are not superior to those only utilizing monthly changes in their input data indicator set.*

Thus the corresponding alternative hypothesis is:

$H_1$  – *Trading models developed from support vector machines and tree-based genetic programming with the aid of the PFFM are superior to those only utilizing monthly changes in their input data indicator set.*

### 1.4.2 – Multi-objective Optimization

This research is attempting to overcome a short fall with a popular classification technique used in financial forecasting with artificial intelligence [12, 13, and 14]. Classifying the movement of an asset as either increasing or decreasing over a pre-determined time period in the future fails to consider the effects of making a wrong prediction when there is a substantial change in the underlying assets value. The consequence being that a higher accuracy does not ensure higher investment returns or lower risk. The solution put forth is to train classifiers over two objectives:

To identify behaviour that predicate as assets movement in terms of:

1. Direction (up or down)
2. Magnitude (above or below 1 standard deviation of the mean change)

The null hypothesis is:

$H_0$  – *The results obtained from training the EANN with a multi-objective approach are not superior from those obtained under a single-objective optimization.*

Conversely the alternative hypothesis is:

$H_1$  – *The results obtained from training the EANN with a multi-objective approach are superior from those obtained under a single-objective optimization.*

### **1.4.3 – N-gram Analysis and Readability Scores of Annual Reports**

This research introduces two novel natural language processing (NLP) techniques for financial forecasting. The first is based on character/word n-gram profiles, where the profiles are classified by the CNG (common n-gram) distance measure [15]. The second method creates vector representations of the annual reports containing commonly used readability scores and performance measures. The vector representations are then used to train a support vector machine. Later the models are combined to add a level of confidence to any investment decisions made.

The null hypothesis is:

$H_0$  – *The combined models do not yield a level of over-perform precision which implies a significant improvement over the individual models.*

The alternative hypothesis is:

$H_1$  – *The combined models yield a level of over-perform precision which implies a significant improvement over the individual models.*

## **1.5 - Organization of Thesis**

This thesis is organized into 7 chapters. The 1<sup>st</sup> chapter introduces the approaches utilized in the thesis and the main objectives for each. It also provides an introduction to investment theory, modeling techniques and data mining tasks common to each approach. The second chapter provided background and related work, the third chapter details the research on the Pseudo Financial Factor Model and the experiment results. Chapter 4 is on multi-objective optimization with evolutionary artificial neural networks for modeling the monthly returns of the Dow Jones Industrial Average (DJIA). Chapter 5 is on Natural Language Processing (NLP) and details the approaches of utilizing character n-gram analysis and readability scores of annual reports for financial forecasting. Chapter 6 is for results analysis and statistical testing of each approach. Finally, chapter 7 gives the general conclusions for the thesis and possible extensions for future work.



## **Chapter 2**

### **Background and Related Work**

The approaches introduced in this thesis touch on several areas of Artificial Intelligence (AI). This chapter will begin with related work and background for the first methodology of financial factor modeling using support vector machines (SVM) and Genetic Programming (GP). Secondly, background on Evolutionary Artificial Neural Networks (EANN) and Multi-objective Optimization (MOO) approaches and finally this chapter concludes with related work in the area of Natural Language Processing (NLP) and text mining.

#### **2.1 Support Vector Machines**

Research that concerned the utilization of support vector machines was done by Fan, Allan and Palaniswami [16] where the authors used a SVM in classification to make predictions on individual securities trading on the Australian Stock Market. The model developed from fundamental data by the SVM was able to produce a 208% return over a 5 year period which was a significant increase over the benchmark return on only 71%. Huang, Nakamori, Wang [17] made weekly predictions in the movement of the NIKKEI 225 index of Japan. The authors concluded that the SVM was a more robust classifier when compared to other methods such as Quadratic Discriminate Analysis and Neural Networks. Their inputs were macro-economic indicators such as Gross Domestic Product (GDP) and short term interest rates. In other work done by Gavrishchaka and Banerjee [18] they used a SVM to extract information from and describe the volatility in the S&P 500, the results suggested that SVM's were superior to the main-stream volatility models. The application of such predictions would be instrumental in developing volatility trading models and risk management strategies.

#### **2.2 Genetic Programming**

Now the use of GP in the financial domain dates back several decades where it has been used to solve problems of investment optimization and risk. Some of the first work with

GP in price prediction was done by Iba and Sasaki to learn investment decisions on the Japanese stock market [19]. Their findings showed GP had some advantages over neural networks but the actual results were hindered by ignoring transaction costs. Also in 1999 Li and Tsang [20] used a GP approach to improve technical trading rules. They developed a forecasting tool called EDDIE (Evolutionary Dynamic Data Investment Evaluator) that is used to discover non-linear functions to explain interactions among variables. Potvin, Soriano, and Vallee used GP to automatically generate trading rules for 14 stocks trading in the Canadian Market [21]. The approach performed well under stable or falling markets but underperformed a traditional buy-and -hold approach in a bull market.

### **2.3 Evolutionary Neural Networks**

Some recent work with evolutionary neural networks includes a paper by Azzini and Tettamanzi [22] where the authors evolved a neural network for financial factor modeling. To counteract the destructive properties of crossover for neural networks of different topologies they implemented a hidden layer insertion mutation operator that was applied to the smallest network to obtain two networks with equal hidden layers. In their model back-propagation was included and was optionally used to decode a genotype into a phenotype. Mora, Castillo, Merelo, Esparcia-Alcazar, and Sharman [23] compared the effectiveness of genetic programming (GP) merged with self-organizing maps (SOM) to an evolutionary ANN method for discovering causes of financial distress. Their EANN was based on a population of multi-layer perceptrons (MLP) with two hidden layers. The number of hidden neurons and weight connections were interchanged and mutated during evolution. Their results suggested that the GP-SOM model was superior to the EANN. A recent attempt with MOO for stock trading was undertaken by Briza and Naval [24] where they created an end-of day trading model with multi-objective particle swarm optimization. Their model optimized on two objective functions, the Sharpe ratio and percent profit and was able to outperform the technical indicators under study, however during the testing periods the market itself was the top performer.

## 2.4 Text Mining

As text processing techniques become more sophisticated their ability to work in the financial domain becomes more attractive. There have been a few publications in which textual information was analyzed in relation to financial performance. In comparison, the novelty of our approach is in applying character n-gram analysis and readability scores with the SVM method to the annual reports in making long-term predictions. Pushing the time-horizon for making predictions creates a more practical model, and thus it has a wider appeal in the investment industry. In [25], the effects of news articles on intra-day stock prices are analyzed. The analysis was conducted using vector space modeling and tfidf term weighting scheme, then the relationship between news stories and stock prices was defined with a support vector machine. The experiments produced results with accuracy as high as 83% which translated to 1.6 times the prediction ability when compared to random sampling. Similarly, Chen and Schumaker (2006) [26] compared three text processing representations combined with support vector machines to test which was the most reliable in predicting stock prices. They analyzed the representations based on bag-of-words, noun phrases and named entities, and all of the models produced better results than linear regression; however named entities proved to be the most robust. Other intra-day predictions facilitated through text mining were done by Mittermayer (2004) [27], where he created NewsCATS— an automated system that could day-trade the major American stock indexes. The model was created to automate the trading decisions based on news articles immediately after they are released. Kloptchenko *et al.* (2002) [28] focused on clustering quarterly financial reports in the telecom industry. They were not making predictions on future performance but attempting to use prototype-matching text clustering and collocational networks to visualize the reports. The collocational networks cut down the time required by an analyst to read the report and identify important developments. This work was improved upon for making predictions and the new results (Kloptchenko *et al.* 2004) [29] were released, in which prototype-matching text clustering for textual information was combined with self-organizing maps for quantitative analysis. Their analysis was performed on quarterly and annual financial reports from three companies in the telecom industry. The results implied that some indication about the financial performance of the company can be gained from the textual

component of the reports; however, it was also noted that the clusters from quantitative and qualitative analysis did not coincide. They explained this phenomenon by stating that the quantitative analysis reflects past performance and the text holds information about future performance and managerial expectations. Before complex text mining methods were developed, the work done by Subramanian, Insley, and Blackwell [30] in 1992 showed that there was a clear distinction between the readability scores of profitable and unprofitable companies. In more recent work by Li [31], he examined the relationship between annual report readability combined with current earnings and earnings persistence, with a firm's earnings. His conclusion was that firms with lower earnings had reports which were more difficult to read and longer.

## Chapter 3

# Optimizing a Pseudo Financial Factor Model with Support Vector Machines and Genetic Programming<sup>1</sup>

### 3.1 Introduction

Over the last several years Machine Learning (ML) methodologies have become increasingly popular for aiding investment decisions. In particular Support Vector Machines (SVM) and Genetic Programming (GP) approaches have garnered increased attention due to their ability to handle the complex and non-linear behavior exhibited in the stock market.

We concentrate on and compare the results from a SVM and tree-based GP. To demonstrate the effectiveness of using the pseudo financial factor model the algorithm outputs are compared to models developed from using only the monthly changes of the inputs. The information that is used to generate the predictions is macro-economic data such as information on inflation and corporate bond ratings. Before the data is introduced to the various ML methodologies it is pre-processed with the aid of PCA to rank relevant attributes and remove excess noise. The relationship between market movements and macro-economic data is not linear or monotonic. To assist in modeling these relationships a financial factor model is created that represents correlations between the market and each indicator in the input set. The task for the models is classification of market movement and whether or not the market will contract or expand on a month to month basis.

### 3.2 Financial Factor Modeling

The inspiration for the financial model used is a linear financial factor model, which relates the returns of an asset to the returns of other correlated assets or factors. Traditionally a financial factor model would be used to explain the returns of an asset by an equation and when the model output and the actual return begin to diverge,

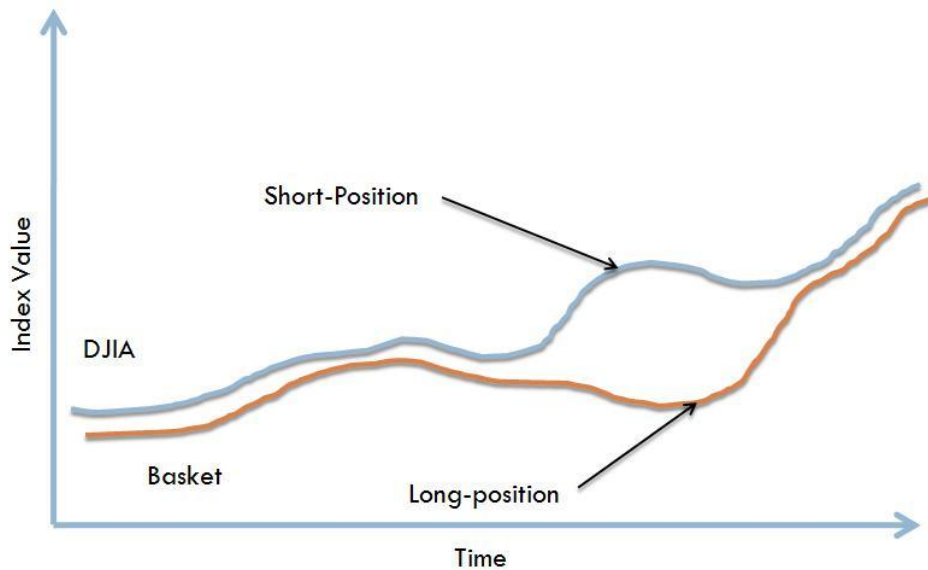
---

<sup>1</sup> Based on a co-authored paper by M. Butler and V. Kešelj. Canadian AI 2009, LNAI 5549, pp. 191-194.

appropriate investments are made under the assumption that the two will converge again in the near future. The canonical equation is shown below:

$$r_i = b_{i1} * f_1 + b_{i2} * f_2 + ... + b_{im} * f_m + \epsilon_i$$

where  $r_i$  is the return on asset  $i$ ,  $b_{i1}$  is the change in return of asset  $i$  per unit change in factor 1,  $\epsilon_i$  is the portion of the return in asset  $i$  not related to  $m$  factors,  $m$  is the number of factors and  $f_1$  is the change in return of factor 1. With the traditional form the basket of assets would not be financial indicators but actual assets which can be invested in such as foreign exchange, individual stocks and derivatives. The model does not make any predictions for the future but rather attempts to model market inefficiencies, represented by the convergence of the model and the underlying asset. Once a convergence has occurred the trader exploits the assumption that the two models will converge again when the market corrects itself. Figure 5 illustrates the canonical use of financial factor modeling.



**Figure 5 – Financial Factor Model Example - An example of the traditional use of a financial factor model. Once the model and the underlying asset begin to diverge a trading opportunity is created. Under the assumption that the model and the DJIA will converge again in the future the trader makes the appropriate trades to exploit the inefficiency. In this example a long position is taken on the basket of assets (expecting them to increase in value) and short position is taken against the DJIA (expecting a contraction in the market value).**

As the name of the paper suggests we are not using the equation in the traditional sense but changing the left hand side of the equation to be a class rather than a price level. Keeping the classes in the preferred form for Support Vector Machines, a class of 1 indicates the DJIA will rise over the next month and -1 suggests it will fall over the same time period. The error term on the right hand side is not supplied to the data set but is left for the algorithm to learn. The new pseudo equation is thus:

$$r_i = b_{i1} * f_1 + b_{i2} * f_2 + ... + b_{im} * f_m$$

where  $r_i \in \{1, -1\}$

### 3.3 Data Description and Pre-processing

The data used to train the models was based on macro-economic data that was utilized by Enke and Thawornwong [32] where they created a market prediction model that outperformed the S&P 500 market index with the aid of a multi-layer perceptron. To make the predictions more realistic some of the inputs were lagged by one or two months to reflect the actual times the data would be released. A full list of input attributes is provided in table 1. All the data was transformed from an actual level to a percentage change on a month to month basis  $([P_t - P_{t-1}] / P_{t-1})$  so that a rate-of-change could be combined with the Beta's (or factors) to give the factor models for the prediction. The Beta's which are an indication of how much the market would move based on a unit movement of 1 in a given indicator was calculated on a rolling 10-year period, for example 2005 predictions were made with Beta's calculated from 1995 – 2004, the Beta equation is:

$$Beta(DJIA, X_i) = \frac{cov(DJIA, X_i)}{var(X_i)}$$

where  $X_i$  is a given change in a macro-economic indicator and DJIA is the monthly change in the Dow Jones Industrial Average. Each algorithm was trained on data from 1977 to 2001 and then tested for 84 months or 7 years up until June 2008. The

justification for the extended training period was to expose each model to market reactions during each stage of the business cycle.

