Modeling Hedge Fund Performance Using Neural Network Models

By

Marinos Tryphonas

Supervised by Dr. Joseph C. Paradi

A thesis submitted in conformity with the requirements for the degree of Master of Applied Science

The Centre for Management of Technology and Entrepreneurship

Graduate Department of Chemical Engineering and Applied Chemistry

University of Toronto

Copyright 2012
ABSTRACT

Hedge fund performance is modeled from publically available data using feed-forward neural networks trained using a resilient backpropagation algorithm. The neural network’s performance is then compared with linear regression models. Additionally, a stepwise factor regression approach is introduced to reduce the number of inputs supplied to the models in order to increase precision.

Three main conclusions are drawn: (1) neural networks effectively model hedge fund returns, illustrating the strong non-linear relationships between the economic risk factors and hedge fund performance, (2) while the group of 25 risk factors we draw variables from are used to explain hedge fund performance, the best model performance is achieved using different subsets of the 25 risk factors, and, (3) out-of-sample model performance degrades across the time during the recent (and still on-going) financial crisis compared to less volatile time periods, indicating the models’ inability to predict severely volatile economic scenarios such as economic crises.
ACKNOWLEDGMENTS

I would like to thank my family, my supervisor, and my friends for their support and advice throughout the development of this thesis.

In particular, Dr. Joseph Paradi has been an exceptional supervisor, mentor, and friend to me. Dr. Paradi’s guidance and support throughout my graduate studies has allowed me to attain my academic goals, professional goals, and personal goals, and for that I cannot thank him enough.
### TABLE OF CONTENTS

ABSTRACT ................................................................................................................................. ii  
Acknowledgments ....................................................................................................................... iii  
List of Tables .............................................................................................................................. vi  
List of Figures ............................................................................................................................. vii  
Glossary of Terms ....................................................................................................................... viii  
Executive Summary ..................................................................................................................... x  
1 Introduction ............................................................................................................................. 1  
1.1 Thesis Outline ...................................................................................................................... 3  
2 Neural Networks ...................................................................................................................... 5  
2.1 Artificial Neural Networks ................................................................................................. 5  
2.2 Training Neural Networks – Resilient Backpropagation .................................................... 9  
2.3 Neural Network Performance Evaluation ......................................................................... 12  
2.4 Neural Network Over-fitting ............................................................................................ 13  
3 Hedge Funds .......................................................................................................................... 15  
3.1 Hedge Fund Strategies ...................................................................................................... 16  
4 Literature Review .................................................................................................................. 18  
4.1 Neural Networks in Finance ............................................................................................. 18  
4.1.1 Mutual Fund Performance Prediction Using Artificial Neural Networks ................ 18  
4.1.2 Mutual Fund Net Asset Value Forecasting Using Artificial Neural Networks ........ 19  
4.1.3 Stock Price Prediction Using Neural Networks ........................................................ 19  
4.1.4 Neural Networks and Hedge Funds ............................................................................ 20  
4.2 Hedge Funds .................................................................................................................... 20  
4.3 Risk Factors ....................................................................................................................... 21  
5 Data ......................................................................................................................................... 24  
5.1 Hedge Fund Data ............................................................................................................... 24  
5.1.1 Hedge Fund Database Biases ...................................................................................... 24  
5.2 Time Periods Studied ........................................................................................................ 26  
5.3 Selected Risk Factors ....................................................................................................... 27  
5.3.1 Domestic Equity Factors ............................................................................................. 27  
5.3.2 International Equity Factors ....................................................................................... 27  
5.3.3 Domestic Fixed Income Factors .................................................................................. 28  
5.3.4 International Fixed Income Factors ............................................................................ 28  
5.3.5 Commodities Factors ................................................................................................. 28  
5.3.6 Non-Linear Factors ..................................................................................................... 29  
5.4 Data Normalization ........................................................................................................... 30  
6 Experimental Approach ........................................................................................................ 32  
6.1 Linear Regression Models ............................................................................................... 32  
6.2 Neural Network Training .................................................................................................. 33  
6.3 Stepwise Regression .......................................................................................................... 35  
6.4 Performance Evaluation .................................................................................................... 36  
7 Results and Discussion ........................................................................................................... 39  
7.1 Stepwise Regression Results ............................................................................................ 39  
7.2 Time Period #1 (Dec 1998-May 2006) ......................................................................... 40  
7.3 Time Period #2 (December 1998-May November 2010) .............................................. 48  
7.4 Out-Of-Sample Modeling Performance Comparison Between Study Periods .............. 55
8 Conclusions........................................................................................................................... 59
8.1 Future Work ................................................................................................................... 60
References..................................................................................................................................... 62
LIST OF TABLES

Table 1: Notable differences between traditional investment programs and hedge funds [3]. ...... 2

Table 2: A summary of the 25 selected risk factors used with their corresponding sources. ....... 30

Table 3: A table demonstrating the input and target normalization process. ......................... 31

Table 4: Hypothetical MSE values for three model types for five different funds. ..................... 38

Table 5: A summary of the stepwise regression backwards elimination results. ..................... 39

Table 6: Performance statistics for regression and ANN models for Time Period #1. ............... 42

Table 7: Funds better modeled by a neural network or a linear regression model for Time Period #1. ........................................................................................................................................................................ 43

Table 8: Performance statistics for regression and ANN models for Time Period #2. ............. 49

Table 9: Neural network and linear regression models' performance for Time Period #2. ....... 50
LIST OF FIGURES

Figure 1: Layers of neurons in neural networks. The weighted connections between the neurons are shown as straight lines connecting the neurons together. Figure reprinted from Dreyfus 2005 [4]. ............................................................................................................................... 7

Figure 2: A figure depicting a three layer neural network structure. Figure reprinted from MATLAB 2011b neural network toolbox documentation [5]. ................................................................. 9

Figure 3: An example of a polynomial over fitting to the data ........................................... 14

Figure 4: The two study periods and their corresponding training and validation sections. ....... 26

Figure 5: A comparison of the number of risk factors contained in specific fund models for both time periods .................................................................................................................. 40

Figure 6: Mean-Squared and standard deviation comparisons across models (Time Period #1) 45

Figure 7: Funds better modeled by a neural network or a linear regression model for Time Period #1 ........................................................................................................................................ 46

Figure 8: Stepwise factors selection performance comparison for Time Period #1 ............... 47

Figure 9: Mean-squared error and standard deviation comparisons across models for Time Period #2 .............................................................................................................................................. 52

Figure 10: Funds better modeled by a neural network or a linear regression model for Time Period #2 ........................................................................................................................................ 53

Figure 11: Stepwise factor selection performance for Time Period #2 ................................ 54

Figure 12: Comparison of the median and mean MSE’s for both time periods ....................... 56

Figure 13: Comparison of the median and mean standard deviations for both time periods ..... 57
GLOSSARY OF TERMS

a. Abbreviation List

ANN  Artificial neural network
BPN  Backpropagation network
Citi USBIG  Citigroup United States Broad Investment Grade Index
Citi WBIG  Citigroup World Broad Investment Grade Index
Citi WGBI  Citigroup World Government Bond Index
DAX  DuetscherAktienIndeX
FFN  Feed forward network
HML  High minus low factor
IMF  International Monetary Fund
LTCM  Long-Term Capital Management
MOM  Momentum factor
MSCI EAFE Index  Morgan Stanley Capital International Europe, Australasia, and Far East
MSCI EM  Morgan Stanley Capital International Emerging Markets Index
MSE  Mean squared error
NAREIT  National Association of Real Estate Investment Trusts
NAV  Net asset value
NYSE  New York Stock Exchange
PTFS  Primitive trend following strategies
RPROP  Resilient backpropagation
SMB  Small minus big factor
STD  Standard deviation
### b. Terminology List

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artificial Neural Network</td>
<td>A black box modeling tool that is based on the biological brain’s impressive ability to learn. An artificial neural network is especially effective at modeling nonlinear relationships.</td>
</tr>
<tr>
<td>Input</td>
<td>An input in this project is one of the 25 economic risk factors that are chosen to be input data for the regression and neural networks.</td>
</tr>
<tr>
<td>Mean Squared Error</td>
<td>The average of the squared difference between the output and target values for any constructed model using out of sample validation data.</td>
</tr>
<tr>
<td>Neuron</td>
<td>A modeling unit within a neural network. A neuron receives a weighted input of values that have a threshold value subtracted from them. The neuron takes this value and transforms it using a transfer function, which results in the activation of the neuron. The activation of the neuron is then output across weighted connections as inputs for further neurons.</td>
</tr>
<tr>
<td>Output</td>
<td>An output is the value produced by a constructed model when fed inputs. The output of the model is compared to the target value (which represents what actually happened) in order to gauge model predictive performance.</td>
</tr>
<tr>
<td>Performance</td>
<td>The predictive ability of a constructed model, typically measured by the mean-squared error (MSE).</td>
</tr>
<tr>
<td>Resilient Backpropagation</td>
<td>A training algorithm used to adjust artificial neural network connection weights in order to improve predictive performance.</td>
</tr>
<tr>
<td>Training Data</td>
<td>The set of input and target data used to construct a model.</td>
</tr>
<tr>
<td>Target</td>
<td>The data that represents what actually happened, e.g. actual monthly hedge fund return performance.</td>
</tr>
<tr>
<td>Validation Data</td>
<td>The set of input and target data used to validate and gauge model performance.</td>
</tr>
</tbody>
</table>
EXECUTIVE SUMMARY

Modeling hedge fund returns has been an exciting and difficult subject ever since hedge funds have seen unprecedented growth in the past two decades. Due to their complex, unregulated, and often undisclosed trading strategies, along with a lack of historical performance data until just recently, modeling hedge fund performance has been difficult to accomplish.

This thesis proposes a new way to model hedge fund monthly performance using a neural network modeling approach. Specifically, a three-layer neural network with 1-8 hidden layer neurons was constructed using a resilient backpropagation training algorithm.

As well, the thesis explores a model’s predictive ability when modeling volatile market events, such as the recent financial crisis. Finally, the thesis introduces the use of a stepwise regression backwards elimination factor selection (SW) approach in order to reduce the number of inputs used in the performance models, as the use of too many factors may impair model performance.

Hedge fund performance data was studied over two time periods: the first from December 1998 to May 2006, the second from December 1998 to November 2010, which includes the recent financial crisis. Monthly return data was collected during these time periods for 293 unique hedge funds. Additionally, 25 risk factors were selected as inputs based on their demonstrated explanatory performance on hedge fund returns as outlined in the literature.

The new neural network modeling approach delivers better out-of-sample performance compared to regression models using identical risk factors as inputs in both time periods. For example, out of 293 funds studied, a neural network with one hidden layer unit and SW risk factor selection achieved better performance with 216 and 213 of the 293 total funds for the first
and second study period, respectively. However, this additional performance comes at a cost of higher variance, as neural networks models consistently had higher standard deviations compared to linear regression models.

Furthermore, stepwise regression backwards elimination risk factor selection consistently improved performance for all model types across both time periods. For example, for the first time period, 235 out of the 293 total funds were better modeled by a neural network with one hidden layer unit and SW risk factor selection compared to a neural network with one hidden layer unit without SW factor selection. This result indicates that not all risk factors have explanatory power for every fund. Hence, including all of them would only increase the in-sample performance of the models by using the extra regression or neuron parameters to fit to the data points, as opposed to the trend represented by the data.

Finally, out-of-sample model performance is significantly worse during the second time period compared to the first time period with comparable models. The second time period covers the recent financial crisis as an out-of-sample set to test the models. However, compared to the first time period’s out-of-sample testing data, the second time period’s comparable models have greater mean squared errors and standard deviations in the range of 150%-1600%. This result indicates that the volatile financial crisis is difficult to model with the chosen risk factors and hedge fund return data. As a result, it is important to understand that while a model may achieve excellent performance over a certain time frame, there is no guarantee that this performance remains during significant economic shifts.
1 INTRODUCTION

The hedge fund industry has seen tremendous growth from the early 90s up to now, with tens of thousands of active funds managing up to $2 trillion in assets [1]. A hedge fund is an investment program that is controlled by a skilled investment manager who seeks positive returns while protecting investors’ principal from losses [2]. The primary distinguishing characteristics of a hedge fund are threefold. First, compared to other traditional investment strategies, hedge funds are highly unregulated, allowing them to execute more complex investment strategies that can include leverage, derivatives trading, and shorting. Secondly, a hedge fund is distinguished by its investors. Hedge fund investors must be accredited, which essentially means they must have a high net worth or a high net income. In Canada, an accredited investor must have a net worth of at least $1 million or an annual net income that is greater than $200,000. Finally, due to the unregulated nature of hedge funds, the holdings and specific investment strategies of the funds are not required to be publically disclosed. As a result, hedge funds are less transparent compared to traditional investment programs [2].

Due to their lack of regulation, hedge fund returns are not as dependent on market direction, whereas other investment program returns are mainly driven by market behavior of the primary type of assets that make up that particular investment program.

Table 1 is an excerpt from the Alternative Investment Management Association’s (AIMA) 2004 Canada Hedge Fund Primer which outlines the main differences between traditional investment programs and hedge funds [3].
Table 1: Notable differences between traditional investment programs and hedge funds [3].

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Traditional Investing</th>
<th>Hedge Fund Investing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Return Objective</td>
<td>Relative returns</td>
<td>Absolute returns</td>
</tr>
<tr>
<td>2. Benchmark</td>
<td>Constrained by benchmark index</td>
<td>Unconstrained by benchmark index</td>
</tr>
<tr>
<td>3. Investment Strategies</td>
<td>Limited investment strategies</td>
<td>Flexible investment strategies</td>
</tr>
<tr>
<td></td>
<td>Take long-only positions</td>
<td>Take long and short positions</td>
</tr>
<tr>
<td></td>
<td>Do not use leverage</td>
<td>May use leverage</td>
</tr>
<tr>
<td>4. Market Correlation</td>
<td>High correlation to traditional asset classes</td>
<td>Generally, low correlation to traditional asset classes</td>
</tr>
<tr>
<td>5. Performance</td>
<td>Dependent on market direction</td>
<td>Often independent of market direction</td>
</tr>
<tr>
<td>6. Fees</td>
<td>Tied to assets under management, not to</td>
<td>Tied primarily to performance</td>
</tr>
<tr>
<td></td>
<td>performance</td>
<td></td>
</tr>
<tr>
<td>7. Manager’s Investment</td>
<td>Manager may or may not co-invest alongside</td>
<td>Manager generally co-invests alongside</td>
</tr>
<tr>
<td></td>
<td>investors</td>
<td>investors</td>
</tr>
<tr>
<td>8. Liquidity</td>
<td>Good liquidity</td>
<td>Liquidity restrictions and initial lock-up periods</td>
</tr>
<tr>
<td>9. Investment Size</td>
<td>Small minimum investment size (e.g., $1,000 minimum)</td>
<td>Usually large minimum investment size (e.g., $25,000 minimum; depends on prospectus exemption)</td>
</tr>
<tr>
<td>10. Structure and</td>
<td>Set up as a trust or investment company</td>
<td>Set up as a private investment, limited</td>
</tr>
<tr>
<td>Documentation</td>
<td>Often sold by prospectus</td>
<td>partnership or a trust.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Usually sold by offering memorandum</td>
</tr>
<tr>
<td>11. Regulation</td>
<td>Highly regulated; restricted use of short</td>
<td>Less regulated; no restrictions on</td>
</tr>
<tr>
<td></td>
<td>selling and leverage. High disclosure and</td>
<td>strategies. Less mandated disclosure, and</td>
</tr>
<tr>
<td></td>
<td>transparency. Can market fund publicly</td>
<td>limited or no position level and risk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>exposure transparency. Market</td>
</tr>
<tr>
<td></td>
<td></td>
<td>restrictions apply.</td>
</tr>
</tbody>
</table>

Hedge funds are able to incorporate multiple complex investment strategies and techniques because they are less regulated compared to more traditional investment programs, such as mutual funds. As a result, there is a greater opportunity for manager skill to play a role in increasing a hedge fund’s return. While loosened regulations on hedge fund positions can allow for better absolute returns for investors, spectacular losses sometimes also occur. Long-Term Capital Management (LTCM) was a hedge fund managed by two Nobel Memorial Prize winning investment managers that averaged annualized returns of over 40% for its initial years of existence. However, due to highly leveraged positions in Russian bonds that were not profitable during the Russian financial crisis, the fund lost most of its approximately $4.8 billion in capital over the course of a few months in 1998. While the LTCM scenario is not common, it is worth noting that hedge fund performance can vary significantly compared to other investment programs. As a result, a significant segment of academic literature focuses on which economic
risk factors best explain hedge fund performance and to what extent, as well as what effect the manager has on hedge fund returns.

Only recently has there been a sufficient amount of hedge fund historical performance data collected in order to conduct studies on various fund attributes such as performance modeling, return characteristics, and exposure to risk factors to name a few. As a result, the vast majority of hedge fund academic literature has been published within the past ten years.

An artificial neural network (ANN) is a powerful modeling tool that was developed based on the impressive learning capabilities of biological neural networks. ANN’s are particularly popular because of their ability to model complex non-linear input-target relationships without any model specification from the user. The appeal for a neural network model in finance applications is obvious, as the exact relationships between risk factors and performance data are not well understood, even though studies demonstrate a clear relationship between the inputs and outputs. In the context of this thesis, neural networks are used to model hedge fund monthly returns.

1.1 THESIS OUTLINE

The rest of this document is organized as follows: Chapter 2, Neural Networks, discusses neural networks, their underlying theory, and their applications in enough detail to be appropriate to our work. Chapter 3, Hedge Funds, gives an overview on hedge funds and what differentiates them from traditional investment programs. Chapter 4, Literature Review, summarizes the relevant literature in hedge fund research and financial based neural network research domains. Chapter 5, Data, details the time periods we studied, the data we selected, the normalization process, and the sources for the data we obtained. Chapter 6, Methodology, outlines the construction of our linear regression, neural network models, and stepwise regression, as well as
discusses how we evaluated the performance of our models. Chapter 7, Results and Discussion, presents the performance results for the linear regression, stepwise regression, and neural network models, offers explanations for the results, as well as present further ways to advance the study results. Finally, Chapter 8, Conclusions and Future Work, summarizes the goals and findings from this thesis and offer possible extensions of this work for the future.

We modeled multiple individual hedge fund monthly returns using linear regression and neural networks, drawing from twenty-five (25) different economic risk factors as inputs for the models. Risk factor selection for each fund is accomplished using stepwise regression to minimize variance and error, increasing the degrees of freedom and therefore the fit of the models. We examined model performance over two time periods, the first from December 1998 to May 2006, and the second from December 1998 to November 2010. The first study period examines model performances over a less volatile time frame, whereas the second examines model performances during the recent global financial crisis.
2 NEURAL NETWORKS

The human brain consists of an estimated 10 billion neurons and 60 trillion synapses (connections) between them. Within the brain are biological neural networks, which are groups of interconnected neurons that work with other networks to perform a specific physiological function [4].

A neuron is a highly differentiated cell that propagates electrochemical signals received from different stimuli in the human body. The neuron is comprised of dendrites, a cell body, and an axon. Essentially, the dendrites receive input electrochemical signals from other neurons and deliver the weighted sum of these signals to the cell body. The cell body receives the signal and, if the signal is strong enough, the cell body will activate and transform the signal. The transformed signal is sent to the axon, which delivers the transformed signal to the dendrites of other neurons via a synapse, which connects the neurons together [4].

It has been hypothesized that the synapses play a crucial role in learning. The synaptic connections between neurons carry the electrochemical signals with varying efficiencies, and these efficiencies can be adjusted over time. It is believed that learning is primarily accomplished through the adjustment of these synaptic efficiencies in a biological neural network. While there are more complexities to human learning, mathematical models based on these simple principles have been able to remarkably model complex input-output relationships in many unique research fields. Artificial neural networks have been inspired from the impressive results of those initial models [4].

2.1 ARTIFICIAL NEURAL NETWORKS
An artificial neural network (ANN) is a very powerful modeling tool that is based on a crude approximation of biological neural networks. The ANN is popular because of its ability to capture nonlinear relationships between input-target data without requiring significant user input describing the particular nonlinear interactions between the input and output data, unlike more advanced non-linear modeling methods. The network requires both existing input data and target data in order to learn the relationships between them, which is a process known as “training”. The input data is the selected inputs that the user chooses to model the target data, which is the data that the user is trying to predict. When training a network, a specific training algorithm adjusts certain parameters within the network (which is discussed later) to increase the predictive power of the network. The network then uses the input data to guess an output, which is always referred to as the “output” of the network. The output of the network is then compared to the supplied target data in order to see how close the prediction was to what actually happened [4].

The appeal for a neural network model in finance applications is obvious, as the exact relationships between risk factors and performance data are not well understood, even though studies demonstrate a clear relationship between the inputs and outputs. As well, ANNs are able to learn by example. Through the use of training algorithms, the network can optimize the adjustment of its parameters in order to achieve the best possible fit with the data supplied to the network [4].

An ANN is made up of artificial neurons. An artificial neuron receives a number of inputs, which are either from supplied input data or from the outputs of other neurons in the ANN. The connections between the inputs and the neuron have a specific weight associated with them, which are analogous to the efficiency of the synaptic connections in biological neural networks. The weighted sums of the inputs are sent to the neuron, at which point a threshold value is
subtracted to determine the activation of the neuron. The activation of the neuron is passed through a transfer function in order to calculate the output of the neuron [4].