Principle component analysis (PCA) was chosen to remove excess noise and rank the attributes in terms of relevancy. Initially the data is projected into principle component space where 99% of the variance is equated for, then the data is projected back into attribute space with a proper rank and excess noise removed. It was decided to project the data back into attribute space due to the fact that there was insufficient data to retain the 29 inputs if PC space was chosen. Through experimentation it was determined that having all inputs present yielded higher classification results. The PCA was performed in WEKA [33]. The results from ranking the attributes are shown in table 2. An interesting result was that the attributes ranked in the same order using either the pseudo financial factor model data representation or just the monthly changes.

**Table 1 - Input Attributes for the PFFM**

<b>Macro-economic Indicators (symbol)</b>			
Moody's seasoned AAA bond yield	(AAA)	Moody's seasoned BAA bond yield	(BAA)
Previous month return for the S&P 500	(S&P)	6-month certificate of deposit rate	(CD6)
Consumer Price Index	(CPI)	Difference between T120 and T3	(TE2)
Industrial Production Index	(IPI)	Difference between T120 and T6	(TE3)
Producer Price Index	(PPI)	Difference between T120 and T12	(TE4)
M1 Money Stock	(M1)	Difference between T60 and T12	(TE8)
M2 Money Stock	(M2)	Difference between T60 and T6	(TE9)
3-month T-bill rate	(T3)	Difference between T60 and T3	(TE10)
6-month T-bill rate	(T6)	Difference between BAA and AAA	(DE1)
1-year T-bill rate	(T12)	Difference between BAA and T20	(DE2)
5-year T-bill constant maturity rate	(T60)	Difference between BAA and T12	(DE3)
10-year T-bill constant maturity rate	(T120)	Difference between BAA and T6	(DE4)
1-month certificate of deposit rate	(CD1)	Difference between BAA and T3	(DE5)
3-month certificate of deposit rate	(CD3)	Difference between CD6 and T6	(DE6)
Previous monthly return for the DJIA	(Prev)		



**Table 2 – Input Attribute Rankings - displays the results from ranking the input attributes of the pseudo financial factor model with principal component analysis.**

Rank	Attribute	Rank	Attribute
1	T6	16	DE1
2	S&P	17	DE2
3	T3	18	DE7
4	T120	19	Prev
5	CD1	20	DE4
6	T12	21	DE5
7	T60	22	TE2
8	CPI	23	TE3
9	AAA	24	CD3
10	BAA	25	CD6
11	M2	26	TE9
12	PPI	27	TE10
13	IPI	28	TE4
14	M1	29	TE8
15	DE3		

### 3.4 Methodology and Software Utilization

The following subsections outline the general concepts related to our approach and the software that was used.

#### 3.4.1 Genetic Programming Approach

The tree-based Genetic Programming (GP) model was created with lilgp [34] a complete GP environment written in C, it allows for all the standard genetic operators and for multiple populations to develop simultaneously. Tree-based genetic programming (GP) as proposed by John Koza [35] is an extension of genetic algorithms (GA), introduced by John Holland [36]. Unlike genetic algorithms, GP has more freedom with its representations and without the constraints of binary bit string to encode the “genetic” information GP is able to have more sophisticated structures. GP has grown in popularity due to its abilities to handle complex optimization problems and efficiently explore solution space. A high level description of the algorithm is as follows:

1. A population of individuals are initialized containing randomly constructed decisions trees based on the available terminal and non-terminal sets
2. Each individual is assigned a fitness value based on a fitness function

3. Individuals are chosen from the current population for various genetic operators (crossover, mutation and reproduction)
4. A new population is generated from the children created from mutation and crossover and individuals from previous population chosen for reproduction
5. Individuals from the most recent generation are ranked by fitness and steps 3 and 4 are repeated until a termination criteria is satisfied

The fitness function for evaluating the populations involves using a basic wrapper function where the program treats any value an individual from the population returns greater than 0 as 1 and any negative value as 0. Where an individual's rank in terms of fitness is a measure of how many instances are classified correctly (or how many hits are accumulated, more hits translates to a higher fitness rank and therefore more likely to be selected for future populations). Below is pseudo code for how the fitness function is evaluating the population.

$$\begin{aligned}
 & \text{if } ((value > 0.0) \& (actual\ class > 0.5)) \xrightarrow{yields} hit \\
 & \text{else if } ((value \leq 0.0) \& (actual\ class \leq 0.5)) \xrightarrow{yields} hit \\
 & \text{else } \xrightarrow{yields} error
 \end{aligned}$$

In table 3 the selection operator “Fitness over-select” refers to a selection methodology where the population is separated into two or more groups based on fitness and the group which holds the fittest individuals is sampled more often than others. This can lead to expedited evolution of a population in relation to regular fitness selection where the probability of an individual being selected is directly proportional to its fraction of the population's total fitness accounted for by that individual. Table 3 displays the main parameters during evolution and an example of an expression tree created during evolution is given below in figure 6.

**Table 3** – GP Parameters

Name	Value
Initialization method	Half-and –half
Selection operators	Fitness over-select
Number of generations	500
Population size	250
Number of populations	3
Terminal Set	{ *, /, +, - }
Crossover rate (p1, p2, p3) <sup>2</sup>	0.8, 0.5, 0.0
Mutation rate (p1, p2, p3)	0.0, 0.0, 0.8
Reproduction rate (p1, p2, p3)	0.2, 0.2, 0.2
Frequency of population exchanges	Every 50 generations
Tree maximal depth	10
Maximum number of nodes	106

### 3.4.2 Support Vector Machine Approach

The Support Vector Machine environment utilized was LIBSVM [37] a very powerful integrated software for support vector classification and regression. It implements an SMO-type algorithm [38] for solving the Karush-Kuhn-Tucker (KKT) conditions. A SVM is a maximum margin classifier that attempts to find a plane or hyper-plane (in multi-dimensional space) that creates the largest distance between classes in an effort to

---

<sup>2</sup> P1, p2 and p3 refer to population 1, 2 and 3 respectively as each population has unique settings for their genetic operators.

minimize errors on new out of sample data. Figure 7 shows an example of SVM classification in 2-dimensional space

```

TREE:
  (- (/ t3
    (* (- (+ de7 cd6)
      (/ (* (/ t120 te2)
        (- cd6 aaa)) te2))
      (- (* (* (- cd6 aaa)
        (* ppi t120))
        (/ (* te2 cd3)
          (/ (* (* bbb cd6)
            (/ te3 ipi))
            (* (* t60 m1)
              (/ ipi te2))))))
      (* (- (- de4 ipi)
        (* m1 ipi))
        (* t3
          (* (+ de3 te2)
            (* de1 ppi))))))
    (* (- (/ ipi te2)
      (- (* (- cd6 aaa)
        (* bbb cd6))
        (* (- cd6 aaa)
          (/ ipi te2))))
      (* (- de4 ipi)
        (- (- de4 ipi)
          (/ (* (+ de7 cd6)
            (* (/ te3 ipi)
              (/ te3 ipi))) te2))))))

```

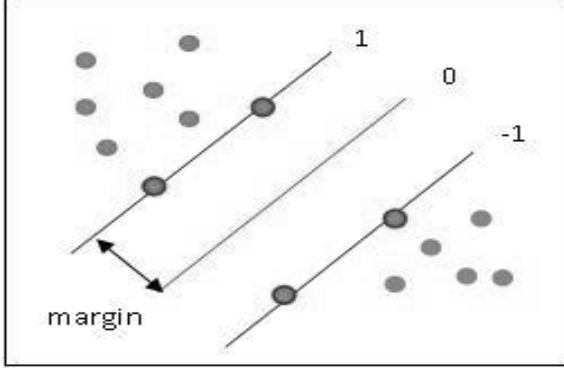
**Figure 6 – Example GP Produced Expression Tree - An example of an expression tree generated during evolution. The tree is more complex and less intuitive in its current state due to intron code, which would normally be pruned for requirements of parsimony. Several input parameters are used multiple times (aaa, cd6 and ipi), while others are not present at all. An examination of the tree can give insights into which inputs are the most effective and describing the problem at hand.**

The generalized primal form of the SVM classification is a quadratic programming (QP) optimization problem of the form

$$\arg \min_{w,b} \frac{1}{2} \|w\|^2$$

this is subject to the constraints of

$$t_n(w^T \phi(x_n + b)) \geq 1, \quad n = 1, \dots, N$$



**Figure 7 - SVM for classification - Shows the hyperplane at  $y=0$  for a linear classifier where the margin is the distance from the hyperplane to the support vectors shown as the bolded points on the line.**

Because working in feature space to solve this equation is particularly difficult the equation has a dual form which makes use of a kernel function, defined by  $k(x, x') = \phi(x)^T \phi(x')$ , and Lagrange multipliers. The dual representation is

$$\tilde{L}(a) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(x_n, x_m)$$

with respect to the constraints

$$a_n \geq 0, \quad n = 1, \dots, N,$$

$$\sum_{n=1}^N a_n t_n = 0.$$

With these formulations the data will be transformed into higher possibly infinite dimensional space by the function  $\phi$  in the kernel function given above.

With simple problems the data is linearly separable in feature space  $\phi(X)$  and new data points can be classified based on the following equation, once a model has been constructed,

$$y(x) = \sum_{n=1}^N a_n t_n k(x, x_n) + b$$

where  $a_n$  is a particular Lagrange multiplier for each constraint,  $t_n$  is the target value,  $k(x, x_n)$  represent the kernel function and  $b$  is a bias parameter. However, in financial forecasting the data is very noisy and for the most part will not be linearly separable. When the data cannot be easily separated the SVM needs to introduce a slack variable

which will be able to penalize the incorrect classification of data points, this is accomplished by minimizing:

$$C \sum_{i=1}^l \varepsilon_i + \frac{1}{2} \|w\|^2$$

where the parameter  $C > 0$  and controls the trade off between the slack variable penalty and the margin.

Next I will discuss the primal and dual form for the c support vector machines examined in this report. The primal form considered for c-support vector classification (SVC) is:

$$\min_{w,b,\varepsilon} \frac{1}{2} w^T w + C \sum_{i=1}^l \varepsilon_i$$

subject to  $y_i(w^T \phi(X_i) + b) \geq 1 - \varepsilon_i$

$$\varepsilon_i \geq 0, i = 1, \dots, l$$

The dual form is

$$\min \frac{1}{2} \alpha^T Q \alpha - e^T \alpha$$

subject to  $y^T \alpha = 0$ ,

$$0 \leq \alpha_i \leq C, \quad i = 1, \dots, l,$$

where  $e$  is a vector of all ones,  $C > 0$  is the upper bound,  $Q$  is an  $l$  by  $l$  positive semi definite matrix,  $Q_{ij} \equiv y_i y_j K(x_i, x_j)$ , and  $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$  is the kernel function. Table 4 displays the values for the various settings of the SVM, the values of  $C$  and gamma were initially estimated with the aid of a grid search supplied with LIBSVM.

**Table 4 - SVM Parameters**

Name	Value
C (slack value)	1000
Gamma	2.5
Normalize	False
Kernel Type	Radial Basis Function

### 3.5 Trading Strategy

The experiment is setup as a semi-active trading strategy where at the beginning of each month a prediction is made as to whether or not the DJIA will contract or expand over the coming month. If the prediction is for the market to go up than the model will take a long position, conversely if the market is predicted to fall than a short position will be taken. Several financial instruments are available to short the DJIA, but essentially they all profit from market contractions. Given that we are only investing directly in the market the only way to outperform it is to avoid times when the DJIA is falling, this can be achieved by either shorting the DJIA, which we have opted to do, or by exiting the market during those times and investing in a risk-free interest rate. The latter is a more conservative approach and would be preferred if model confidence is low.

### 3.6 Testing Results

The best model for both algorithms from each data set, determined by the training results, was supplied the out-of-sample data that spanned 84 months from 2001 up until June of 2008. In Table 5 we display the testing results for each algorithm. The investment returns are based off an initial investment of \$1000 and for simplicity reasons transaction costs are ignored. Reported in the results is the Sharpe Ratio, which is a gauge of how much additional return the trading system generates for the extra risk it is exposed to—the higher the Sharpe ratio the better the risk-adjusted performance. It is calculated as:

$$\mathcal{S} = \frac{E[R - R_f]}{\sqrt{\text{var}[R - R_f]}}$$

where  $R$  is the return on the asset and  $R_f$  is a risk-free rate. For our calculations the risk-free rate is replaced by the DJIA monthly return.

The testing results in Table 5 clearly show the advantages of using the financial factor model to create the inputs for the SVM and GP algorithms, where the overall accuracy and investment return were superior.