When using neural networks, the number of neurons and the way they are all interconnected is determined by the user. Typically, a feedforward structure is used as it has the most success in common modeling problems. A typical feedforward network has three different groups of neurons which are connected to one another in a fully connected fashion. A fully connected network means that each neuron in a specific layer has an individual weighted connection with every single neuron in the layers directly before and after it. For example, consider a network with four neurons in the input layer, three neurons in the hidden layer, and three neurons in the output layer as shown in *Figure 1*. The second neuron in the second layer has seven connections: four connections with the four neurons in the first (input) layer, and three connections with the three neurons in the third (output) layer as shown in *Figure 1* [4].

![Figure 1: Layers of neurons in neural networks. The weighted connections between the neurons are shown as straight lines connecting the neurons together. Figure reprinted from Dreyfus 2005][4].](image-url)
The first group of neurons, known as the input layer, receives the input data that the user supplies. The input layer neurons pass their input data across their weighted connections to the next layer of neurons. The next group of neurons, known as the hidden layer, receives the input data from the input layer neurons across its weighted connections. A threshold value is subtracted from the weighted sum from each neuron in order to calculate what is known as the activation of the neuron. The activations are then passed through the neurons’ individual transfer functions to calculate the outputs of all of the hidden layer neurons. The outputs of these hidden layer neurons are then passed across a second set of weighted connections to the third group of neurons, known as the output later. The output layer neurons receive the weighted sum of the signals sent from the hidden layer neurons, and like the hidden layer neurons, the threshold values for each neuron are subtracted from the weighted sum to calculate the activation of the output neurons. The activations of the output layer neurons are then passed through another transfer function to determine the output of the output layer neurons [4].

The values of the output layer neurons are what the network has predicted should happen based on the inputs supplied. The actual output value, or the target value, is what the output actually was. Therefore, the error of the network is the difference between the output value of the network and the target value associated with the specific inputs supplied to the network [4].

The flow of data through these three types of layers in a network is further shown in Figure 2 with more detail from the MathWorks Neural Network Toolbox Documentation Website[5].
As shown in Figure 2, the network has k inputs which are fully connected to the three neurons in the input layer. The Σ-boxes represent the function that takes the weighted sum of the inputs and then subtracts the threshold value, denoted by “b” in the figure. The value produced from the summing function is then passed through to the transfer function, as represented by the f-boxes in the figure. The transfer function transforms the value produced by the summing function, producing the activation of the neuron, represented by \( a \). The activation of the neuron is then passed through weighted connections, represented by \( w \), which are connected to neurons in the next layer. The values of the weighted connections, \( w \), are adjusted during the training process [4,5].

2.2 TRAINING NEURAL NETWORKS – RESILIENT BACKPROPAGATION

In order to use neural networks to model complex data sets, the network must be trained to minimize the difference in the output of the network and the target value associated with the
input data. Additionally, the data input to the network needs to be normalized and presented to the network.

Neural network backpropagation training is described explained by the following pseudocode:

1. Initialize network by randomly assigning weight values
2. Input the training vector into the network
3. Hidden nodes calculate their outputs
4. Output nodes calculate their outputs based on hidden node outputs
5. Determine the network error by calculating the difference between the output nodes and the target output values
6. Calculate the weight change for the weighted connections between neurons nodes based on the network error, specified learning parameters, and calculated gradients.
7. Repeat steps 2-6 until a training goal is met.

There are many ways to train neural networks, and in this project we use resilient backpropagation (RPROP) to train our networks. Backpropagation uses an iterative approach to train the neural network, going through a number of “epochs”. In an epoch, the network is adjusted based on the training algorithm, and then the output is compared to the target, determining the error of the network [6].

Batch backpropagation repeatedly uses the chain rule to determine the importance of each specific weight in the entire neural network. This analysis is done by examining an error function, $E$, in equation 1:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial s_i} \frac{\partial s_i}{\partial net_i} \frac{\partial net_i}{\partial w_{ij}}$$  \tag{1}

Where $j$ and $i$ represent the identification number for a neuron, $w_{ij}$ is the weight from neuron $j$ to neuron $i$, $s_i$ is the output, and $net_i$ is the weighted sum of the inputs of neuron $i$. When the partial derivative is determined, the error function is decreased by using a gradient descent:
\[ w_{ij}(t + 1) = w_{ij}(t) - \epsilon \frac{\partial E}{\partial w_{ij}}(t) \]  \hspace{1cm} (2)

Where \( t \) is the iteration number and \( \epsilon \) is the learning rate, which multiplies the derivative, thus dramatically affecting the time of convergence for the neural network. The learning rate is set by the user: a large rate can cause the weight to oscillate and never converge, whereas a small rate results in too many iterations to reach convergence.

In resilient back propagation, the magnitude of the partial derivative is not used, only the sign is as shown in equations 3 and 4:

\[ w_{ij}(t + 1) = w_{ij}(t) - \Delta w_{ij}(t) \]  \hspace{1cm} (3)

\[ \Delta w_{ij}(t) = -sign\left(\frac{\partial E}{\partial w_{ij}}(t)\right)\Delta_{ij}(t) \]  \hspace{1cm} (4)

The sign is used with an update function (\( \Delta_{ij} \)) to adjust the weight towards a converging value. If in subsequent iterations the derivative has the same sign, then the update function is increased to accelerate the weight to reach a minimum value. Similarly, if in subsequent iterations the derivative has opposite signs, then that indicates that the weight has passed a minimum value. As a result, the update value is decreased to a lower the rate of change in order to ensure that the weight does not skip its convergence conditions. If necessary, update value limits can be set to ensure they do not become too large, causing oscillations, or too small, requiring too many iterations for convergence [6].

The network is finished training after a certain number of epochs are met, or if the error function falls within a certain range, or if the error stops improving.
The data that the network uses for training and validation is presented to the network in a batch form, meaning that the ANN evaluates the output of the network for each instance, compares it to the target value, calculate the error, and only adjusts the weights of the network after all error instances have been evaluated. For example, if the network is modeling monthly financial data for 2004, the model takes the corresponding chosen input values for January 2004 and passes them through the network to generate an output. The output of that network is compared to what the actual return was (the target) to calculate the error for January. The network does the same process for February, March, and so on until all data instances have been evaluated. Once all data instances have been evaluated, the network adjusts its weights based on the utilized training algorithm. After the weights have been adjusted, the process reevaluates the output for the network for all instances and computes the corresponding error. The process continues until a specific training goal has been met.

After a network has converged, it is important to retrain the network using different sets of data to verify that the network has converged to the correct point and that the error is as small as possible. In many cases, networks can converge to local minima and it is not possible to determine if the network has converged to the local or global minimum point without retraining the network. Therefore, retraining is required to see if the network converged to the correct point[7]. However, there is not enough hedge fund return data available to retrain networks with different sets of data. As a result, a small validation data set is used to approximate whether the network has converged to a local minima.

2.3 Neural Network Performance Evaluation

In order to evaluate a neural network’s ability to model the input-output relationship in the presented data, an appropriate performance statistic must be used. Typically, the mean-
squared error (MSE) of the network is used, which is simply the average of all squared error values evaluated by the neural network [4]. Therefore, the performance of the network is negatively proportional to the MSE of the network. In other words, as the MSE is increasing, the performance is decreasing, and vice versa. When the network is training, the MSE is calculated for each epoch for the training set and the validation set. Once the MSE starts to increase on the validation set, the network stops training [4]. The MSE is calculated using equation 5:

$$MSE = \frac{\sum_{i=1}^{n}(Output_i-Target_i)^2}{n}$$  \hspace{1cm} (5)

2.4 NEURAL NETWORK OVER-FITTING

When training a neural network, it is important to set an appropriate number of layers and neurons per layer. If there are too many layers and units, then the network “overfits” to the data, similar to how a polynomial function with many terms can fit a noisy set of data perfectly, instead of following the trend of the data as illustrated in Figure 3[7]. A typical rule of thumb is to ensure that the network has twice as many training instances compared to the number of connections in the network. For example, a network with five inputs, three hidden layer neurons, and two output neurons has 21 connections (5x3 + 3x2 = 21) and in order to ensure that overtraining is minimized, at least 42 training instances should be supplied to the network[8]. Keeping this rule of thumb in mind, additional neuron units in a neural network do not always correspond to better performance, which can be counter-intuitive when initially determining network structure.

Furthermore, if stopping goals for network training are not set, then the network trains indefinitely and continually improves training performance. However, this actually decreases the validation performance. Put another way, even though training set performance continues to
improve indefinitely, it is achieved by adjusting the connection weights to fit more to the individual data points as opposed to the relationship described by the data. As a result, validation sets have very poor performance unless there is a reasonable stopping goal [4].

Figure 3: An example of a polynomial over fitting to the data.

Figure 3 shows the detrimental effect of including too many parameters in a polynomial fit for a small data set, as the polynomial model fits the hypothetical efficiency-dollars spent dataset perfectly but does not fit the trend demonstrated by the data. As a result, the polynomial model fits out-of-sample data very poorly. Clearly, a simple linear model achieves better out-of-sample performance, as it better fits the trend that the supplied data illustrates.
3 Hedge Funds

A hedge fund is an investment program that is controlled by a skilled investment manager who seeks positive returns while protecting investors’ principal from losses[2]. Unlike other traditional investment strategies, hedge fund returns are not as dependent on market direction, whereas other investment program returns are mainly driven by market behavior of the primary type of assets that make up that particular investment program.

A hedge fund manager differs from traditional investment program managers in two significant ways. The first is that a hedge fund manager defines investment risk as the actual potential loss of investment capital, whereas traditional managers tend to measure risk as the potential difference of the fund’s monthly performance compared to a benchmark index, which is known as tracking error. Secondly, hedge fund managers tend to chase absolute returns which are independent of market directions, whereas traditional managers strive to deliver returns that are greater than a certain market benchmark. Based on this, it is clear that hedge fund returns are strongly tied to the manager’s skill, whereas traditional program returns are often driven primarily by market activity[3]. As well, hedge fund managers are typically aligned with their investors in that the managers tend to invest a large proportion of their own personal wealth into the fund.

Hedge fund managers tend to use a range of investment strategies, such as leverage and derivatives, in order to maximize and/or minimize the potential return and risk related to certain positions the manager has taken with the fund. Additionally, hedge fund managers earn commissions based on the returns of the fund, but only if a certain return is met, which is known as a water mark. If the manager earns a return below the water mark, then they are not awarded
any commission. Typically, the water marks are high and not easily attainable, motivating the hedge fund manager to avoid low risk and low return positions that are generally taken by traditional investment programs [3].

Some hedge funds typically have lock-up periods where investors are unable to redeem their investment from anywhere from 3 months to 1 year, or even potentially longer in certain cases. Some funds offer early redemption at a penalty [3].

Hedge funds are not transparent as managers tend to not disclose their positions, holdings, or risk exposure. Investors and the public are typically unable to gain access to this information, and only performance and net-asset value information tends to be available through third-party hedge fund commercial databases [3].

Finally, hedge funds require higher minimum investments, starting at approximately $100,000. Additionally, investors must be accredited in order to invest into a hedge fund. In Canada, an individual investor can be accredited if they have financial assets exceeding $1,000,000, or if they have net income before taxes of $200,000. An entity (e.g. pension fund or a corporation) can also be an accredited investor, but in that case the entity must have assets of at least $5 million [3].

3.1 Hedge Fund Strategies

Hedge fund managers use a multitude of strategies to earn returns, but there is no particular standard to classify hedge funds based on their strategy. Regardless, the AIMA hedge fund primer divides hedge funds into three broad investment strategies: (1) relative value, (2) event driven, and, (3) opportunistic [3].
Based on the AIMA definition, a relative value hedge fund manager chases returns by taking advantage of mispriced assets or securities. The manager attempts to purchase the asset or security at the perceived incorrect price and then resell it at the actual market price in order to earn a return. Typically, hedge funds will borrow money in order to maximize the return from these types of transactions, which is known as leverage [3].

An event-driven fund manager attempts to capture price movements from pending corporate events, such as restructuring, bankruptcies, and mergers [3].

An opportunistic hedge fund manager generally has an evolving investment strategy, taking advantage of current investment opportunities and market conditions. As a result, an opportunistic hedge fund tends to have greater exposure to market movements [3].
4 LITERATURE REVIEW

The literature review section first discusses neural network applications in stock and mutual fund performance prediction. Hedge fund performance modeling and related literature is then discussed. Finally, risk factors and data biases are reviewed.

4.1 NEURAL NETWORKS IN FINANCE

Neural networks have successfully been used in mutual fund and stock prediction, lending to the hypothesis that neural networks can also capture the relationship between hedge fund returns and selected risk factors.

4.1.1 MUTUAL FUND PERFORMANCE PREDICTION USING ARTIFICIAL NEURAL NETWORKS

A study published in 1997 evaluated an ANN’s ability to predict mutual fund performance using fund specific historical operating characteristics [9]. The study also evaluates the effect on model performance by including heuristic variable selections for neural networks compared to a stepwise variable selection for a linear regression model. The study proposes the use of a general-purpose nonlinear optimizer, which automatically selects the number of hidden layer nodes for the ANN to achieve optimal model performance. In this particular study, three types of mutual fund strategies were studied: growth, value, and blend. A value fund invests in undervalued stocks, a growth fund invests in stocks that are anticipated to have high earning growth potentials, and a blend fund incorporates both growth and value strategies[9].

Neural networks were trained with one hidden layer, seven input nodes, one output node, and 9-12 hidden nodes (varying based on the fund strategy that was studied) using a mutual fund database that contained monthly return data for approximately one thousand funds from 1993-1995. Linear regression models were also constructed to compare with the neural network
performance. The results of the study showed that for growth and blend funds, neural networks delivered superior predictive performance. Additionally, the models with variable selection performed better compared to models that did not have variable selection included[9].

4.1.2 **Mutual Fund Net Asset Value Forecasting Using Artificial Neural Networks**

A study published in 1996 evaluated the performance of a backpropagation ANN to estimate the end of year net asset values (NAV) for mutual funds and compared the model performance with linear and nonlinear regression models[10]. Unlike the previous study, this study uses economic risk factors for inputs to the network as opposed to fund specific operating characteristics. Some of the inputs used in the study include the unemployment rate, inflation rate, treasury bill rate, the level of spending in by households, the level of investment spending by firms, and the level of US government spending. The ANN has a three layer structure, with 15 neurons in the input layer, 20 neurons in the hidden layer, and 1 neuron in the output layer. The study demonstrated that mean absolute percent error was 8.76% for the ANN, 15.16% for the linear regression model, and 21.93% for the nonlinear regression model. The ANN clearly outperformed both linear and nonlinear regression models using out of sample data. The authors further conclude that the network has particularly strong performance compared to linear and nonlinear models when limited data is available. The authors further state that ANN performance could be reasonably improved by using newer ANN models, as they used only a simple back propagation ANN[10].

4.1.3 **Stock Price Prediction Using Neural Networks**

Modeling stock price changes using neural networks is the subject of many papers. A 2011 paper proposes a new type of ANN known as a Wavelet De-noising-based Back Propagation neural network (WDBPNN). The WDBPNN effectively predicted the monthly closing price data
for the Shanghai Composite Index from January 1993 to December 2009 [11]. The WDBPNN outperformed a standard back propagation neural network using the same sets of data for training and validation. The WDBPNN model is particularly successful because it is capable of decomposing the original data set into multiple layers in an attempt to separate the noise in the data from the trends in the data. The network is then trained on the layers that exclude the noise in the data, allowing the network to make more precise predictions.

A 2005 paper studies the performance of linear regression models compared to ANNs for stock price prediction performance. The paper compares the difference in performance from selecting inputs from the capital asset pricing model (CAPM) or Fama and French’s 3-factor model [12][13]. The paper concludes that the CAPM inputs provide better performance for both the ANN and linear regression models when predicting stock prices from the NASDAQ, AMEX and NYSE. More importantly, the paper demonstrates that the ANN model better predicts the stock prices compared with regression models. There are several other papers that continue to demonstrate ANNs superior predictive performance[11, 14-16].

4.1.4 **NEURAL NETWORKS AND HEDGE FUNDS**

Neural networks have been used in hedge funds, mainly, to derive benchmark indices for the different styles of hedge funds using a Self-Organizing Map (SOM), which is a neural network classification tool[17],[18].

4.2 **HEDGE FUNDS**

Hedge funds only became popular as an investment program in the 90s which is why their returns have only recently been modeled in the literature. Hedge fund predictability was first investigated in 2003 when multifactor linear models were designed and these models demonstrated that certain classifications of hedge funds could have their returns predicted [19].
Another study in 2006 demonstrated a regression model with improved predictability by using a larger data set and a larger set of risk factors. It outlined potential asset allocation strategies that could deliver returns above passive benchmarks [20]. Another model outlined in an article in 2008 improves on model formulation and predictability using a Bayesian averaging model [21]. Prediction accuracy for hedge fund models was further improved using a backward elimination method with a rolling in-sample estimation window. By accounting for heteroscedasticity and non-normality, prediction accuracy was improved [22].

A 2008 paper examined the relationship between the performance of a hedge fund and its exposure to risk factors. The higher the exposure to a risk factor, the more predictable the fund is. The authors hypothesized that funds that were less predictable were more likely to have stronger overall performance. The hypothesis stems from the belief that if a fund’s performance is strongly explained by risk factors, the performance of the fund likely follows market movements and therefore the role of the manager does not have as strong of an influence on the fund performance. On the other hand, if a fund is not predictable, the fund’s performance is more difficult to model, meaning that the manager is playing a more active role in the fund’s performance which allows for greater returns. The paper verified the hypothesis, showing that top performing hedge funds that were modeled using regression techniques had a lower R-squared value compared to poorly performing hedge funds[23].

4.3 Risk Factors
Since individual hedge funds have different investment strategies and asset holdings, a large set of risk factors should be considered when modeling the wide range of hedge fund return profiles. Research from Liang used eight factors[24], Schneeweis and Spurgin used 13 factors[25], Brealey and Kaplan used 31 factors[26], and Titman and Tiu used 27 factors[23].
The set of factors from all of these studies represents almost all of the factors that have been used to explain hedge fund performance in the literature. Titman and Tiu by examining twenty-four (24) inputs that represent domestic equity factors, international equity performance factors, domestic fixed income factors, international fixed income and foreign exchange factors, commodities factors, and nonlinear factors. Titman and Tiu’s twenty-four factors represent the majority of hedge fund risk factors that have been discussed in the literature[23]. The comprehensive set of risk factors are detailed further since we use a similar set of risk factors for inputs to our models.

The domestic equity factors they use are the Russel 3000, the NASDAQ and NAREIT and indices, the Fama and French size, value and momentum factors[23]. The international equity factors they use are the FTSE100 index, the NIKKEI 225, the Morgan Stanley Capital International (MSCI) index EAFE, the Morgan Stanley International Emerging Markets Free (MSCI EMF) index, and the DAX and CAC 40 indices[23].

The domestic fixed income factors they use are the Lehman Brothers Aggregate Bond index, the Salomon Brothers 5 year index of Treasuries, the default spread, the duration spread, the Lehman Brothers Aggregate of Mortgage Backed Securities, and the Lehman Brothers index of 10 year maturity Municipal Bonds[23]. The international fixed income / foreign exchange factors they use are the Salomon Brothers non-US Weighted Government Bonds index with a 5 to 7 year duration, and the Salomon Brothers non-US Unhedged Dollar index[23].

The commodities factors they use are the Goldman Sachs Commodity Index, the AMEX Oil index, and the returns on gold[23].
The nonlinear factors they use are the Primitive Trend Following Strategies factors for bonds, stock, currencies and commodities which are derived by Fung and Hsieh[23]. Fung and Hsieh described the Primitive Trend Following Strategies (PTFS) factors in order to capture the possibility of hedge funds that employ options or option-like payoffs which would affect their returns[27].
5 Data

The Data section describes the input and target data used in the models. In particular, we detail the time periods we studied, the risk factors we selected, the sources of the risk factors, the hedge fund data we studied, the source of the hedge fund data, and the data normalization process.

5.1 Hedge Fund Data

Hedge funds are not required by government regulations to disclose their performance results. Hedge fund data that is available is voluntarily provided by individual hedge funds. Data is collected by data vendors and can be provided, along with performance indices, to investors and researchers for a fee.

For this thesis, we purchased a North American hedge fund database from Eurekahedge, a reputable hedge fund research house and data vendor. The database contains 2,352 unique live funds and 1,954 discontinued funds with return and assets-under-management profiles spanning from June 2011 to as far back as the 1980s. Hedge fund classifications, performance characteristics, location, and many other unique fund characteristics are also provided.

5.1.1 Hedge Fund Database Biases

Hedge fund databases are subject to biases. The main source of these biases is from the fact that hedge funds are self-reporting their data to the databases. Four of these biases that are documented in the literature are survivorship bias, non-reporting bias, questionable-numbers bias, and instant-return-history bias [20].

5.1.1.1 Survivorship bias

When a fund closes due to poor performance, naturally its performance is no longer reported in the hedge fund database. As a result, there is a tendency for the average returns in a
hedge fund database to be higher than what they actually may be. This bias can be reduced by including available defunct fund data in the data set. However, this bias is only important to remove or reduce when evaluating and comparing hedge fund performance [20]. Within the scope of this thesis, this bias is not important to correct for.

5.1.1.2 Non-reporting bias
Hedge funds are not required to report their returns and therefore databases do not encompass the entire hedge fund industry. Furthermore, some hedge funds report their data to only certain databases, meaning individual databases may demonstrate different trends for the same reporting periods if for no reason than they have a different data mix. Naturally, there is bias as a result of this, which is known as non-reporting bias [20]. Similar to survivorship bias, this bias does not have to be corrected for.