**Table 5 - Testing results for each model – GP and SVM with and without the PFFM.**

	SVM		GP	
	Factors	No Factors	Factors	No Factors
Overall Accuracy	69.05%	59.52%	66.67%	57.72%
Precision (contractions)	68.60%	60.00%	63.41%	55.55%
# of contraction predictions	35	25	41	27
Precision (expansions)	69.40%	59.30%	69.76%	57.89%
Yearly Investment Yield (%)	21.70%	4.13%	16.00%	4.04%
Cumulative Return (\$)	\$4505	\$1335	\$3042	\$1327
Excess return to market <sup>3</sup>	20.60%	3.03%	14.90%	2.94%
Sharpe Ratio <sup>4</sup>	3.994	0.822	2.813	0.795

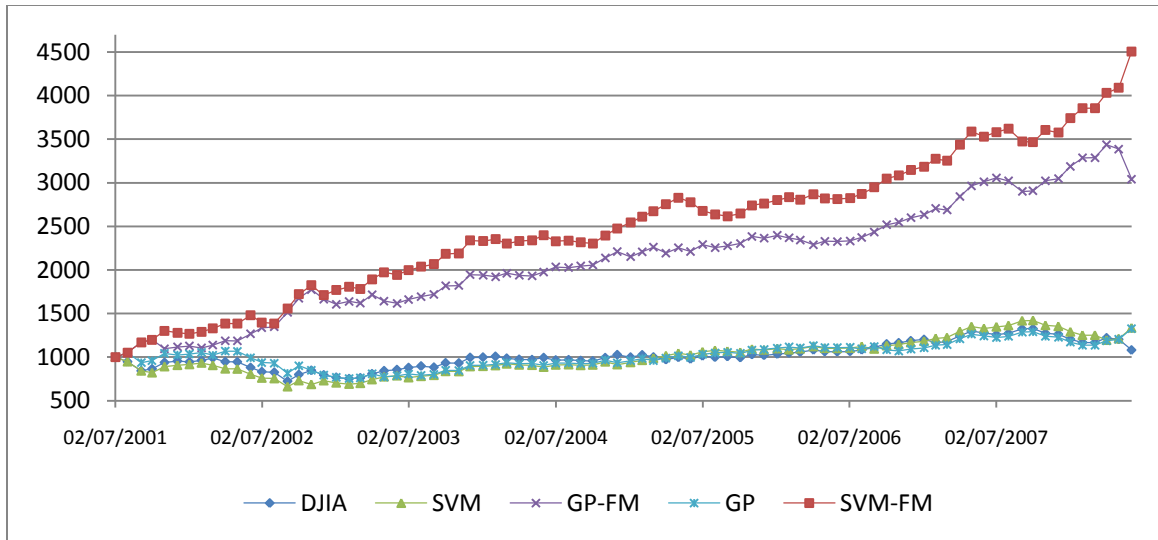
Due to the aggressive nature of the trading strategy it is particularly important to have a high precision on contraction predictions. Where an incorrect decision leads to the investment portfolio to a negative return when the market is increasing in value. When the benchmark is the market you are investing in, a precision of less than 50% will most likely lead to underperformance. We can see that the algorithms utilizing the pseudo financial factor model had superior accuracy in this category and over a larger number of predictions, this is a main contributor to the larger excess returns. Whether or not the differences are statistically significant will be discussed in chapter 6. Figure 8 graphs the performance of each algorithm and the benchmark (DJIA) in terms of cumulative investment return.

The superior accuracy and contraction precision of the pseudo financial factor models results in the significant difference for cumulative investment return over the out-of-sample testing period. At the bottom of the chart the non-factor models blend into the DJIA the benchmark return. To help with visualization, figure 9 plots only the non-factor models and the DJIA.

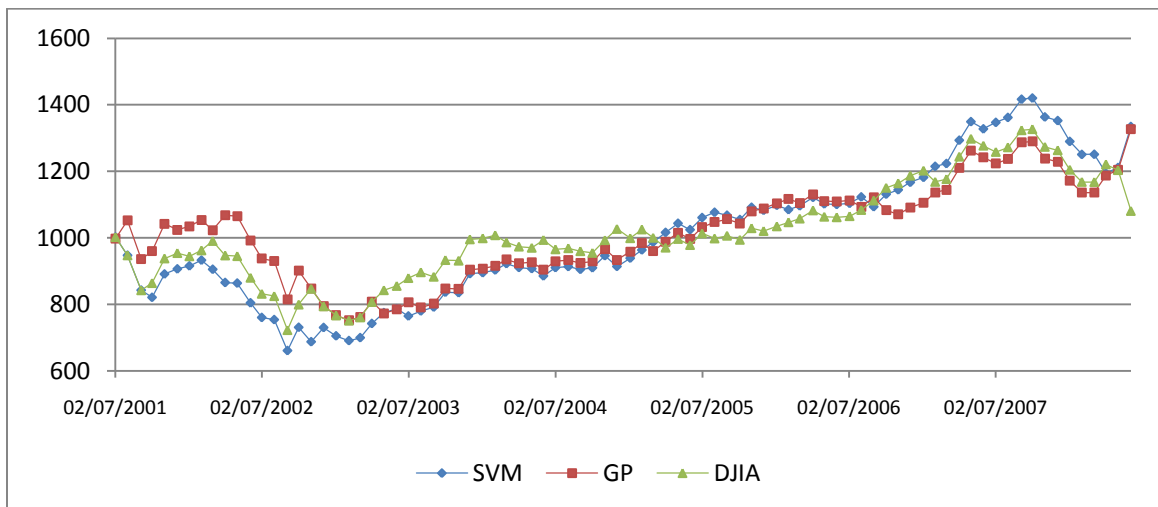
<sup>3</sup> DJIA yearly investment return over testing period was 1.10%.

<sup>4</sup> The risk-free rate in the calculation was replaced by the market rate.





**Figure 8 – Factor Model Cumulative Investment Returns – displaying the resturns for each of the four models and the DJIA (the benchmark return) during the out-of-sample testing data.**



**Figure 9 – Non-Factor Model Cumulative Investment Returns - a closer look at the non-factor models and the DJIA over the testing data.**

Figure 9 provides a closer look of the aforementioned cumulative returns. Although the algorithms create excess returns to the market, they are not substantial. The returns do not include transaction costs which would negatively impact the investment returns, this problem is compounded by the fact that the models are more risky than the benchmark. As a result the investor is not adequately compensated; this conclusion is represented by the low Sharpe ratios for each of the non-factor models.

### **3.7 Discussion and Conclusions**

In this study we compared the effectiveness of a novel data representation to optimize SVM and GP trading models to make accurate predictions on the movement of the DJIA. In each of the performance measures the algorithms achieved superior performance when the inputs reflected the pseudo financial factor model. Precision for contraction predictions was of particular interest in this study due to the trading strategy. Since we are investing directly in the DJIA and also using it as the benchmark the only way to outperform is to avoid market contractions. The effectiveness of using the factor model could be explained by the fact that the algorithms are given more information about the problem with this type of data representation. Not only is the model training on the returns of the indicators but they are also supplied a ratio that describes the relationship between said indicator and the market. This enables the algorithm to have a more complete picture and therefore is able to create a more robust market model. Each of the models presented in this paper were able to outperform the DJIA, however the non-financial factor models did so by a much smaller margin. Ultimately the SVM proved to be the most effective in terms of risk and return, its Sharpe ratio was the highest reflecting the most efficient use of the extra risk the model took on to achieve the excess returns. In chapter 6 the accuracies for the SVM and GP will be tested for statistical significance. The obtained results for investment returns are not entirely accurate as transaction costs were ignored. However, because the trading strategy was semi-active and only made trades on a month to month basis, and only if required, the transaction costs would be less inhibitory to overall profits than that of other more active trading approaches.

## Chapter 4

# Multi-objective Optimization with an Evolutionary Artificial Neural Network for Financial Forecasting<sup>5</sup>

### 4.1 Introduction

Two traditional methodologies for classifying investment returns are direction and relative performance to a relevant benchmark. An inherent problem with classification is that the magnitude of the price change is not considered and therefore a more accurate model does not guarantee a higher return. To combat this short coming a multi- objective approach is taken that trains the algorithm on movement and magnitude. To help the algorithm insulate itself from excess risk it is trained to indentify behaviour in the economy that is indicative of larger swings in the stock market. The inputs to the system are macro-economic data that are readily available to the public. This includes, among others, information pertaining to inflation (consumer price index) and corporate bond ratings. There are several advantages to using evolutionary algorithms for multi-objective optimization as outlined by Abraham and Jain [39], one being their ability to escape local minima and efficiently explore large and complex solution space. Our approach is an implementation of the NeuroEvolution of Augmenting Topologies (NEAT) [40] method, which starts with a population of simple perceptrons and gradually evolves more complex network structures. The NEAT method has been effective at solving problems in several domains such as video gaming [41] and automobile crash warning systems [42]. The NEAT method is attractive for financial forecasting due to the complex nature and non-linearity of stock market returns. The optimal neural network topology is not known a priori; therefore the optimal structure is more likely to be evolved slowly taking advantage of NEAT's characteristics of complexification and speciation. Our implementation included a multi-objective approach and the use of back-propagation to update the weights of the fittest individuals in each generation and as a mutation operator.

---

<sup>5</sup> Based on a co-authored paper by M. Butler and A. Daniyal. In GECCO 09: Proceedings of the 2009 conference on Genetic and Evolutionary Computation, pages 1451-1457. The contribution to this paper that is not that of the author of the thesis is the implementation of the algorithm in JAVA.

Back-propagation was used with sigmoid and hyperbolic-tangent activation functions. Our trained model will create a semi-active trading system that will make predictions on the direction of market movement on a month to month basis for the Dow Jones Industrial Average (DJIA). To capitalize on all market conditions the model will take a short position in the market when it is predicted to fall and take a long position when it is expected to increase. A short position can be achieved with different financial instruments but in effect they all profit from market contractions. The goal is to achieve an investment return that is superior to the market return of the DJIA over the same time period, because we are directly investing in the market this can only be achieved by avoiding market contractions or appropriately investing to profit from them.

## **4.2 Algorithm**

The neuro-evolutionary approach implemented for the following experiments is based on the method NeuroEvolution Augmenting Topologies (NEAT). In NEAT node creation and connection weights are preformed with evolutionary strategies, however in this implementation, back-propagation (a greedy search strategy) is included as a mutation operator in addition to the canonical use of the algorithm. This section will outline the areas of the algorithm in the context of how they were implemented, although the algorithm is based on NEAT, it may slightly differ from other implementations. A high level overview of the algorithm is as follows:

1. Generate initial population of perceptron style ANNs.
2. Repeat the following steps for a user defined number of generations:
  - a. Perform fitness evaluation for each individual in population.
  - b. Perform speciation.
  - c. Create marker for the fittest individual and call it “fittest\_ANN”.
  - d. Perform backpropagation on “fittest\_ANN” if enabled.
    - i. Replace “fittest\_ANN” with trained “fittest\_ANN” if its fitness is higher.
  - e. If current generation < maximum number of generations then create new population.

The details of each sub-procedure are explained in the following sections, this includes the parameters passed to each function and the data-structures used.

#### **4.2.1 Parameter and Data-structure description**

While there are several parameters to consider when training NEAT for financial forecasting, some have been deemed more important than others based on NEATs sensitivity to changes in said parameters. This section will discuss some of the more important parameters to optimize when using this implementation of NEAT. The “probability of connection” between an input node and output node controls how likely a given input attribute is to be connected to an ANN in the initial population of perceptrons. The lower the probability the fewer inputs each ANN will have upon initialization. The “seed” which will be different for each computer will be the seed for the random number generator and for a given static set of parameters will alter the algorithms output. As with most evolutionary inspired algorithms the “offspring percentage” controls the number of individuals chosen for crossover for the next generation. The “connection mutation probability” is the probability that a given node will be mutated by adding a connection to another node. “Node mutation probability” will define the probability that an individual will be mutated by having an additional node added. If a node is created that has not been seen before it will be accompanied by a new innovation number that will be explained in greater detail later in this section. “Weight mutation probability” controls how likely an ANN will be affected by a connection weight mutation. The “weight variance” of the population will define the upper and lower thresholds that the mutation in connection weight can oscillate between. The number of connections within an individual that are mutated can be controlled by “weight mutation within single genome” which defines a percentage and as this parameter is increased can lead to destructive behaviour within the population rather than efficiently exploring the solution space. Finally the “backpropagation probability” (if enabled) will determine the amount of individuals chosen in the current population to have their connection weights trained by backpropagation (this was developed for sigmoid and hyperbolic-tangent activation functions).

In NEAT an ANN is represented by a genome, which is a sequence of genes that encodes the following information. This information is essential to allow the algorithm to

effectively evolve the population through the generations and combat the potential destructive properties of mutation and crossover.

Each genome includes:

- A unique innovation number
- A record of all neuron connections coming to and going from said neuron
- Weight of each connection
- A flag which identifies if connection is enabled or not

#### **4.2.2 Initial Population**

The initial population is generated with each individual limited to the topology of a perceptron with no hidden layers and randomly assigned weight values between nodes. As explained in section 4.2.1 the number and type of inputs to each ANN will depend on the probability of connection and as a result there will be diversity within the initial population across two dimensions the connection weights and the input attributes.

#### **4.2.3 Crossover Operation**

The unique innovation number for nodes introduced in the NEAT method is essential for combating the destructive properties of crossover. In neuro evolution and evolutionary computation in general, the inspiration for crossover is that two fit individuals in a population can be breed to produce offspring which are more fit and inherit all the learning the parents have already experienced through previous generations. Unfortunately this outcome can be difficult to realize when crossover is performed between two individuals that are not compatible, under these circumstances the offspring are often considerably less fit individuals. The unique innovation number allows for markers to be present in the population to represent sequences that are similar, whenever a new node is created which has been evolved before in any other topology it is given the same innovation number as the previous one. If the creation of said node is completely unique that it is assigned a new innovation number. The algorithm keeps a record of each innovation number to ensure that there is no duplication. Once two individuals have been chosen for crossover their innovation numbers are lined up and for each innovation number occurring in either of the genomes a copy is created in the new genome. Then

when both of the input genomes have genes with the same innovation number the operation chooses one of the genes randomly for one-point crossover.