5.1.1.3 Questionable-numbers bias
Hedge fund data is submitted from the hedge fund and are not externally verified. Therefore, there is the possibility that data are prone to errors or have been purposely misreported. This bias can affect the results of the study as outliers can significantly reduce regression performance, but neural network performance is less sensitive to outliers [20]. Regardless, it is not reasonably possible to correct for this bias, but the study results should not be adversely affected from the existence of this bias.

5.1.1.4 Instant-return-history bias
Instant-history bias is when a manager controls multiple hedge funds with different strategies without reporting their returns. Then, after a few years, the manager reports the returns of the successful funds. As a result, a higher than expected average return would be calculated in hedge fund databases. Again, in the context of modeling hedge fund returns, this bias does not have to be corrected for [20].
5.2 TIME PERIODS STUDIED

This thesis creates performance models for hedge funds over two different time periods. The first time period is from December 1998 to May 2006 (N=90). For this time period, the training set is from December 1998 to February 2005 (N=75), and the validation set is from March 2005 to May 2006 (N=15). The second time period is from December 1998 to November 2010 (N=144). For this time period, the training set is from December 1998 to December 2006 (N=97), and the validation set is from January 2007 to November 2010 (N=47). The models produced using the first time period dataset serve as a baseline to compare the performance of the models during the recent financial crisis, which are produced using the second time period dataset.

The training data set is used to construct the linear regression and ANN models. The model predictive performance is then measured by using the validation data as an out of sample test as shown in Figure 4.

We created linear regression and neural network models for funds that had complete monthly return history over both study periods. Based on this requirement, 293 funds were studied.

![Figure 4: The two study periods and their corresponding training and validation sections.](image-url)
5.3 **SELECTED RISK FACTORS**

The set of factors from all of these studies represents almost all of the factors that have been used to explain hedge fund performance in the literature. We follow Titman and Tiu by examining twenty-five (25) inputs that represent domestic equity factors, international equity performance factors, domestic fixed income factors, international fixed income and foreign exchange factors, commodities factors, and nonlinear factors. When we could not access the exact risk factor Titman and Tiu used, we used a different risk factor that had similar characteristics[23]. Risk factor historical data was obtained online using Yahoo! Finance, original authors’ data websites (where noted), or from the Yield Book for Citigroup indices.

5.3.1 **DOMESTIC EQUITY FACTORS**

We use the following seven risk factors to capture US equity returns: (1) Russel 3000 index, (2) NASDAQ Composite index, (3) NASDAQ 100 index, (4) National Association of Real Estate Investment Trusts® (NAREIT) index, (5) Small Minus Big (SMB) factor, (6), High Minus Low (HML) factor, and, (7) Momentum factor (MOM) factor. The SMB, HML, and MOM factors are derived by Fama and French and are used to explain anomalies in hedge fund returns. The SMB factor calculates the difference between the average return of three small portfolios and three big portfolios, the HML factor calculates the difference between the average return on two value portfolios and two growth portfolios, and the MOM factor calculates the difference between the average return of two high prior return portfolios and two low prior return portfolios¹[12].

5.3.2 **INTERNATIONAL EQUITY FACTORS**

---

¹Fama/French factor data are obtained online from Kenneth French’s website (http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/Data_Library/f-f_factors.html)
We use the following six factors to capture international equity returns: (1) FTSE100 Index, (2) NIKKEI 225 Index, (3) MSCI EAFE (Morgan Stanley Capital International Europe, Australasia, and Far East) Index, (4) MSCI EM (Emerging Markets) Index, (5) DAX (DuetscherAktienIndeX), and, (6) CAC40 Index.

5.3.3 Domestic Fixed Income Factors
We use the following four factors to capture domestic fixed income returns: (1) Citigroup World Government Bond Index-US Portion (Citi WGBI-US), (2) Citigroup US Broad Investment-Grade Index-Mortgage Portion (Citi USBIG-Mortgage), (3) the entire Citigroup US Broad Investment-Grade Index, and, (4) Citigroup World Broad Investment-Grade Index-US portion (Citi WBIG-US).

The Citigroup WGBI index calculates an index for bond based fixed-income for different geographical regions. Citigroup shows how the WGBI index performance is broken down each region, and therefore for a domestic fixed income factor we choose to use only the US based portion of the WGBI index. The same holds true for the Citigroup USBIG index, as we choose to use only the mortgage section of US broad investments-grade index as another domestic fixed income factor that may capture hedge fund returns. Finally, we choose to only use the US section of the Citigroup Broad Investment-Grade index in order to represent domestic fixed-income broad investment performance that may be captured by hedge fund returns.

5.3.4 International Fixed Income Factors
We use the following two factors to capture domestic fixed income returns: (1) Citigroup World Government Bond Index-non US Portion, (2) Citigroup World Broad Investment-Grade Index-non US portion.

5.3.5 Commodities Factors
We use the following two factors to capture commodity returns: (1) International Monetary Fund Commodity AllIndex, and, (2) International Monetary Fund Commodity Non-Fuel Index.

5.3.6 Non-Linear Factors

Fung and Hsieh described Primitive Trend Following Strategies (PTFS) that capture the possibility of hedge funds that employ options or option-like payoffs which would affect their returns[27]. The PTFS use “lookback straddles” which have payoffs that are equal to the difference of the maximum and minimum price of the underlying assets during their option life. Based on these lookback straddles, Fung and Hsieh formed five PTFS portfolios based on stocks, bonds, three-month interest rates, currencies, and commodities. The portfolios have been shown to have high predictive power on hedge funds, which is why we include the PTFS for bonds, stocks, currencies, and commodities in our models [27].

Risk factor historical data was obtained online using Yahoo! Finance, original authors’ data websites (where noted), or from the Yield Book for Citigroup indices. The selected risk factors that were discussed in the literature review are summarized, with their corresponding sources, in Table 2.

---

2 PTFS data are obtained from David Hsieh’s website (http://faculty.fuqua.duke.edu/~dah7/DataLibrary/TF-FAC.xls).
<table>
<thead>
<tr>
<th>Risk Factor</th>
<th>Source</th>
<th>Risk Factor</th>
<th>Source</th>
<th>Risk Factor</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Russel 3000</td>
<td>Yahoo Finance</td>
<td>Nikkei 225</td>
<td>Yahoo Finance</td>
<td>Citi WBGI-</td>
<td>nonUS Yield Book</td>
</tr>
<tr>
<td>NASDAQ Composite</td>
<td>Yahoo Finance</td>
<td>MSCI EAFE Index</td>
<td>Yield Book</td>
<td>Citi WBIG-</td>
<td>Yield Book</td>
</tr>
<tr>
<td>NASDAQ 100</td>
<td>Yahoo Finance</td>
<td>MSCI EM Index</td>
<td>Yield Book</td>
<td>Citi US BIG</td>
<td>Yield Book</td>
</tr>
<tr>
<td>NAREIT index</td>
<td>Yahoo Finance</td>
<td>DAX</td>
<td>Yahoo Finance</td>
<td>IMF Commodity</td>
<td>Index (All) IMF Data Library</td>
</tr>
<tr>
<td>SMB</td>
<td>Fama French Data Library</td>
<td>CAC40</td>
<td>Yahoo Finance</td>
<td>IMF Commodity</td>
<td>Index (Non- Fuel Index) IMF Data Library</td>
</tr>
<tr>
<td>HML</td>
<td>Fama French Data Library</td>
<td>Citi WGBI-US</td>
<td>Yield Book</td>
<td>PTFS for bonds, stocks, currency, and commodities Fung and Hsieh Data Library</td>
<td></td>
</tr>
<tr>
<td>MOM</td>
<td>Fama French Data Library</td>
<td>Citi US BIG-Mortgage</td>
<td>Yield Book</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FTSE100</td>
<td>Yahoo Finance</td>
<td>Citi WBIG-US</td>
<td>Yield Book</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 5.4 Data Normalization

The collected risk factor data is all reported in monthly percent change and is normalized to a range of -1 to +1 by dividing each individual risk factor dataset by the maximum absolute percent change from the same dataset. The normalization is required since neural networks require inputs to be in the range from -1 to +1 [8]. Regression performance is not affected as the normalization process is simply a scaling of the data.

Similar to risk factor normalization, the hedge fund return data is normalized by dividing the return data for all funds by the maximum absolute return ever obtained by the funds as shown in Table 3. As a result, the funds’ returns and model performances can still be compared as the funds all have been scaled by the exact same value. This normalization is performed because neural networks require their outputs to be in the range of -1 to +1 [8].
As shown in Table 3, the hedge fund returns (or, the target data) are normalized by dividing all values by the maximum absolute value along the entire time period, which in this case is 10%. The same process is conducted for each and every risk factor input.

*Table 3: A table demonstrating the input and target normalization process.*

<table>
<thead>
<tr>
<th>Month</th>
<th>Hedge Fund #1 Return</th>
<th>Hedge Fund #2 Return</th>
<th>Norm Return Hedge Fund #1</th>
<th>Norm Return Hedge Fund #2</th>
<th>Input #1</th>
<th>Norm Input #1</th>
<th>Input #2</th>
<th>Norm Input #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feb 03</td>
<td>5%</td>
<td>1%</td>
<td>0.5</td>
<td>0.1</td>
<td>4%</td>
<td>0.8</td>
<td>9%</td>
<td>0.45</td>
</tr>
<tr>
<td>Mar 03</td>
<td>10%</td>
<td>2%</td>
<td>1</td>
<td>0.2</td>
<td>2%</td>
<td>0.4</td>
<td>1%</td>
<td>0.05</td>
</tr>
<tr>
<td>Apr 03</td>
<td>2%</td>
<td>-4%</td>
<td>0.2</td>
<td>-0.4</td>
<td>1%</td>
<td>0.2</td>
<td>20%</td>
<td>1</td>
</tr>
<tr>
<td>May 03</td>
<td>-2%</td>
<td>-9%</td>
<td>-0.2</td>
<td>-0.9</td>
<td>-5%</td>
<td>-1</td>
<td>-8%</td>
<td>-0.4</td>
</tr>
</tbody>
</table>
6 EXPERIMENTAL APPROACH

The Experimental Approach section outlines the construction of the linear regression models, the neural network model training process, neural network parameter selection, a detailing of stepwise backwards elimination factor selection, and the chosen performance benchmarks that we used to evaluate the accuracy of our models.

Neural network models are constructed to examine whether their demonstrated nonlinear modeling power in financial modeling problems can be taken advantage of in hedge fund performance modeling [9-16]. Successful hedge fund models have been constructed in the literature and therefore, with correct network construction and data selection, neural networks should be able to successfully model hedge fund returns [17-27]. Linear regression models are constructed to serve as a benchmark for the neural network models.