#### **4.2.4 Mutation**

There are four possible mutation operators which can be performed: add node mutation, add connection mutation, weight mutation and backpropagation mutation.

Adding a node randomly to an ANN as it is done with “add node mutation” can be destructive as the newly created node may not make sense and can lead to the individual becoming less fit and therefore the innovation will not propagate through the population, even if it had potential to improve fitness later on. To counter act this NEAT adds a node is the following way:

1. Select two connected neurons  $n1$  and  $n2$  and interconnection weight of  $w1$
2. Disable connection between  $n1$  and  $n2$
3. Add new a neuron between  $n1$  and  $n2$  labelled  $n3$
4. Check innovation list to see if  $n3$  was created in any other topology
  - a. If duplicate assign same innovation number
  - b. If not, assign unique innovation number and add to innovation list
5. Connect  $n1$  to  $n3$  with weight  $w1$  and  $n3$  to  $n2$  with weight 1.

By adding a new node in this fashion it is less likely to disrupt the network and damage any learning which has already taken place, such that, newly created connections and neurons will be able to propagate throughout the population. The “add connection mutation” will add a connection between to neurons which are not currently connected. This mutation operator is susceptible to creating cycles within the ANN and when a new connection is introduced it must be checked first to ensure a cycle is not created. If a cycle is found the algorithm will try another connection and test for a cycle once again, this step will be repeated up to five iterations, if no acceptable connection can be formed the ANN is skipped over. For “weight mutation” an existing connection weight is chosen between two nodes and randomly changed based on the allowable variance set by the user. “Backpropagation mutation” is the only major deviation between NEAT and this implementation; in NEAT the ANNs update their connection weights through complexification. To introduce backpropagation into evolution the neurons of each ANN

chosen for this type of mutation needed to be ordered. During backpropagation the error gradient at a particular internal node cannot be evaluated unless the error gradient for every node that it has an outgoing connection to is evaluated. Backpropagation is applied when the hyperbolic tangent and sigmoid activation functions are used. For the neuron the error gradient is calculated as  $\delta_j = y_j'(y_{d,j} - y_j)$  where  $y_{d,j}$  is the desired output at neuron  $j$  and  $y_j$  is the output of the network at neuron  $j$ . For the hidden neurons the error gradient is evaluated as  $\delta_i = y_j' \sum (w_{i,j} \delta_j)$  where  $j$  is the index of neuron to which  $i$  has an outgoing connection with weight  $w_{i,j}$ . When using the *sigmoid* activation function we have  $y_j' = y_j(1 - y_j)$  while in case of *hyperbolic tangent* we have  $y_j' = 1 - y_j^2$ .

#### 4.2.5 Speciation

NEAT introduced another solution for combating the destruction caused from crossover, it is called speciation and it is used to define ANNs which are similar (the same species) and therefore are more suitable to be paired as parents. In neuro evolution smaller networks are often more fit in the early generations and would inhibit the proliferation of larger networks which could potentially solve more complex problems once given time to fine tune their topology. A similarity measure is used to determine if two ANNs are of a similar species. The equation is as follows:

$$d(g_1, g_2) = c_1 n_c + c_2 n_d + c_3 n_e$$

where,

$n_c$  = number of genes with innovation numbers common in both genomes

$n_d$  = number of genes with innovation numbers uncommon in both genomes but less than ,

$k = \min\{i_1, i_2\}$ , where  $i_j$  is the maximum innovation number in genome  $j$ .

$n_e$  = number of genes with innovation numbers uncommon in both genomes but greater than  $k$ , as defined above.

The three weight factors are supplied by the user and will control the number of species within a population. An individual is compared to the first elements in each species and if the distance lies within a pre-defined threshold then said individual is added to that species. Once an individual is added to a species the algorithm moves on to the next to ensure the same ANN does not end up in multiple species. If no suitable species is found



then it creates a new one. Total fitness sharing is utilized for each species, where each individual in a species shares the overall species fitness. This ensures that no one species will grow prohibitively large and take over the entire population.

#### 4.2.6 Fitness of an individual

The fitness of an individual is determined by the amount of incorrectly classified instances (ICI). For the single-objective optimization (SOO) the ANNs only have one output neuron while the multi-objective approach has two. The calculation of ICI is evaluated as follows:

$$ici = \frac{(n_{1,0} + n_{0,1})}{\sum_{i=1}^2 \sum_{j=1}^2 n_{i,j}} \times 100$$

where  $n_{i,j}$  refers to the number of instances of class  $i$  classified as  $j$ . The fitness of an individual under the SOO approach was evaluated as  $1/ici_{direction}$  where as for MOO it was the average over both objectives, given by  $f_1/ici_{direction} + f_2/ici_{magnitude}$ . Where  $f_1$  and  $f_2$  are user supplied weights that determine the importance of each objective.

### 4.3 Semi-active Trading System

The following section will outline our investment strategy, the macro-economic input factors and the multi-objective optimization approach.

#### 4.3.1 Trading Strategy

The trading strategy was similar to the one implemented in Chapter 3 where the model is semi-active and makes trades on a month-to-month basis and only if required. As well the strategy relies on short-selling the market when the prediction is for a contraction over the next month. For a more thorough explanation of the trading strategy please see sections 1.2.1 and 3.6.

#### 4.3.2 Data Description

The input data set was influenced by work done by Enke and Thawornwong, and was explained in Section 3.4. The only difference between the input data is that for the experiments concerned in this chapter only the monthly changes in the indicators were used, therefore the PFFM was not included.

The input data creates a multi-dimensional optimization problem, the original authors only started with an input set of this size and then applied data mining attribute selection algorithms to drill down to a more appropriate set size. This is not required for our implementation of NEAT as not all inputs are made available to the EANNs upon initialization. The algorithm parameter setting “probability of connection” allows us to control how many inputs each EANN receives during initialization and through evolution the inputs that are helpful should prevail and others which are redundant or ineffective will be bred out. This somewhat resembles traders in the real world given that different traders on an exchange floor may rely on different indicators to make their trades: some may believe in the interactions of moving averages and others on Bollinger bands but ultimately their combination may prove to be most appropriate.

#### **4.4 Multi-objective Optimization**

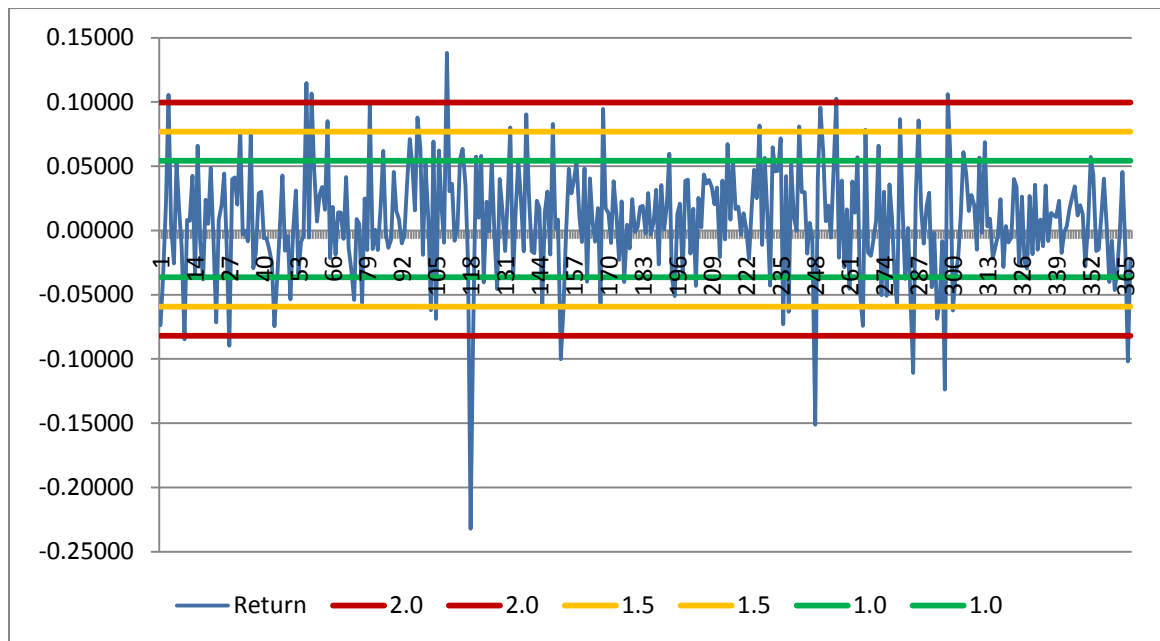
Under a single objective classification problem for financial forecasting a model is trained to recognize movement in only one dimension, did the underlying asset in question move up or down (or over/under perform a benchmark). This can cause a problem when deciding which models are superior as higher classification accuracy could yield a lower investment return if it happens to make incorrect predictions during a large gain/loss in the market. To counteract this problem we are using a multiobjective optimization approach that trains the EANN on direction and magnitude. There may be behaviour in the economy that is measurable by macro-economic indicators that suggest more volatility in the coming month. Training an ANN to recognize both of these traits will create a more robust and less risky investment model. In times of greater volatility, market prediction can be more difficult and therefore it would be an asset to be able to predict such events. Variance in the stock market represents risk, the more a stock varies about its mean return the larger its price fluctuations. This is commonly represented in a financial indicator called Beta, which represents how much a particular asset moves in relation to its benchmark. A beta of one would indicate an asset moves in unison with the market for a beta lower or greater than one would represent a less risky and more risky asset respectively.

To determine a market movement that warranted classification, basic statistics were gathered about the monthly returns and it was calculated that a monthly change that was one standard deviation above/below the mean return was considered relevant. This was based on the results that 26% of the data was above/below one standard deviation and represented 74% of the variance and therefore risk in the market during our training and testing periods. Below in figure 10 is a graphical representation of the DJIA during the training and testing time periods, showing the monthly changes in the market and the thresholds for 1, 1.5 and 2 standard deviations around the mean monthly change.

## 4.5 Experiments and Results

### 4.5.1 Training Phase

Training of the EANNs was conducted over a 23 year period that spanned from 1978 up until 2001, this represents roughly three quarters of the collected data. The rationale for the training period was the EANN would have an opportunity to see market reactions under all conditions in the business cycle and therefore be better equipped to handle the testing phase. Several runs were conducted for both the multiobjective and single objective approaches to find the optimal settings for the parameters discussed in Section 3. The



**Figure 10- Charting Magnitude of Index Returns - monthly changes of the DJIA with the thresholds for 1, 1.5 and 2 standard deviations around the mean monthly change.**

weights for each class could be set to determine influence when calculating individual fitness, after several training runs it was decided that a 50/50 split would be most appropriate, giving each class equal importance. Table 6 lists the optimal settings for both algorithms. Backpropagation has a probability of zero as the optimal settings were achieved with an activation function that was not appropriate for backpropagation to be applied.

**Table 6 – EANN Parameter Settings**

<b>Name</b>	<b>Multi</b>	<b>Single</b>
Population Size	200	100
Number of Generations	100	100
Initial Percentage Connection	.30	.30
Seed	100	50
Offspring Percentage	70	80
Node Mutation Probability	.70	.60
Weight Mutation Probability	.80	.70
Connection Mutation Probability	.75	.85
Weight Variance	5	2
% of Genes affected in Weight Mutation	.50	.60
Backpropagation Probability	0	0
Activation Function	Step	Step

Reported from the training phase are overall accuracy for direction and magnitude, investment return, Sharpe ratio and the fraction of correctly classified market contractions to incorrect. This last figure is important due to the aggressive nature of our investment strategy, if this percentage is less than 50% it is less likely that the model will outperform the market. The accuracy for magnitude is a count of correct predictions for direction during a higher magnitude change in the market.

The results from the training phase are shown in table 7, where the cumulative investment return is based on an initial \$1000 investment. For simplicity reasons, transaction costs are ignored.

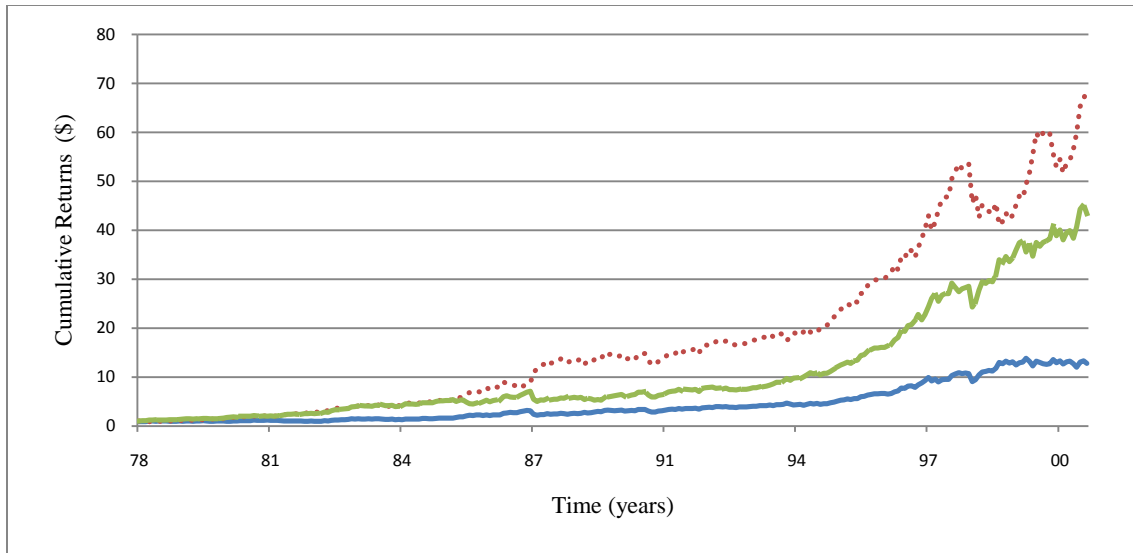
**Table 7 – EANN Training Results**

<b>Performance Measure</b>	<b>Multi</b>	<b>Single</b>
Accuracy (Direction)	68.44%	71.28%
Accuracy (Magnitude)	69.33%	62.67%
% of accurate contraction predictions	61.00%	67.04%
Number of contraction predictions	100	88
Yearly Investment Yield (%)	18.16%	16.15%
Cumulative Investment Return (\$)	\$69,237	\$43,373
Yearly Excess Return to the Market <sup>6</sup> (%)	7.26%	5.25%
Sharpe Ratio	1049.73	768.66

The training results are a good example of how a higher accuracy does not always translate to better investment returns. The SOO model had a slightly higher overall accuracy for predicting market direction but considerably lower yearly investment return. This is reflected in magnitude accuracy where the SOO model did not perform as well as the MOO. The Sharpe ratio was higher for the MOO model, representing a more efficient use of the extra risk the model was exposed to. As stated earlier, the Sharpe Ratio used the market rate instead of risk-free rate, and as a result the lower value for the SOO is reflected in the lower percentage of accurate magnitude predictions. In figure 11 we have a plot of cumulative investment returns over the training phase. It displays the advantages of the MOO methodology and the trading style. Short selling the market allows both portfolios to grow, even in times when the market is relatively flat.