We use stepwise regression backwards elimination factor selection to minimize the amount of inputs we feed into our model based on the fact that not all of the risk factors we have chosen will have explanatory behavior on all of the 293 funds we study [23]. By minimizing the inputs we feed into our networks, we should see improvements in neural network performance as it will be less likely for overtraining to occur since the amount of weighted connections will have decreased, satisfying the rule of thumb that dictates that the number of data points should be at least double the number of weighed connections [8].

6.1 LINEAR REGRESSION MODELS

As a base comparison model, multi linear regression models are created for all funds over both study periods using all 25 risk factors as inputs as well as including an intercept term in the model for a total of 26 parameter estimates as shown in equation 6:
\[ y^N_t = m_N x_t + b \]  

Where \( y^N_t \) is the hedge fund monthly return at time=t for fund=N, \( m_N \) is a 1x25 parameter estimate vector that represents the 25 coefficient parameter estimates for fund=N, \( x_t \) is a 25x1 vector that represents the set of 25 risk factor inputs at time=t, and \( b \) is the intercept estimate term.

### 6.2 Neural Network Training

Training neural networks is not an exact science in that certain parameters have to be specified before training, such as the number of hidden layers, the number of neurons for each layer, the transfer functions for each layer, the training function, and when should the network stop training. There is no exact way to determine which choices and parameter values results in the best network performance, but a few assumptions can be made to aid the decision making process.

First, the number of hidden layers is set to one, as any additional hidden layers will most likely lead to overtraining. An easy way to visualize this is to think that each hidden layer adds a significant number of parameters that the model can use to fit to the data as opposed to the pattern presented by the data. Since we use only 75 training points in the first study period, it is highly likely that this occurs with more than one hidden layer.

Since we are not using a vast amount of training data, networks train very quickly. As a result, we can train many networks with the same datasets to see the effect of the number of hidden layer neurons on network performance. We trained sixteen (16) networks for each time period. The first eight networks are trained using all 25 risk factors as inputs, and the networks only vary with their hidden layer neuron count from one neuron to eight neurons. The next eight
networks are trained using only the risk factors that remained after the stepwise regression backwards elimination method was conducted on the multi linear regression model. Again, the networks only vary their hidden layer neuron count. Therefore, there are thirty-two (32) networks trained for each individual fund over both study periods.

Based on a crude rule of thumb[8], it is important that a network has at least twice as many data points compared to the number of neuron connections in the network. For example, a network with 25 input neurons, two hidden layer neurons and one output layer neuron has 50 connections (25*2 + 2*1), which is more than half of the 94 data points in the first study period and less than half of the 144 data points in the second study period. Therefore, it is likely that networks evaluated in this thesis with at least three hidden layer neurons will have significantly poorer performance compared to networks with one or two hidden layer neurons [8].

The transfer functions used are tan-sigmoid functions for the hidden layer and linear functions for the output layer. The tan-sigmoid function is shown in equation 7 and the linear transfer function is shown in equation 8:

$$\varphi_{\text{tansig}}(n) = \frac{2}{1 + e^{-2n}} - 1 \quad (7)$$

$$\varphi_{\text{linear}}(n) = n \quad (8)$$

Where $n$ represents the input to the transfer function.

These transfer functions tend to be the default choices and yield little to no change in performance compared to different combinations of logarithmic-sigmoid, tan-sigmoid, and linear transfer functions.
From the literature review, we use resilient backpropagation as the training function as it tends to avoid local minima and trains very quickly.

Finally, during each training epoch, the validation data set performance is calculated. The change in validation performance is compared to the performance during the previous epochs. Once validation performance stops improving, the network stops training as any further training epochs results in worse validation performance.

6.3 **STEPWISE REGRESSION**

For each individual hedge fund, a subset of the 25 risk factors may be able to equally or better model its return profile compared to a multi linear regression with all 25 risk factors. In order to decide which set of risk factors are chosen for each hedge fund, a stepwise regression backwards elimination approach is used using the stepwisefit function in Matlab.

Stepwise regression is the process of adding and removing factors from a multi linear model based on the statistical significance of the terms in the regression. Based on the initial model, the method compares the explanatory power of each new model as terms are removed or added. During each iteration, the $p$-value of an $F$-statistic is computed for models that include and do not include each individual risk factor. For a term that is not currently in the model, the null hypothesis is that the term’s coefficient would be zero in the model. If the statistic indicates that the null hypothesis can be rejected, then the term is added. On the other hand, for a term that is in the model, the null hypothesis is that the term’s coefficient is zero in the model. If the hypothesis cannot be rejected, then the term is removed from the model. The hypotheses are rejected based on the $p$-enter and $p$-exit boundaries that are specified by the user. We used the default parameters suggested by Matlab as they yielded excellent performance improvements.
which are discussed in the results section. The stepwise process works using the following three steps:

(1) The initial model is fitted.

(2) If any of the terms not included in the model have \( p \)-values less than an entrance tolerance of 0.05 (i.e. reject the null hypothesis), then add the term with the lowest \( p \)-value. Repeat this step until terms can no longer be added, then proceed to step three.

(3) If any of the terms included in the model have \( p \)-values greater than an exit tolerance of 0.10 (i.e. the null hypothesis cannot be rejected), then remove the term with the highest \( p \)-value and repeat step two. If terms can no longer be removed, then the stepwise regression has finished.

We use a stepwise regression backward elimination approach, meaning that our initial model is fitted with all of the terms. Terms are removed first and then the method evaluates whether to add them back, as opposed to forward selection where terms are added first and the method evaluates whether to remove them again.

It is important to note that the stepwise regression is not globally optimal. Depending on the initial model and the \( p \)-enter and \( p \)-exit values, the method may converge on a model that has a subset of risk factors that may have poorer performance compared to a model that has a different subset of risk factors.

6.4 PERFORMANCE EVALUATION

The mean-squared error (MSE) is calculated for each individual linear regression model, stepwise regression model, and neural network model. The models are created using the training data from the corresponding Study Period, and the models are tested using the validation data.
from the Study Period. The output of the model using the validation data is compared to the target data (what actually happened), and the MSE is calculated using these two values.

As the MSE decreases, the performance of the model increases. Since all of the network outputs were normalized using the same scaling factor, the MSE of all models can be compared to one another as is. By doing this, 18 sets of 293 MSE values are calculated for a total of 5274 MSE values. The 18 sets are (1) regression models, (2) SW regression models, (3-10) neural network model with 1-8 hidden layer neurons, and, (11-18) SW neural network models with 1-8 hidden layer neurons.

For each set, the average and median MSE’s are calculated with corresponding standard deviations. These values are used to evaluate which models have the best performance for modeling hedge fund returns. The MSE is calculated using equation 9:

\[
MSE = \frac{\sum_{i=1}^{n}(Output_i - Target_i)^2}{n}
\]  

Where \( n \) is the number of instances in the validation set, \( Output_i \) is the output value of the model at time=\( i \), and \( Target_i \) is the actual value of the hedge fund return at time=\( i \). Since the hedge fund monthly return is reported in percentage, the MSE and standard deviation values are reported in percentage squared (\( \%^2 \)) units.

As well, the standard deviations for the models are calculated using equation 10:

\[
\sigma = \sqrt{\frac{\sum_{i=1}^{n}(SE_i - MSE)^2}{n}}
\]  

Where \( n \) is the number of instances in the validation set, \( SE_i \) is the squared-error of the model at time=\( i \), and \( MSE \) is the mean-squared error for all \( n \) model outputs.

The performance of the regression models compared to the neural network models are also evaluated by counting how many funds are better modeled by regression models or neural
networks. For example, consider the hypothetical fund data in Table 4. The table shows that for the first five hypothetical funds we study, three of the funds are best modeled by a neural network (two networks with one hidden layer neuron, and one network with two hidden layer neurons) and two of the funds are best modeled by linear regression. We then count the number of funds that are best modeled by regression and neural networks. Therefore, when we evaluate our 293 funds, we use this approach to gauge in another way which model has better predictive performance. For example, if out of the 293 funds 230 were better modeled by neural networks, then it would be clear that a neural network model has better predictive performance compared to a linear regression model.

<table>
<thead>
<tr>
<th>Fund #</th>
<th>Linear Regression MSE</th>
<th>NN 1 MSE</th>
<th>NN 2 MSE</th>
<th>Best Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5%</td>
<td>3%</td>
<td>6%</td>
<td>NN 1</td>
</tr>
<tr>
<td>2</td>
<td>2%</td>
<td>3%</td>
<td>4%</td>
<td>Linear Regression</td>
</tr>
<tr>
<td>3</td>
<td>10%</td>
<td>1%</td>
<td>12%</td>
<td>NN 1</td>
</tr>
<tr>
<td>4</td>
<td>9%</td>
<td>10%</td>
<td>5%</td>
<td>NN 2</td>
</tr>
<tr>
<td>5</td>
<td>3%</td>
<td>12%</td>
<td>4%</td>
<td>Linear Regression</td>
</tr>
</tbody>
</table>

Count: Linear Regression: 2  Neural Networks: 3
7 RESULTS AND DISCUSSION

The results and discussion section reviews stepwise regression results, model formulation results for the trained neural networks and linear regression models, and their stepwise regression backward elimination (SW) variants, for both time periods.

7.1 STEPWISE REGRESSION RESULTS

For each of the 293 funds, a backwards elimination stepwise regression was executed to select a subset of the 25 risk factors we use as inputs for our models. Table 5 and Figure 5 summarize the number of risk factors that are selected as inputs for the fund models after using stepwise regression for factor selection.

Table 5: A summary of the stepwise regression backwards elimination results.

<table>
<thead>
<tr>
<th>Factors Included In Models With Stepwise Regression for 293 Hedge Funds for Time Period #1</th>
<th>Average Factors Per Fund = 6.21</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factor 1</td>
<td>1</td>
</tr>
<tr>
<td>In Funds</td>
<td>46</td>
</tr>
<tr>
<td>Factor</td>
<td>14</td>
</tr>
<tr>
<td>In Funds</td>
<td>66</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Factors Included In Models With Stepwise Regression for 293 Hedge Funds for Time Period #2</th>
<th>Average Factors Per Fund = 6.68</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factor 1</td>
<td>1</td>
</tr>
<tr>
<td>In Funds</td>
<td>57</td>
</tr>
<tr>
<td>Factor</td>
<td>14</td>
</tr>
<tr>
<td>In Funds</td>
<td>72</td>
</tr>
</tbody>
</table>

Risk Factor Legend

- Factor 1: Russel 3000
- Factor 2: Russel Comp
- Factor 3: Nasdaq 100
- Factor 4: NAREIT
- Factor 5: FF Size
- Factor 6: FF HML
- Factor 7: FF MOM
- Factor 8: FTSE100
- Factor 9: NIKKEI225
- Factor 10: MSCI EAFE
- Factor 11: MSCI EM
- Factor 12: DAX
- Factor 13: CAC40
- Factor 14: Citi US WGBI
- Factor 15: Citi USBIG MTG
- Factor 16: Citi USBIG
- Factor 17: Citi WorldBIG US
- Factor 18: Citi non-US WGBI
- Factor 19: Citi non-US WorldBig
- Factor 20: F&H PTFS BD
- Factor 21: F&H PTFS FX
- Factor 22: F&H PTFS COM
- Factor 23: F&H PTFS STK
- Factor 24: IMF Commodity AllIndex
- Factor 25: IMF non-fuel Commodity

39
Figure 5: A comparison of the number of risk factors contained in specific fund models for both time periods.

The three Fama and French factors, the Nasdaq 100, MSCI EAFE, MSCI EM, CAC40 are represented in over 100 of the 293 fund models after we used stepwise regression. Between the two time periods, the number of factors in each fund model do not vary much as clearly shown in Figure 5.