---

<sup>6</sup> DJIA yearly return during the training period was 10.90%.



**Figure 11 – MOO and SOO Training Data Investments Comparisons - comparison between cumulative investment returns, shown in thousands, of the multiobjective model (red dots), single-objective model (green dashes) and the DJIA market return (solid blue) for the training set.**

#### 4.5.2 Testing Phase

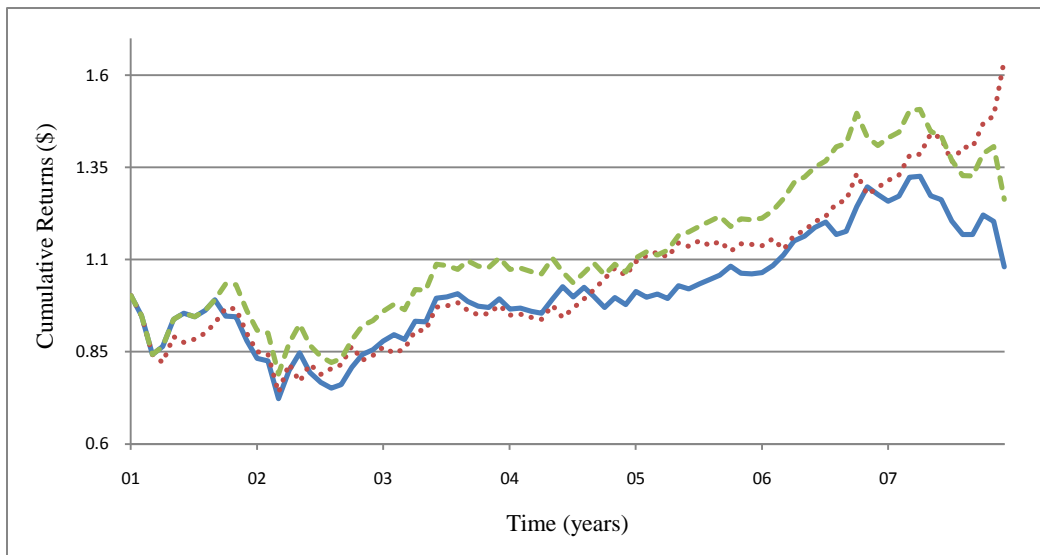
The fittest EANNs from the training phase were chosen to be applied to out of sample testing data which contained 84 periods spanning from 2001-2008. The results are displayed in Table 8. Cumulative investment return is based off an initial \$1000 investment at the beginning of the testing phase.

**Table 8 – EANN Testing Results**

Performance Measure	Multi	Single
Accuracy (Direction)	63.10%	58.34%
Accuracy (Magnitude)	58.82%	47.06%
% of accurate contraction predictions	63.33%	62.50%
Number of contraction predictions	30	16
Yearly Investment Yield (%)	7.10%	3.34%
Cumulative Investment Return (\$)	\$1641	\$1264
Yearly Excess Return to the Market <sup>7</sup> (%)	6.00%	2.24%
Sharpe Ratio	13.78	8.99

<sup>7</sup> DJIA yearly return during the testing period was 1.10%.

The out of sample testing data provided a more realistic representation of the abilities of an EANN. The overall investment returns suggest that the EANN models were able to outperform the market, however transaction costs were ignored and although we attempted to minimize them, they would negatively impact the reported results. However the MOO, proved to be superior in terms of accuracy, investment return and risk-adjusted investment return in comparison to the SOO. The SOO method made only about half the market retraction predictions, meaning that it was often incorrect in predicting market expansion, but only suffered the same loss as the DJIA. This is a more conservative approach and in the short-term was performing better. However the MOO model was able to make more correct predictions in times of larger movement on the market and capitalized more efficiently on its trades. Figure 12 plots the cumulative investment returns over the testing period.



**Figure 12 - MOO and SOO Testing Data Investments Comparisons - comparison between cumulative investment returns, shown in thousands, of the multiobjective model (red dots), single-objective model (green dashes), and the DJIA market return (solid blue) for the testing set.**

#### 4.5.3 Results for Backpropagation

Unfortunately the more robust EANNs were created with a step activation function and therefore backpropagation could not be applied. Where appropriate, backpropagation was applied in two ways: (1) as a mutation operator and (2) applied to the highest ranking EANN from each generation. In the later application the EANN trained with back-

propagation was only included in the following generation if it was superior to its previous untrained topology. The results indicated that the application of backpropagation tended to over fit the data leading to exceptional training results but very poor testing accuracy. In some cases the backpropagation also lead to the algorithm getting trapped in local minima during evolution which lead to inferior results even for the training set. This behaviour was exhibited by both the sigmoid and hyperbolic tangent activation functions.

#### **4.6 Discussion and Conclusions**

Presented in this chapter is an attempt to create a robust trading model for the DJIA from an evolutionary artificial neural network trained for multiobjective optimization. The main contribution was to show that an EANN trained to recognize direction and magnitude in the stock market was better equipped to create superior investment returns than that of one trained only to recognize direction changes. The advantages to using an implementation of NEAT for evolving the neural networks is apparent, as in both the training and testing data the EANNs were able to produce comparable investment returns to the market index. Complexification showed to be better left to natural evolution as the implementation of backpropagation to fine tune the weights was destructive and lead to over fitting and local minima.

Based on the training and testing results the EANN trained for multiobjective optimization is more robust and better equipped to make market predictions. Having the extra dimension of learning the behavior of the economy that predicates larger than normal movements in the market was an advantage and on more occasions the MOO EANN was able to take advantage of these returns. This gives weight to the hypothesis that the information from the macro-economic indicators is non-monotonic and dependent on each other. A rise in one indicator could mean a contraction or an expansion in the market depending on the results of other factors.

In the testing phase the overall accuracy was 63.10% for the top performing EANN. Given that the investment results outperformed the market it seems that the EANN was better than just random guessing. Inaccurate predictions can be attributed mainly to the incomplete and noisy data that is inherent in the financial domain. The



macro-economic indicators are able to partly explain the relationship between time  $t$  and  $t - 1$  market prices but do not reflect all information available to investors. To create a more robust model other measurable factors of market forces have to be considered. Information pertaining to the individual companies was not included and it is most likely vital to include this information to build on the results obtained in this section.

## Chapter 5

# Financial Forecasting Using Character N-gram Analysis and Readability Scores on Annual Reports<sup>8</sup>

### 5.1 Introduction

The Securities and Exchange Commission (SEC) requires that each year all publicly-traded companies supply a third-party audited financial report, which states the company's financial position and performance over the previous year [43]. Contained in these annual reports, inter alia, are financial statements, a letter to the share-holders, and management discussion and analysis. Over the years several research endeavors have been focused on the numbers contained in the financial statements, computing a variety of ratios and price projections without considering textual components of the reports. Peter Lynch, a famous investment "guru," once said that "charts are great for predicting the past," pointing out that there is more to making good investments than just processing the numbers. The textual components give insight into the opinions of the senior management team and provide a direction of where they feel the company is going. This information should not be trivialized or overlooked; it should be processed in a similar way to processing quantitative information, to extract meaningful information to aid in the forecasting process. Up until recently an analyst would have to read an annual report and use their expertise to determine if the company is going to continue to do well or if there is trouble ahead. They would apply their skill and judgment to interpret what the Chief Executive Officer (CEO) is saying about the company and its direction for the future. This process can be very time consuming and it is a somewhat heuristic approach, considering that two experienced analysts could read the same report and have a different feeling about what it is saying. If an analyst has several companies to consider and even more annual reports to read it could be difficult to take in all the relevant information when it is most likely surrounded by noise and other erroneous information that has no effect on the stock price. Most numeric calculations can be automated to remove human

---

<sup>8</sup> Based on a published co-authored paper by M. Butler and V. Kešelj. Canadian AI 2009, LNAI 5549, pp. 39-51.

error, and complex data mining and machine learning algorithms can be applied to extract meaningful relationships from them. It would be extremely valuable if the same could be done for the textual components, having a quick, efficient and accurate tool to analyze an annual report and make recommendations on its implications for the stock price over some given time period. This could erase some of the subjective judgments that arise from an individual's interpretation of the report, which could change from person to person. Also, given the sheer amount of annual reports that are produced each year, one would be able to analyze a larger number of companies and have a greater opportunity to find good investments. In this paper an attempt is made at achieving this goal: two novel approaches to analyzing the text are put forward and then a combined model is also analyzed to see if a union of these approaches is more robust. The first novel technique is to convert the textual components to n-gram profiles and use the CNG distance measure to classify reports. The second is to generate three readability scores (Flesch, Flesch-Kincaid and Fog Index) for each report, and after combining with the previous year's performance, make class predictions using a support vector machine (SVM) method. The combined model will only make a recommendation on a particular annual report when the two models are in agreement; otherwise, the model outputs no decision. The models make predictions whether a company will over- or under-perform S&P 500 index over the coming year. This is an appropriate benchmark as all the companies being analyzed are components of this index. We believe that this is a very meaningful comparison. In some published results, performance of an algorithm was evaluated by measuring how accurately one can predict increase or decrease of a stock price. This evaluation approach may lead us to believe that an algorithm has a good performance, while it may be worse than the index performance. Hence it would be useless to an investor, who could simply invest in the index, achieve higher return, and be exposed to lower risk.

## **5.2 Data Pre-processing**

### **5.2.1 Data Collection**

There are no known publicly available data sets that would contain a preprocessed sample of annual reports to analyze, so the data set was created from scratch. To facilitate this, the website of each company considered was visited and the relevant annual reports were downloaded from the investor relations section. Prior to downloading, every report's security features were checked to ensure the PDF was not protected; if it was, then it was discarded as the file could not be converted to text (text format is required to apply n-gram and readability programs). Once a sufficient sample size of annual reports was collected, they were converted to text using a Perl script with program pdftotext.

### **5.2.2 Data Labelling**

The most sensitive and time consuming process of the experiment was class labeling of the training and testing data. It is not mandated by the SEC that companies file their annual reports at the same time, so as a result, each performance measure has to be individually calculated for each company, based on different months. To expedite this process, a matrix of relative returns was created based on monthly closing prices for each stock from data obtained from Yahoo! Finance [44]. The returns for each month were calculated as a numeric figure, and introduced as a class attribute as either over or under performing the S&P 500 over the trailing 12 month period. Next, the filing date for the reports was captured from the SEC website and the appropriate text file is labeled. This was done manually for each report.

### **5.2.3 Generating N-Gram Profiles**

The n-gram profiles were created as defined by the CNG method using the Perl n-gram module Text::Ngrams developed by Kešelj [45]. The character six grams and word tri-grams were used, and various profile lengths up to 5000 unique, normalized, most-frequent n-grams from an annual report were used.

### 5.2.4 Generating Readability Scores

A Perl script was created that generated the three readability scores from source code developed by Kim Ryan [46] and made publicly at CPAN [47]. The scores for each annual report are combined with the underlying securities' 1-year past performance to form the input attribute set for the SVM. The previous year's performance was represented in two ways: first by its relative performance to the S&P 500, and by an indicator whether or not it decreased or increased in value over the last year. To make the data appropriate for the SVM it was scaled between 0 and 1 to cut down on computation size and transformed into the required format. The three readability scores considered were the Gunning Fog Index, Flesch Reading Ease, and Flesch-Kincaid Grade Level. The Gunning Fog Index developed by Robert Gunning in 1952 is a measure of readability of an English sample of writing, the output is a reading level that indicates the number of years of formal education required to understand the text, and the equation is as follows:

$$\text{Gunning Fog Index} = 0.4 * \left( \left( \frac{\# \text{ words}}{\# \text{ sentence}} \right) + 100 \left( \frac{\# \text{ complex words}}{\# \text{ words}} \right) \right)$$

where *#words* is the number of words in text, *#sentences* number of sentences, and *#complex words* number of words that are not proper nouns and have three or more syllables. The Flesch Reading Ease (FRE) and Flesch-Kincaid Grade Level (FKL) were both created by Rudolph Flesch. The higher the FRE score the simpler the text and the output for the FKL is similar to the Gunning Fog Index, where it generates a Grade Level that reflects the number of years of formal education required to understand it. The two scores are imperfectly correlated and therefore it is meaningful to consider them both. Their respective equations are given below:

$$\text{Flesch Reading Ease} = 206.835 - 1.015 \left( \frac{\text{total words}}{\text{total sentences}} \right) - 84.6 \left( \frac{\text{total syllables}}{\text{total words}} \right)$$

$$\text{Flesch - Kincaid Grade Level} = 0.39 \left( \frac{\text{total words}}{\text{total sentences}} \right) + 11.8 \left( \frac{\text{total syllables}}{\text{total words}} \right) - 15.59$$

The algorithm for syllable count was implemented as the Perl module `Lingua::EN::Syllable` [48], with estimated accuracy of 85–90%.