Based on stepwise regression, the 293 funds were best modeled by an average of 6.21 risk factors for Time Period #1 and 6.68 risk factors for Time Period #2. It is clear from these results that there is no ideal subset of the 25 risk factors to model all 293 funds that we study. Instead, a unique subset of the 25 risk factors must be chosen for each individual fund to achieve better model performance.

7.2 TIME PERIOD #1 (DEC 1998-MAY 2006)

For the first study period, 18 models were formed for each of the 293 individual funds: two regression models, one with all 25 risk factors included in the model, and one with risk factors
selected using SW; and sixteen neural networks, eight with all 25 risk factors included in the network with varying hidden layer units from 1-8, and eight with risk factors selected using SW with varying hidden layer units from 1-8. For each model type, we calculate the in sample and out of sample average MSE and standard deviation in order to visualize which model type can best model hedge fund monthly returns.

The in sample and out of sample performance of the models are shown in Table 6. When evaluating the in sample neural network performance, it is important to note that the neural networks stopped training when the validation data mean-squared error began to increase. As a result, the in sample performance of the neural networks is not a true estimate of the model’s ability to predict hedge fund returns, and therefore we examined the out of sample (validation) performance of the neural networks to evaluate neural networks’ predictive ability.
Table 6: Performance statistics for regression and ANN models for Time Period #1.

**Time Period #1:** December 1998 to May 2006 (N=90). Validation set from March 2005 to May 2006 (N=15). Mean and median values are based on 293 results for each model type.

### Training Data (In Sample) Performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Average MSE (%)</th>
<th>Average STD (%)</th>
<th>Median MSE (%)</th>
<th>Median STD (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>13.4</td>
<td>22.8</td>
<td>16.3</td>
<td>29.6</td>
</tr>
<tr>
<td>NN 1</td>
<td>37.0</td>
<td>54.7</td>
<td>26.9</td>
<td>49.6</td>
</tr>
<tr>
<td>NN 2</td>
<td>42.7</td>
<td>76.4</td>
<td>40.2</td>
<td>85.3</td>
</tr>
<tr>
<td>NN 3</td>
<td>87.6</td>
<td>132.0</td>
<td>68.5</td>
<td>141.8</td>
</tr>
<tr>
<td>NN 4</td>
<td>203.9</td>
<td>321.4</td>
<td>73.7</td>
<td>174.5</td>
</tr>
<tr>
<td>NN 5</td>
<td>242.3</td>
<td>346.7</td>
<td>99.1</td>
<td>209.3</td>
</tr>
<tr>
<td>NN 6</td>
<td>231.6</td>
<td>329.7</td>
<td>120.4</td>
<td>243.1</td>
</tr>
<tr>
<td>NN 7</td>
<td>311.0</td>
<td>486.9</td>
<td>139.1</td>
<td>273.4</td>
</tr>
<tr>
<td>NN 8</td>
<td>321.6</td>
<td>496.3</td>
<td>213.2</td>
<td>399.6</td>
</tr>
</tbody>
</table>

### Validation Data (Out Of Sample) Performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Average MSE (%)</th>
<th>Average STD (%)</th>
<th>Median MSE (%)</th>
<th>Median STD (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>17.1</td>
<td>10.5</td>
<td>8.8</td>
<td>5.4</td>
</tr>
<tr>
<td>NN 1</td>
<td>28.5</td>
<td>37.1</td>
<td>10.7</td>
<td>13.5</td>
</tr>
<tr>
<td>NN 2</td>
<td>25.4</td>
<td>32.8</td>
<td>11.3</td>
<td>14.6</td>
</tr>
<tr>
<td>NN 3</td>
<td>87.7</td>
<td>121.2</td>
<td>30.7</td>
<td>40.9</td>
</tr>
<tr>
<td>NN 4</td>
<td>148.3</td>
<td>200.1</td>
<td>41.4</td>
<td>54.4</td>
</tr>
<tr>
<td>NN 5</td>
<td>187.0</td>
<td>235.9</td>
<td>64.1</td>
<td>81.6</td>
</tr>
<tr>
<td>NN 6</td>
<td>217.9</td>
<td>297.0</td>
<td>83.1</td>
<td>112.9</td>
</tr>
<tr>
<td>NN 7</td>
<td>266.5</td>
<td>338.7</td>
<td>112.3</td>
<td>145.7</td>
</tr>
<tr>
<td>NN 8</td>
<td>291.6</td>
<td>392.9</td>
<td>136.4</td>
<td>180.4</td>
</tr>
</tbody>
</table>

From Table 6, the regression average MSE of 17.1%\(^2\) for a regression model was calculated by constructing 293 linear regression models for the 293 selected hedge funds using the training data from Study Period #1. The validation data from Study Period #1 was then passed through each constructed linear regression model, and the MSE for each individual model was calculated based on the outputs of the models from the validation set input data and corresponding target data. The 293 MSE values were then averaged, resulting in the 17.1%\(^2\) shown in the table.

Table 6 demonstrates that regression models and neural network models with one to two hidden layer units have smaller MSE’s (higher performance) and smaller standard deviations. In particular, neural networks with SW factor selection and one hidden layer neuron achieved the lowest average MSE of 12.6%\(^2\), which is 9% better than the MSE of 13.8%\(^2\) for the linear
regression with SW risk factor selection model. However, the average standard deviation for the network is still higher at 16.2 %$^2$ compared to 6.6 %$^2$ for the regression model. Furthermore, Table 6 shows that with SW factor selection included, the MSE and standard deviation decreases across all models, or, performance of all models increases with SW factor selection.

**Table 7: Funds better modeled by a neural network or a linear regression model for Time Period #1.**

<table>
<thead>
<tr>
<th>Units</th>
<th>NN</th>
<th>Regression</th>
<th>NN + SW</th>
<th>Regression + SW</th>
<th>NN</th>
<th>NN + SW</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN 1</td>
<td>125</td>
<td>168</td>
<td>216</td>
<td>77</td>
<td>58</td>
<td>235</td>
</tr>
<tr>
<td>NN 2</td>
<td>122</td>
<td>171</td>
<td>162</td>
<td>131</td>
<td>87</td>
<td>206</td>
</tr>
<tr>
<td>NN 3</td>
<td>43</td>
<td>250</td>
<td>112</td>
<td>181</td>
<td>66</td>
<td>227</td>
</tr>
<tr>
<td>NN 4</td>
<td>27</td>
<td>266</td>
<td>62</td>
<td>231</td>
<td>62</td>
<td>231</td>
</tr>
<tr>
<td>NN 5</td>
<td>15</td>
<td>278</td>
<td>46</td>
<td>247</td>
<td>57</td>
<td>236</td>
</tr>
<tr>
<td>NN 6</td>
<td>6</td>
<td>287</td>
<td>33</td>
<td>260</td>
<td>56</td>
<td>237</td>
</tr>
<tr>
<td>NN 7</td>
<td>6</td>
<td>287</td>
<td>25</td>
<td>268</td>
<td>47</td>
<td>246</td>
</tr>
<tr>
<td>NN 8</td>
<td>1</td>
<td>292</td>
<td>21</td>
<td>272</td>
<td>46</td>
<td>247</td>
</tr>
</tbody>
</table>

From Table 7, the 125 and 168 values from the first row and first two columns illustrate that for the 293 funds, 125 of the funds had a lower MSE (better performance) with a neural network model with one hidden layer neuron compared to the regression model. This means that 163 of the funds had a lower MSE (better performance) with a linear regression model compared to the neural network model with one hidden layer neuron. Therefore, regression models have better predictive performance compared to neural networks with one hidden layer neuron. However, the result is reversed when stepwise regression is used, as 216 of the 293 funds had a lower MSE (better performance) with a neural network with one hidden layer neuron. Finally, the last two columns clearly illustrate that any model with stepwise factor selection has a lower MSE (better performance) compared to a model that does not have stepwise factor selection.

Table 7 demonstrates that the number of funds that achieved better fits with neural network models decreases as the number of hidden layer units increases. In fact, without SW factor selection, 168 of the 293 funds are better fit with linear regression compared to a one hidden
layer unit neural network. However, with SW factor selection, only 77 of the 293 funds are better fit with regression compared to a one hidden layer unit neural network. It is only when neural networks have three hidden layer units does overtraining occur, resulting with 181 of the 293 funds being better fit with regression.

From Figure 6, it is clear that as the number of hidden layer units increase, the MSE and standard deviation of the network significantly increases (performance decreases). The result may be counter-intuitive because, typically, as hidden layer units increase, a neural network should be able to fit the model with greater precision due to the introduction of further parameters. However, since there are only few data points available (N=90), the network is likely fitting itself to the data points, as opposed to the pattern represented by the data points, with the extra parameters. As a result, out of sample performance significantly degrades.
Figure 6 clearly illustrates that a neural network with one hidden layer neuron and stepwise factor selection achieves the lowest MSE, or highest performance. The neural networks with at least four hidden layer neurons were not included in these figures because their error is significantly larger and as a result if we included them it would be difficult to view the
differences in MSE and standard deviation for the regression models and neural network models with 1-3 hidden layer neurons.

From Figure 7, the 125 and 168 values from the first graph for NN 1 illustrate that for the 293 funds, 125 of the funds had a lower MSE (better performance) with a neural network model with one hidden layer neuron compared to the regression model. This means that 163 of the funds had a lower MSE (better performance) with a linear regression model compared to the
neural network. Therefore, regression models have better predictive performance compared to neural networks with one hidden layer neuron. The linear regression model consistently outperforms neural network models as clearly illustrated by the first graph. However, the result is reversed when stepwise regression is used, as 216 of the 293 funds had a lower MSE (better performance) with a neural network with one hidden layer neuron. Neural networks have better performance (lower MSE) compared to regression models with up to two hidden layer neurons when factor selection is included. However, regression models outperform the networks when at least three hidden layer neurons are used with stepwise factor selection.

Figure 8: Stepwise factors selection performance comparison for Time Period #1.