### 5.3 CNG Classification of N-Gram Profiles

The n-gram classification technique was inspired by work done by Kešelj, Peng, Cercone and Thomas (2003) [14], where n-gram profiles were used, with a high degree of accuracy, to predict author attribution for a given unlabeled sample of writing. A generalized profile for a given author was generated and then used to gauge a distance calculation from new testing documents. For financial forecasting a general n-gram profile was created from all of the company annual reports for a given class. The classifier would concatenate all the files from one class or another and then generate one overall n-gram profile with the same settings as discussed in the data pre-processing subsection. For each testing year  $x$  the training profiles would be generated from years  $x - 1$  and  $x - 2$ . Once the two generalized profiles are created, one for over-performing and one for under-performing stocks, the profiles of documents from the testing year are compared with the training profiles using the CNG distance measure:

$$\sum_{s \in profiles} \left( \frac{f_1(s) - f_2(s)}{\frac{f_1(s) + f_2(s)}{2}} \right)^2$$

where  $s$  is any n-gram from one of the two profiles,  $f_1(s)$  is the frequency of the n-gram in one profile, or 0 if the n-gram does not exist in the profile, and  $f_2(s)$  is the frequency of the n-gram in the other profile.

### 5.4 SVM Classification with Readability Scores

The input attributes to the SVM method were vector representations of the annual reports that contained the three readability scores and the stock's performance over the previous year. An SVM is a very robust classifier that has proven effective when dealing with highly complex and non-linear data, which is indicative of data found in the financial domain. For a more thorough description of SVM classification and the method used in this paper, please see section 3.4.2. SVM's had been widely experimented with financial forecasting in both classification [49] and level estimation or regression [50] domains. Because the scores are not time sensitive and the SVM does not take into account any time dependencies when evaluating the data, all of the vector representations were used to train the system, except for the particular year it was tested on at any given time. The Support Vector Machine environment utilized was LIBSVM. A polynomial

kernel of degree 3 was used, with the c-SVM approach; i.e., the use of slack variables to allow for “soft” margin optimization. Five input attributes are used in SVM classification: three readability scores from annual reports, and two performance measures in the previous year: one whether the stock over or under performed, and the second whether the stock price increased or decreased in the previous year.

## 5.4 Experiment Results

In general all three individual models and the two combinations performed well and overall, they each outperformed the benchmark return in the testing period. To display the results, a special attention is given to the three criteria: overall accuracy, overperformance precision, rate and investment return. Over-performing precision is a point of interest on its own as positive predictions classify a stock as a future over-performer, and therefore would initiate an investment in the market. This opens the portfolio up to potential losses since an actual position has been taken. However, when the model predicts an underperforming stock, it passes it over for investing and when the prediction is wrong it is only penalized by missing out on a return—an opportunity cost and not an actual dollar loss. Next, we look at each model’s performance individually, and then on some comparisons between them and the benchmark. The benchmark portfolio consists of an equal investment in all available stocks in each of the testing periods. The S&P 500 was not used as the experiment sample did not include all underlying assets in the S&P 500 index. Table 1 displays comparative models’ performance year over year for percentage return, cumulative dollar returns and accuracy, and over- and underperformance precision of the model.

**Character N-grams with CNG (C-grams)** method outperformed the benchmark portfolio return overall and in five of the six years.

**Word N-grams with CNG Classification (W-grams)** model had superior accuracy and over-performance precision to that of the character n-gram model, and it also outperformed the benchmark return.

**Readability Scores with SVM (Read)** performed well, and in all but one year outperformed the benchmark and the n-gram model.

**Combined Readability-scores with Character N-grams (Combo-char)** makes a recommendation only when there is an agreement between the two combined methods. In

addition to previously mentioned measures, for the combined models we also consider the percentage of cases with no decision due to the disagreement of the models.

**Combined Readability-scores with Word N-grams (Combo-word)** performed better than the benchmark, but significantly worse than the Combo-char model.

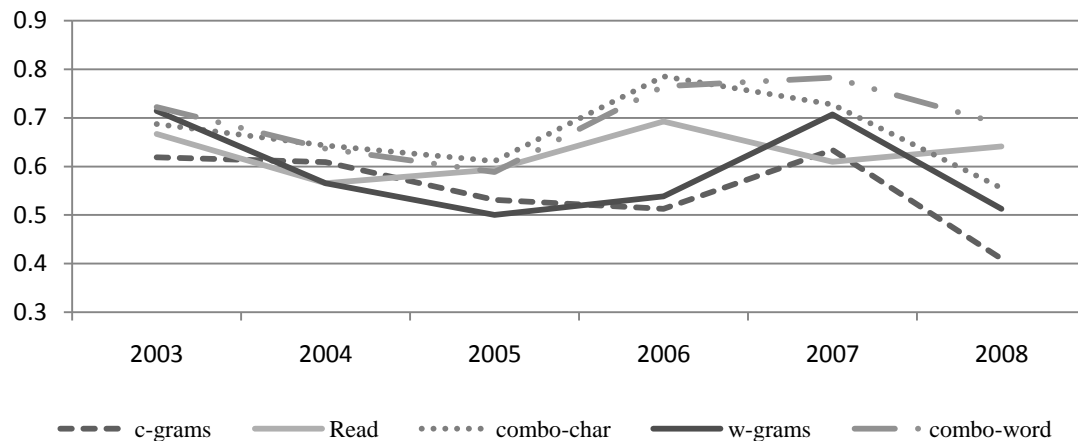
**Table 9 – Detailed Experiment Results for NLP Based Models**

<b>Character N-gram Model</b>						
Year	Return (% and \$)		Accuracy	Over-perf.	Under-perf.	No Decision
2003	-6.59%	\$9341.18	61.91%	70.59%	25.00%	
2004	47.80%	\$13806.26	60.87%	65.00%	33.33%	
2005	20.32%	\$16611.11	53.12%	52.63%	53.85%	
2006	31.48%	\$21839.65	51.28%	52.38%	50.00%	
2007	34.67%	\$29410.73	63.41%	75.00%	58.62%	
2008	-10.33%	\$26371.62	41.02%	26.67%	50.00%	
Overall	163.72%	\$26371.62	55.27%	57.04%	45.13%	
<b>Word N-gram Model</b>						
Year	Return (% and \$)		Accuracy	Over-perf.	Under-perf.	No Decision
2003	-3.00%	\$9700.00	71.43%	80.00%	50.00%	
2004	50.53%	\$14601.35	56.52%	64.71%	33.33%	
2005	15.82%	\$16911.02	50.00%	50.00%	50.00%	
2006	27.94%	\$21636.71	53.85%	55.56%	47.62%	
2007	36.60%	\$29555.75	70.73%	80.00%	65.38%	
2008	-9.29%	\$26808.80	51.28%	41.18%	59.09%	
Overall	168.09%	\$26808.80	58.97%	61.91%	50.90%	

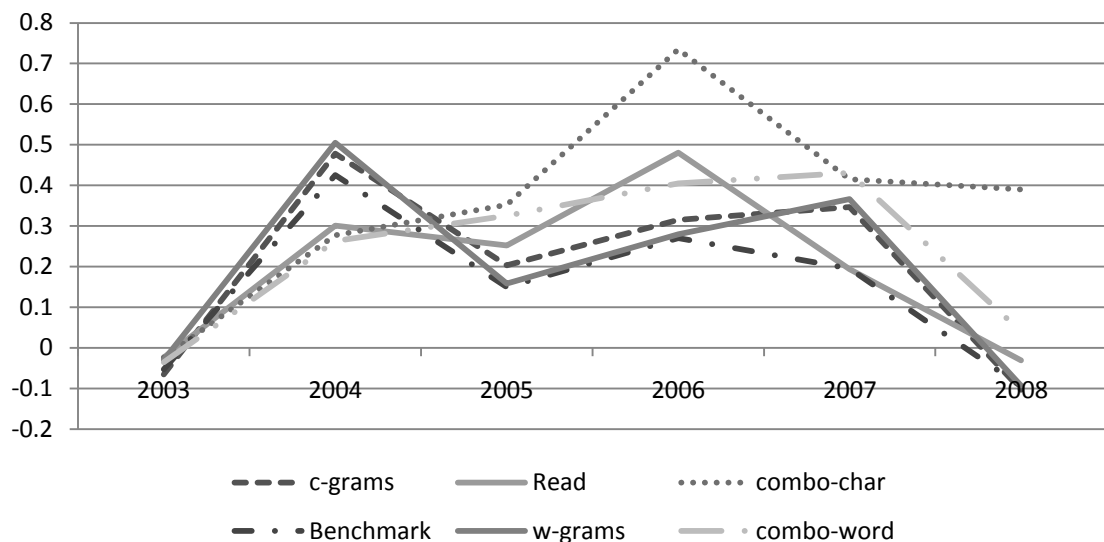
(table continued on the next page)



<b>Readability Model with SVM</b>						
Year	Return (% and \$)		Accuracy	Over-perf.	Under-perf.	No Decision
2003	-2.42%	\$9758.33	66.67%	81.82%	44.44%	
2004	30.07%	\$12692.34	56.52%	66.67%	37.50%	
2005	25.23%	\$15894.71	59.38%	61.54%	57.89%	
2006	48.06%	\$23534.11	69.23%	75.00%	65.22%	
2007	19.33%	\$28084.04	60.98%	59.26%	64.29%	
2008	-3.13%	\$27206.41	64.10%	62.50%	64.52%	
Overall	172.06%	\$27206.41	62.81%	67.80%	55.64%	
<b>Combined Readability and Character N-grams</b>						
Year	Return (% and \$)		Accuracy	Over-perf.	Under-perf.	No Decision
2003	-2.42%	\$9,758.33	68.75%	83.33%	5.88%	5.60%
2004	27.69%	\$12,460.64	64.29%	61.54%	25.49%	9.97%
2005	35.22%	\$16,849.56	61.11%	66.67%	9.80%	7.72%
2006	73.50%	\$29,233.98	78.57%	83.33%	8.82%	7.54%
2007	41.50%	\$41,366.08	72.73%	90.00%	11.44%	9.09%
2008	39.00%	\$57,498.85	55.56%	100.00%	1.04%	1.06%
Overall	474.99%	\$57,498.85	66.83%	76.47%	62.48%	6.83%
<b>Combined Readability and Word N-grams</b>						
Year	Return (% and \$)		Accuracy	Over-perf.	Under-perf.	No Decision
2003	-3.55%	9,645.45	72.22%	83.33%	50.00%	14.29%
2004	26.30%	12,182.20	63.64%	60.00%	100.00%	52.17%
2005	32.50%	16,141.42	58.82%	70.00%	42.86%	46.88%
2006	40.50%	22,678.70	76.47%	66.67%	81.82%	75.86%
2007	43.08%	32,449.44	78.26%	91.67%	63.64%	43.90%
2008	4.00%	33,747.42	68.75%	100.00%	66.67%	58.97%
Overall	237.47%	33,747.42	69.69%	76.47%	65.68%	48.68%



**Figure 13 – NLP Based Models Year over year accuracy**

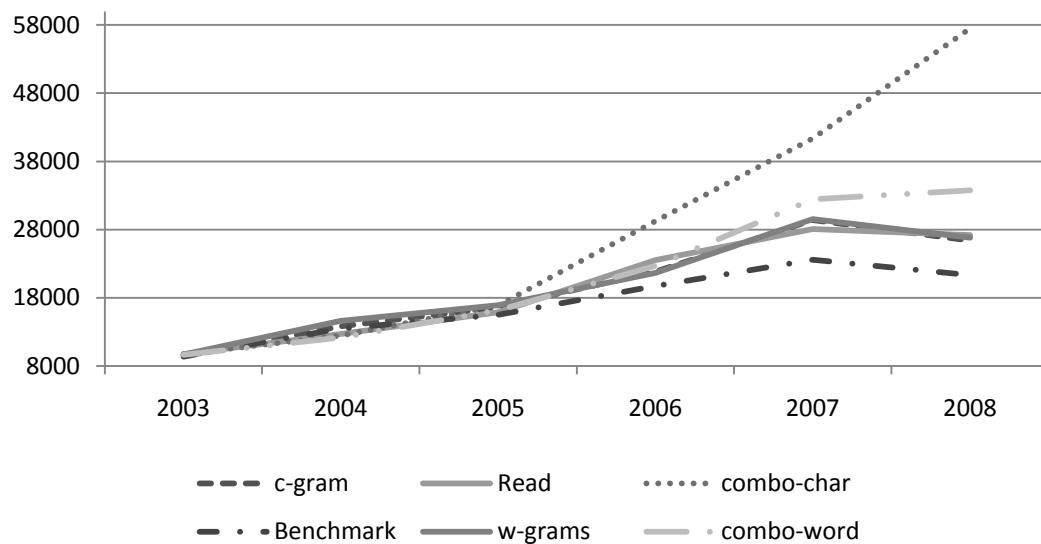


**Figure 14 – NLP Based Models Year over year percentage returns**

## 5.6 Model Results Comparisons

To adequately compare the models we present in this subsection performances graphically on a combined plot. Figure 13 plots the year over year percentage accuracy of the five models. We can see that the word-combo model had better accuracy in all six years including 2008 when the market experienced a major trend shift. It is worth noting that the character-gram model slipped below the 50% margin in the last year during the trend change in 2007–2008. This was the only occurrence of any of the models performing below 50% accuracy. Figures 14 and 15 chart the percentage return and

overall dollar return respectively for the five models and the benchmark portfolio. Comparing the plots between the models and the benchmark portfolio it appears that their trends all match a general shape, only that in the majority of the years the benchmark is the poorest performer. In 2008 the only models to produce a positive return were the combined models and this was achieved when the benchmark lost nearly 10%. By a large margin the character n-gram combination model had the superior investment strategy. For the first three years all 4 portfolios were quite close but in 2006 the character n-gram combination model pulled away and in 2008 picked up its most significant relative gain. This 2008 return is a direct result from the benefit of having a perfect over-performance precision rate.



**Figure 15 – NLP Based Models Cumulative Investment Returns - displays investment returns in dollars, with initial investment of \$10,000.**

## 5.7 Discussion and Conclusions

In general, the endeavour put forth here is an attempt to automate the analysis of annual reports. The expected benefit is that one could quickly evaluate the textual component and remove some of the uncertainty that arises from analysts having different opinions. More specifically, two novel NLP techniques are applied to solving the aforementioned problem. This section details the results, and gives some explanations as to what worked and what did not.

### **5.7.1 N-Grams with CNG Classification**

It has been shown that this methodology can be effective the problem of authorship attribution. In changing from the authorship attribution task to recognizing language indicative to one type of behaviour to another is a bit of a stretch. The belief is that certain language and phrases are used when the outlook is bleak and is measurably different than that when the outlook is positive. Overall, both the n-gram models were the weakest of the five models constructed, however they were still superior to the benchmark portfolio and that fact alone makes the experiment a success. The two n-gram based models had similar results, with the word-grams performing slightly better in overall accuracy and investment return. Although neither n-gram approach could capture all the information in the report, it was able to model a portion of it, such that, sufficient enough to give above average returns. The n-grams proved to be least effective when the market trend drastically shifted in 2007–2008. This may not necessarily be a short-coming of the n-grams themselves but the classification approach applied to them. It would be interesting to use a SVM for the n-gram profiles as a comparison to the CNG method. The overall accuracy of the models were about 55% and 59% for character-grams and word-grams respectively which is quite typical of investment models and is good evidence that it is better than random guessing.

### **5.7.2 Readability Scores with SVM**

As noted earlier, SVM's have proven very effective at producing robust investment models and dealing with the highly complex and non-linear data that is inherent in financial forecasting. Part of the success of this model could be attributed to the SVM choice of the classifier. Based on our preliminary tests, some other algorithms such as Artificial Neural Networks or Naïve Bayes could not achieve the same accuracy. Readability scores and their relation to stock performance have been well documented and the favourable results of this method are not unexpected as this model combined a proven linguistic analysis technique with a powerful classification algorithm. This model outperformed the n-grams technique and the benchmark portfolio on investment return (percentage and dollars) and in over-performs precision, which made for more efficient trades. The overall accuracy and over-performance precision was 62.81% and 67.80% respectively, giving evidence that the model was more than just random guessing. This

technique also demonstrated an ability to partly understand the text in the annual reports and learn what it indicated for future performance.

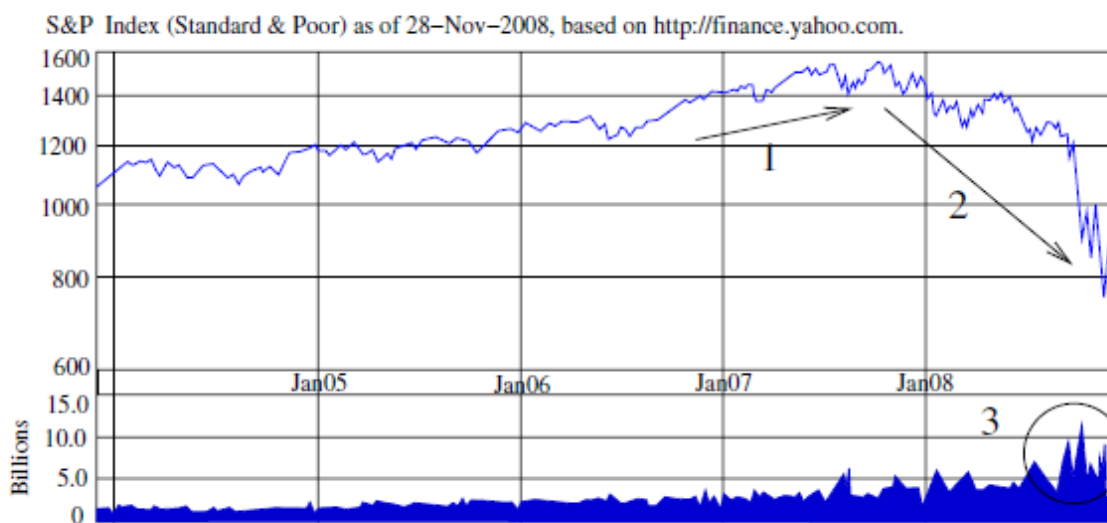
### **5.7.3 Combined Models**

Choosing to only make decisions when the models agreed proved to be a valuable approach. This approach could be characterized as an ad hoc ensemble approach. It is evident that the three individual models were each able to explain part of the relationship between performance and the textual components of the annual reports and that what they learned was not completely overlapping. The combined models consistently outperformed the individual models and the benchmark portfolio. The combined models were also the most efficient as they made only about half the number of trades as the other three. This fact is evident from the “no decision” figures in table 9, where on average 40% (character n-grams combo) and 48% (word n-gram combo) of the time the two models did not agree and therefore no position was taken. Having the two models agree introduced a further confidence factor into the combined model which makes it more robust to noise in the market. In the majority of the years and overall the combined models proved superior in terms of investment return (dollar and percentage), over-performance precision, accuracy, and efficiency of investments. The most significant difference came in 2008 when the other three portfolios posted negative returns and the combined models made a positive gain of 39% (character n-gram combo) and 4% (word n-gram combo). It is also interesting that in this year the character combined model was not as accurate as the Readability model but it did, like the word n-gram combo, have a perfect 100% for over-perform precision and therefore made no poor choices when an actual position in the market was taken. This abnormal investment return in 2008 is a bit of an anomaly and is not entirely realistic and will be discussed in the next section.

### **5.7.4 The 2008 Investment Anomaly**

An over-perform precision of 1 and an investment return of 39% or 4% when the market losses almost 10% seems very good, however the problem is the models are suppose to build a portfolio of investments to spread the risk. Due to the volatile nature of the markets in 2007-2008 the two models were only able to agree once on an over-performer and therefore only made one investment each in the market. In reality an investment

manager would most likely not have accepted this response and either moved some of the assets to the money market or conducted further analysis on the companies to find other suitable investments. The annual reports that the 2008 returns are calculated from are the 2006 annual reports produced sometime in 2007. Figure 16 illustrates the massive shift of market momentum in 2007. The arrow labelled '1' represents the time period when the 2006 annual reports were being published and arrow '2' represents the time period when the actual performance was being evaluated. It is quite clear that the market environment drastically changed between those two time periods and the increase volatility is supported by the large increase in market volume highlighted by the circle labelled '3'.



**Figure 16 - S & P 500 Index Returns**

## 5.8 Drawbacks and Limitations

Although the results are persuasive that the techniques presented are effective at analyzing annual reports, there still is a need for more thorough testing with an expanded data set that contains more of the companies in the S&P 500 index. The n-gram profiles were set size 6 and 3 for the character grams and word grams respectively taking up to the top 5000, these settings are most likely a local optimum and require fine tuning to optimize the model. With all the recent turmoil and volatility in the financial markets it will be worth applying the models to the newly released annual reports over the coming year to see how the models hold up under such extreme conditions. There is also a lot of information that is generated and can be learned from the experiment and deeper drilling down through the data could reveal more interesting information. For example, it would

be interesting to know if there are some companies that produce more easily read annual reports making them more transparent, and therefore a safer investment, or if the distance scores that the CNG classifier reports is an indication of how sure the model is and could a threshold be introduced to improve overall accuracy and overperform precision. Additionally the labelling process should be automated to cut down on pre-processing time and human error. Finally the analysis could be fine tuned to only include relevant portions of the reports that would contain forward looking statements and other discussion related portions.

## Chapter 6

### Results Analysis and Discussion

This chapter will detail the results from the additional statistical tests that were performed on each of the experiments conducted in chapters 3-5, they will be in reference to the null hypotheses outlined in chapter 1 and restated here. All the statistical tests were performed with a 95% significance level on a one-sided t-test with binary distributions. When comparing the outputs of two algorithms a two-sample t-test is used.

#### 6.1 – Pseudo financial factor modeling

The null and alternative hypotheses for this chapter are:

$H_0$  – *Trading models developed from support vector machines and tree-based genetic programming with the aid of the PFFM are not superior to those only utilizing monthly changes in their input data indicator set.*

Thus the corresponding alternative hypothesis is:

$H_1$  – *Trading models developed from support vector machines and tree-based genetic programming with the aid of the PFFM are superior to those only utilizing monthly changes in their input data indicator set.*

The results from chapter 3 provide evidence that superior investment returns are obtained when the algorithms are trained with the aid of the pseudo financial factor model. The investment returns are generated from the output of the algorithms, either a correct or incorrect prediction. To further gauge the superiority of the pseudo financial factor model a statistical test has been performed, the results are listed below in Table 10.

From the results displayed in Table 15 we can conclude that the algorithms trained with the pseudo financial factor model produced statistically significant results in terms of accuracy. In addition, the results between the algorithms are not statistically significant, meaning that the pseudo financial factor model aided each algorithm with similar



**Table 10 – P-values for the pseudo financial factor models - p-values generated from a one-sided t-test for various binary distributions.**

Test	p-value
SVM-Factors vs. SVM-No Factors	0.034404
GP-Factors vs. GP-No Factors	0.036312
SVM-Factors vs. GP-Factors	0.320404
SVM-No Factors vs. GP-No Factors	0.329258

benefits. From these results we can reject the null hypothesis and therefore accept the corresponding alternative hypothesis  $H_1$  that the utilization of the PFFM produced statistically significant superior returns.

## **6.2 – Multi-objective optimization of stock markets returns**

The null and alternative hypotheses for this chapter are:

$H_0$  – *The results obtained from training the EANN with a multi-objective approach are not superior from those obtained under a single-objective optimization.*

Conversely the alternative hypothesis is:

$H_1$  – *The results obtained from training the EANN with a multi-objective approach are superior from those obtained under a single-objective optimization.*

The objective of this research was to investigate if whether or not magnitude was an important behaviour to model in the stock market in terms of maximizing cumulative investment returns. The results presented in Chapter 4 were convincing that a multi-objective approach utilizing magnitude and direction was superior to the single-objective optimization where only directionality was modeled. In terms of statistical significance there is not a sufficient sample size for magnitude to make the analysis relevant. To counteract the lack of sample size the algorithms have been retrained and tested over an extended out-of-sample data set. The details of the training results are displayed below in table 11.

**Table 11 – EANN Training Results – displays the training results for the shortened training data spanning 250 consecutive months from January 1978 to October 1998.**

<b>Performance Measure</b>	<b>Multi</b>	<b>Single</b>
Accuracy (direction)	65.60%	68.80%
Accuracy (magnitude)	66.13%	59.68%
% of accurate contraction predictions	56.63%	65.57%
Number of contraction predictions	83	61
Yearly investment yield (%)	15.64%	14.55%
Cumulative investment return (\$)	\$25,458	20,367
Yearly excess return to the market (%)	4.38%	3.29%
Sharpe Ratio	296.55	285.38

The training results reflect similar short comings of the SOO approach where accuracy in terms of magnitude determines the model with the higher investment returns. In table 12 the testing results using the same metrics are reported and will subsequently be used to test for statistical significance.

**Table 12 – EANN Testing Results - displays the testing results for extended out-of-sample data set containing 116 consecutive months spanning from November 1998 to June 2008.**

<b>Performance Measure</b>	<b>Multi</b>	<b>Single</b>
Accuracy (direction)	58.62%	56.03%
Accuracy (magnitude)	56.67%	36.67%
% of accurate contraction predictions	59.26%	52.94%
Number of contraction predictions	27	34
Yearly investment yield (%)	4.56%	0.66%
Cumulative investment return (\$)	\$1552	\$1066
Yearly excess return to the market (%)	1.65%	- 2.24%
Sharpe Ratio	8.27	-7.30

In table 13 the p-values for accuracy over direction and magnitude are displayed, as noted in section 4.6.1, accuracy over magnitude is calculated based on the algorithm predicting

the right market direction at a time when magnitude is above one standard deviation of the mean monthly return.

**Table 13 - P-values for multi-objective optimization - p-values generated from a one-sided t-test for binary distributions.**

Test	P-value
Overall accuracy	0.286868
Accuracy during larger magnitude	0.014418

The larger sample size has produced similar results to those reported in chapter 4, where the MOO approach has generated more correct predictions in times of higher volatility which has resulted in larger cumulative investment returns. With the results obtained from the one-sided t-test (a p-value of 0.014418) we can reject the null hypothesis and therefore accept the alternative. Not considered in the main hypothesis is whether or not modeling magnitude aids in predicting direction for any magnitude. From the data above the superior accuracy over direction obtained by the MOO approach was not statistically significant this was consistent with the results obtained in chapter 4. The sample size in chapter 4 for direction was large enough to produce a relevant statistical test (sample size was greater than 30).

### **6.3 – NLP based models**

The null and alternative hypotheses for this chapter are:

$H_0$  – *The combined models do not produce superior results from all of the individual models (character n-gram, word n-gram, and readability score models) in terms of over-perform precision.*

The alternative hypothesis is:

$H_1$  – *The combined models yield a level of over-perform precision which implies a significant improvement over the individual models.*

In chapter 5, two novel approaches were introduced to aid in the automatic processing of annual reports. Each technique on its own was able to model a portion of the information contained in said reports, however the models which combined the information captured in each of the approaches appeared to be the most robust where they produced the best

results in terms of accuracy and precision on over performing stocks. In addition to the metrics reported in Chapter 5 a statistical analysis was completed which aimed at determining if the improved performance of the combined models are statistically significant from each of the individual models. Table 14 is a legend of the model that was examined and the symbol that represents said model for use in Table 15.

**Table 14 – NLP Based Models Symbol Legend - Legend of each model that was analyzed in chapter 5 and its corresponding symbol.**

<b>Model</b>	<b>Symbol</b>
Character N-gram CNG analysis	c-gram
Word N-gram CNG analysis	w-gram
Readability scores with SVM analysis	read
Combo model of c-gram and read	c-combo
Combo model of w-gram and read	w-combo

Displayed in table 15 are the reported p-values from each of the statistical significance tests over each of the performance metrics (accuracy and precision of over performers).

**Table 15 - P-values for NLP based models – displays the associated p-values for various statistical tests, the (\*) represents results which are statistically significant.**

<b>Accuracy Test</b>	<b>Accuracy p-value</b>	<b>Over-perform precision p-value</b>
c-gram & c-combo	0.002706*	0.000205*
w-gram & w-combo	0.004273*	0.001909*
read & c-combo	0.161446	0.042749*
read & w-combo	0.043345*	0.042749*
c-combo & w-combo	0.267685	0.5

With the results above we can examine the validity of the null hypothesis. The p-values for each of the tests concerning the criteria given in the null hypothesis have values of less than 0.05 which means that the results from the combined models are statistically significant and therefore the null hypothesis can be rejected and the alternative can be accepted. Not considered in the null hypothesis was whether or not the overall accuracy was a significant increase. The tests were for the most part consistent with the over-

perform precision, however the one anomaly was that the character n-gram combo model did not have a significant increase over the model produced from readability scores alone. In the analysis this result is not as important since the emphasis was on choosing correct investments and not as concerned with the opportunity cost of missing out on them. The anomaly is a result of the readability model having a greater interpretation of poor performing stocks than that of those which are going to outperform the market benchmark.

## **Chapter 7**

### **Conclusion**

The main objective of this thesis was to develop and expand the techniques used in financial modeling with the aid of artificial intelligence (AI), as stated in the introduction the results are not intended as evidence for or against the weak form of the EMH but is conducted under the assumption that technical analysis is able to produce sustainable excess returns. Each of the techniques developed in this thesis were tested to be superior to their relevant benchmark, be that another data representation, objective optimization or another algorithm. The results also indicated that the models were able to outperform the market but this comparison is not so easily tested as transaction costs are not considered and depending on the situation can and will negatively impact the reported investment returns. The statistical significance reported in the previous section provides support that the methods developed are indeed preferred to their predecessors. However back testing is only a method to add confidence to the models and does not guarantee future performance.

Each of the machine learning algorithms used in this thesis is from the area of supervised learning. As stated previously there are short comings with using accuracy to decide which models are preferred since the investment returns are what are actually important. Another alternative to the multi-objective approach introduced in chapter 4 is to use reinforcement learning (RL). This area has already been explored with regards to financial modeling and has shown promise [51, 52]. Where in supervised learning we are training an algorithm to make predictions on market behaviour with RL the algorithm evaluates the quality of a decision based on the amount of investment return the decision yielded and is therefore used to judge the quality of the decision given the state the stock market was in.

There are several extension for each approach introduced in this thesis, some of which were identified in their respective sections and will be reiterated here. The optimization of a SVM and GP with the aid of Pseudo Financial Factor Model (PFFM) could be extended to testing with other algorithms such as Bayesian classification methods, decision trees and other optimization algorithms (such as Particle Swarm

Optimization). It would also be interesting to test the adaptation of the PFFM with other types of data such as stock, bond or derivative products. Finally the investment objective itself may have an effect on the data representations abilities and therefore other models that differ in their level of activity such as day trading and long-term investing could be explored.

As mentioned in section 4.7 inaccuracies of the model developed from the Evolutionary Artificial Neural Network (EANN) can partly be attributed to the incomplete and noisy input data. Performance may benefit from additional input factors or latent variables that accurately measure or infer other market forces such as major economic news, currency exchange rates and market sentiment. The MOO approach did not explore a Pareto Front for optimization which could lead to more fit individuals and a more thorough exploration of the solution space, in addition, the objectives could be expanded to include factors such as risk and return.

The character and word n-gram profiles generated in Chapter 5 are done so with the entire annual report. It would be beneficial to only include the portions of the annual reports which contain forward looking statements. This would eliminate sections which are quite uniform between reports, such as the notes to financial statements. By eliminating potential similar portions which are of no use, the CNG distance measure may uncover greater distinction between reports. These improvements would also benefit the calculation of the readability scores in a similar fashion.

## References

- [1] Eugene Fama. Efficient Capital Markets: A Review of Theory and Empirical Work. *Journal of Finance*, XXV, No. 2 (March 1970), pages 226-241.
- [2] Allan Borodin, Ran El-Yaniv and Vincent Gogan. Can We Learn to Beat the Best Stock. *Journal of Artificial Intelligence Research* 21 (2004) 579-594.
- [3] Stefan Zemke. Data Mining for Prediction. Financial Series Case. The Royal Institute of Technology, Sweden (2003). ISBN 91-7283-613-X.
- [4] D. Enke, and S. Thawornwong. The adaptive selection of financial and economic variables for use with artificial neural networks. *Nerocomputing* 56 (2004) 205-232.
- [5] A.W. Lo, A.C. MacKinlay, Stock market prices do not follow random walks: evidence from a simple specification test, *Rev. Financial Stud.* 1 (1988) 41–66.
- [6] Azoff, E.M. (1994). Neural network time series forecasting of financial markets. John Wiley and Sons, Chichester.
- [7] Bachelier, L. (1900a), "Théorie de la spéculation", *Annales Scientifiques de l'École Normale Supérieure* 3 (17): 21–86.
- [8] Fama, Eugene F. (*September/October 1965*). Random Walks In Stock Market Prices. *Financial Analysts Journal* 21 (5): 55–59.
- [9] SEC (2008-09-19). "SEC Halts Short Selling of Financial Stocks to Protect Investors and Markets". <http://www.sec.gov/news/press/2008/2008-211.htm>. Last visited on June 10th, 2009.
- [10] Seth Klarman. (1991) 'Margin of safety' published by Harpercollins, ISBN 0-88730-510-5.
- [11] M. Leung, H. Daouk and A. Chen. Forecasting stock indices: a comparison of classification and level estimation models. *International Journal of Forecasting* 16 (2000) pages 173-190.
- [12] N. O'Connor and M. G. Madden. A Neural Network Approach to Predicting Stock Exchange Movements using External Factors. *Applications and Innovations in Intelligent Systems XIII* (2006). ISBN 978-1-84628-224-9.
- [13] M. Leung, H. Daouk and A. Chen. Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index. *Computers and Operations Research* 30 (2003) 901-923.
- [14] D. Woods and B. Dasgupta. Classifying trend movements in the MSCI U.S.A. capital market index—A comparison of regression, arima and neural network methods. *Computers and Operations Research* 23 (1996) 611-622.
- [15] Kešelj, V., Peng, F., Cercone, N., Thomas, C.: N-gram-based author profiles for authorship attribution. In: *Proceedings of the Conference Pacific Association for Computational Linguistics, PACLING 2003*, Dalhousie University, Halifax, Nova Scotia, Canada, pp. 255–264 (August 2003).



- [16] Fan, Allan and Palaniswami. Stock selection using Support Vector Machines. Neural Networks, 2001. Proceedings IJCNN 01. Vol. 3, pages 1793-1798.
- [17] Huang, Wei, Nakamori, Yoshiteru, and Wang, Shou-Yang. Forecasting stock market movement direction with support vector machine. Computers and Operations Research 32 (2005) 2513-2522.
- [18] Gavrishchaka, Valeriy V., Banerjee. Support vector machine as an efficient framework for stock market volatility forecasting. Computational Management Science, Volume 3, (2006), pages 147-160.
- [19] Iba, H. and Sasaki, T. (1999). Using Genetic programming to Predict Financial Data. Evolutionary Computation, 1999. CE 99. Volume 1, page 251.
- [20] Li, J. and E.P.K. Tsang (1990a). Improving Technical Analysis Prediction: An Application of Genetic Programing. Proceedings of The 12th International Florida AI Research Society Conference. Pages108-112.
- [21] Potvin, J., Soriano, P. and Vallée, M (2004). Generating trading rules on the stock markets with genetic programming. Computers and Operations Research 31, 2004, pages 1033-1047.
- [22] A. Azzini, A. Tettamanzi. A Neural Evolutionary Approach to Financial Modeling. In GECCO 06: Proceedings of the 2006 conference on Genetic and evolutionary computation pages 1605-1612.
- [23] A.M. Mora, P.A. Castillo, J.J. Merelo, E. Alfaro-Cid, A.I. Esparcia-Alcázar, K. Sharman. Discovering Causes of Financial Distress by Combining Evolutionary Algorithms and Artificial Neural Networks. In GECCO 08: Proceedings of the 2008 conference on Genetic and evolutionary computation, pages 1243 – 1251.
- [24] A. Briza and C. Naval Jr. Design of Stock Trading System for Historical Market Data Using Particle Swarm Optimization of Technical Indicators. In GECCO 08: Proceedings of the 2008 conference on Genetic and evolutionary computation, pages 1871 – 1879.
- [25] Falinouss, P.: Stock trend prediction using news articles. Master's thesis, Lulea University of Technology (2007) ISSN 1653-0187.
- [26] Schumaker, R., Chen, H.: Textual analysis of stock market prediction using financial news articles. In: Proc. from the America's Conf. on Inform. Systems (2006).
- [27] Mittermayer, M.: Forecasting intraday stock price trends with text mining techniques. In: Proc. of the 37th Hawaii Int'nal Conf. on System Sciences (2004).

- [28] Kloptchenko, A., Magnusson, C., Back, B., Vanharanta, H., Visa, A.: Mining textual contents of quarterly reports. Technical Report No. 515, TUCS (May 2002) ISBN 952-12-1138-5.
- [29] Kloptchenko, A., Eklund, T., Karlsson, J., Back, B., Vanharanta, H., Visa, A.: Combined data and text mining techniques for analysing financial reports. *Intelligent Systems in Accounting, Finance and Management* 12, 29–41 (2004).
- [30] Subramanian, R., Insley, R., Blackwell, R.: Performance and readability: A comparison of annual reports of profitable and unprofitable corporations. *The Journal of Business Communication* (1993).
- [31] Li, F.: Annual report readability, current earnings, and earnings persistence. *Journal of Accounting and Economics* (2008).
- [32] D. Enke, and S. Thawornwong. The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with applications* 29 (2005) pages 927-940.
- [33] Ian H. Witten and Eibe Frank (2005). *Data Mining: Practical machine learning tools and techniques*. 2nd Edition, Morgan Kaufmann, San Francisco, 2005.
- [34] Bill Punch, Douglas Zongker (1998) Michigan State University GARAGe <http://garage.cse.msu.edu/>
- [35] Koza, John (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press. ISBN 0-262-11170-5
- [36] Holland, John H (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor
- [37] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [38] R.E. Fan, P.H. Chen, and C.J. Lin. Working set selection using second order information for training SVM. *Journal of Machine Learning Research* 6, 1889-1918, 2005.
- [39] A. Abraham and L. Jain. Evolutionary Multiobjective Optimization. In Ajith Abraham, Lakhmi Jain, and Robert Goldberg, editors. *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 1-6. Springer, USA, 2005.
- [40] K.O. Stanley and R. Mikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2), 2002.

- [41] K. Stanley, B. D. Bryant and R. Mikkulainen. Real-time Neuroevolution in the NERO Video Game. *IEEE Transactions on Evolutionary Computation*, volume 9, number 6, pages 653-668, December 2005.
- [42] N. Kohl, K. Stanley, R. Mikkulainen. Evolving a Real-World Vehicle Warning System. In *GECCO 06: Proceedings of the 2006 conference on Genetic and evolutionary computation*, pages 1681-1688.
- [43] Securities and Exchange Commission (SEC). Rules of Practice and Rules on Fair Fund and Disgorgement Plans. <http://sec.gov/about/rulesprac2006.pdf>. Last access June 15, 2009.
- [44] Yahoo! Inc.: Yahoo! finance, <http://ca.finance.yahoo.com/> (last access 2008).
- [45] Kešelj, V.: Text:Ngrams Perl module for flexible ngram analysis (2003–2009), <http://www.cs.dal.ca/~Evlado/srcperl/Ngrams/Ngrams.html> Ver, 2.002. Available at CPAN.
- [46] Ryan, K.: Lingua:EN: Fathom Perl module for measuring readability of English text. Available at CPAN (2007).
- [47] CPAN Community: CPAN—Comprehensive Perl Archive Network (1995–2009), <http://cpan.org>.
- [48] Fast, G.: Lingua:EN: Syllable Perl module for estimating syllable count in words. Available at CPAN (1999), <http://search.cpan.org/perldoc?Lingua:EN:Syllable>.
- [49] Fan, R., Chen, P., Lin, C.: Working set selection using second order information for training SVM. *Journal of Machine Learning Research* 6, 1889–1918 (2005).
- [50] Kim, K.: Financial time series forecasting using support vector machines. *Neurocomputing* 55, 307–319 (2003)
- [51] J. W. Lee and J. Park. A Multiagent Approach to Q-Learning for Daily Stock Trading. *IEEE Transactions on Systems, Man and Cybernetics- Part A: Systems and Humans*, VOL. 37, NO. 6, November 2007.
- [52] J. W. Lee. Stock Price Prediction Using Reinforcement Learning. *Proceedings from ISIE 2001, IEEE International Symposium on Industrial Electronics*, VOL. 1, pages 690-695.