Figure 8 show that neural networks with stepwise factor selection consistently outperform neural networks without factor selection. For example, a neural network with one hidden layer neuron and factor selection had better predictive performance for 235 out of the 293 funds compared to a network with one hidden layer neuron and without factor selection. Table 7 and Figure 7 also reflect this, as 235 out of the 293 funds achieved better modeling performance with a one hidden layer unit neural network that had SW risk factor selection compared to one that did not. Similarly, a linear regression model with SW risk factor selection achieved better
performance with 227 out of 293 funds compared to a regression model without SW risk factor selection. Figure 7 also illustrates that regardless of the number of hidden layer units, the neural network that had SW factor selection consistently performed better than a network that included all 25 risk factors as inputs. The results indicate that funds that have their returns modeled using all 25 risk factors do not necessarily need their returns explained by all of these factors to achieve a similar fit. By lowering the number of inputs, the risk of overfitting due to excessive parameters in the model is lowered, which is likely why SW factor selection reduced the average MSE and standard deviation in the models.

7.3 Time Period #2 (December 1998-May November 2010)

The summary of the performance and standard deviation values for time period #2, which includes the most recent financial crisis, is shown in Table 8. 18 models were formed for each of the 293 individual funds: two regression models, one with all 25 risk factors included in the model, and one with risk factors selected using SW; and sixteen neural networks, eight with all 25 risk factors included in the network with varying hidden layer units from 1-8, and eight with risk factors selected using SW with varying hidden layer units from 1-8. For each model type, we calculate the in-sample and out-of-sample average MSE and standard deviation in order to visualize which model type can best model hedge fund monthly returns.

The in-sample and out-of-sample performance of the models is shown in Table 8. When evaluating the in-sample neural network performance, it is important to note that the neural networks stopped training when the validation data mean-squared error began to increase. As a result, the in-sample performance of the neural networks is not a true estimate of the model’s ability to predict hedge fund returns, and therefore we examined the out-of-sample (validation) performance of the neural networks to evaluate neural networks’ predictive ability.
From Table 8, the regression out of sample average MSE of 37.4\% was calculated by constructing 293 linear regression models for the 293 selected hedge funds using the training data from Study Period #2. The validation data from Study Period #2 was then passed through each constructed linear regression model, and the MSE for each individual model was calculated based on the outputs of the models from the validation set input data and corresponding target data. The 293 MSE values were then averaged, resulting in the 37.4\% shown in the table.

<table>
<thead>
<tr>
<th>Model</th>
<th>Average MSE (In Sample)</th>
<th>Average STD</th>
<th>Median MSE</th>
<th>Median STD</th>
<th>Average MSE (Out Of Sample)</th>
<th>Average STD (SW)</th>
<th>Median MSE (SW)</th>
<th>Median STD (SW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>13.2</td>
<td>24.0</td>
<td>15.4</td>
<td>30.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NN 1</td>
<td>26.1</td>
<td>46.3</td>
<td>20.8</td>
<td>42.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NN 2</td>
<td>30.7</td>
<td>56.0</td>
<td>28.9</td>
<td>55.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NN 3</td>
<td>53.0</td>
<td>82.3</td>
<td>47.1</td>
<td>90.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NN 4</td>
<td>67.9</td>
<td>111.3</td>
<td>59.6</td>
<td>111.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NN 5</td>
<td>67.3</td>
<td>104.6</td>
<td>85.0</td>
<td>139.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NN 6</td>
<td>114.2</td>
<td>175.5</td>
<td>76.6</td>
<td>144.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NN 7</td>
<td>118.8</td>
<td>178.5</td>
<td>132.8</td>
<td>229.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NN 8</td>
<td>158.0</td>
<td>237.9</td>
<td>170.6</td>
<td>319.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Similar to Time Period #1, regression models and neural network models with one to two hidden layer units have smaller MSE’s (higher performance) and smaller standard deviations as shown in Table 8. Neural networks with SW factor selection and one hidden unit achieved the
lowest average MSE (28.2 $\%^2$), but their average standard deviation is still higher (51.7 $\%^2$) compared to linear regression with SW factor selection (30.9 +/- 22.0 $\%^2$).

Table 9: Neural network and linear regression models' performance for Time Period #2.

<table>
<thead>
<tr>
<th>Units</th>
<th>Number of Funds Better Predicted By Specific Model Type</th>
<th>Number of Funds Better Predicted By Specific Model Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NN Regression</td>
<td>NN + SW</td>
</tr>
<tr>
<td>NN 1</td>
<td>152 141</td>
<td>213</td>
</tr>
<tr>
<td>NN 2</td>
<td>112 181</td>
<td>150</td>
</tr>
<tr>
<td>NN 3</td>
<td>38 255</td>
<td>78</td>
</tr>
<tr>
<td>NN 4</td>
<td>12 281</td>
<td>40</td>
</tr>
<tr>
<td>NN 5</td>
<td>4 289</td>
<td>22</td>
</tr>
<tr>
<td>NN 6</td>
<td>2 291</td>
<td>18</td>
</tr>
<tr>
<td>NN 7</td>
<td>2 291</td>
<td>10</td>
</tr>
<tr>
<td>NN 8</td>
<td>1 292</td>
<td>5</td>
</tr>
</tbody>
</table>

From Table 9, the 152 and 141 values from the first row and first two columns data illustrate that for the 293 funds, 152 of the funds had a lower MSE (better performance) with a neural network model with one hidden layer neuron compared to the regression model. This means that 141 of the funds had a lower MSE (better performance) with a linear regression model compared to the neural network model with one hidden layer neuron. Therefore, neural network models with one hidden layer neuron have better predictive performance compared to regression models. The neural network performance significantly improves when stepwise regression is used, as 213 of the 293 funds had a lower MSE (better performance) with a neural network with one hidden layer neuron compared to regression models with factor selection. Finally, the last two columns clearly illustrate that any model with stepwise factor selection has a lower MSE (better performance) compared to a model that does not have stepwise factor selection.

Table 8 also shows that with SW factor selection included, the MSE and standard deviation decreases across all models, or, performance of all models increases with SW factor selection.
For example, 213 out of the 293 funds achieved better modeling performance with a one hidden layer unit neural network that had SW risk factor selection compared to one that did not. Similarly, a linear regression model with SW risk factor selection achieved better performance with 225 out of 293 funds compared to a regression model without SW risk factor selection. Figure 11 also illustrates that regardless of the number of hidden layer units, the neural network that had assistance from SW factor selection consistently performed better than a network that included all 25 risk factors as inputs. The results indicate that funds that have their returns modeled using all 25 risk factors do not necessarily need their returns explained by all of these factors to achieve a similar fit. By lowering the number of inputs, the risk of over fitting due to excessive parameters in the model is lowered, which is likely why the SW factor selection reduced the average MSE and standard deviation in the models.
Figure 9: Mean-squared error and standard deviation comparisons across models for Time Period #2.

Figure 9 illustrates that a neural network with one hidden layer neuron and stepwise factor selection achieves the lowest MSE, or highest performance. The neural networks with at least four hidden layer neurons were not included in these figures because their error is significantly larger and as a result if we included them it would make it difficult to view the differences in MSE and standard deviation for the regression models and neural network models with 1-3 hidden layer neurons.
From Figure 10, 152 and 141 values from the first graph for NN 1 illustrate that for the 293 funds, 152 of the funds had a lower MSE (better performance) with a neural network model with one hidden layer neuron compared to the regression model. This means that 141 of the funds had a lower MSE (better performance) with a linear regression model compared to the neural network. Therefore, neural network models with one hidden layer neuron have better predictive performance compared to regression models. The regression models have better performance with networks with at least two hidden layer neurons. However, when stepwise regression is used 213 of the 293 funds had a lower MSE (better performance) with a neural
network with one hidden layer neuron. Neural networks have better performance (lower MSE) compared to regression models with up to two hidden layer neurons when factor selection is included. Nevertheless, regression models outperform the neural networks when at least three hidden layer neurons are used with stepwise factor selection.

![Figure 11: Stepwise factor selection performance for Time Period #2.](image)

Figure 11 show that neural networks with stepwise factor selection consistently outperform neural networks without factor selection. For example, a neural network with one hidden layer neuron and factor selection had better predictive performance for 213 out of the 293 funds compared to a network with one hidden layer neuron and without factor selection.

It is clear that as the number of hidden layer units increase, the MSE and standard deviation of the network significantly increases (performance decreases). The result may be counter-intuitive because, typically, as hidden layer units increase, a neural network should be able to fit the model with greater precision due to the introduction of further parameters. Again, similar to Time Period #1, since there are only limited number of data points available (N=144), the network is likely fitting itself to the data points, as opposed to the pattern represented by the
data points when extra parameters are present. As a result, out of sample performance significantly decreases.

The results from models fitted in Time Period #2 verify the findings from Time Period #1 with regards to which models produce the best performance, specifically neural networks with low hidden layer unit counts with SW factor selection.

The following section outlines the differences in model performances between the two study periods.

7.4 Out-of-Sample Modeling Performance Comparison Between Study Periods

This section discusses the difference in performance and standard deviation for comparable models that vary only in the two time period data sets used to fit the models. The comparison shows whether the volatile recent financial crisis can be reasonably predicted by a model that is formulated using data before the financial crisis. The performance of these models is compared to the first time period to assess their relative performance.

Figure 12 clearly illustrates that models constructed using data from Time Period #2 had significantly higher out-of-sample mean-squared-error compared to Time Period #1.
Figure 12: Comparison of the median and mean MSE’s for both time periods.
Figure 13: Comparison of the median and mean standard deviations for both time periods.

Figure 13 illustrates that models constructed using data from Time Period #2 had significantly higher out-of-sample standard deviation compared to Time Period #1.
As expected, model performance is not as strong during the relatively volatile recent financial crisis compared to model performance over a less volatile time period. Additionally, it is clear neither model is particularly better than the other at modeling extreme events as their out of sample validation data shows. Put another way, the current input dataset supplied to the models do not contain enough information for the models to effectively handle extreme financial crises as their out of sample validation set. Additional risk factors that can better explain this behavior, perhaps qualitative in nature (e.g. fear indices), should be included in models to see their effect on the proportional difference in performance. Notwithstanding all of this, it is not surprising that such a serious financial event as the recent recession is not predictable in this way. If this was possible, much pain and suffering, in a financial sense, could be avoided – nobody has been successful in this so far.
8 CONCLUSIONS

The process of modeling hedge fund returns has been an interesting subject ever since hedge funds have seen very significant growth from the past two decades. Due to their complex, unregulated, and often undisclosed trading strategies, along with a lack of historical performance data until just recently, this task has been difficult to accomplish.

This thesis proposes a new way to model hedge fund monthly performance using a neural network approach that is combined with a stepwise regression backwards elimination approach for risk factor selection. We examine the out-of-sample performance for neural network models with varying hidden layer units, with different sets of risk factors, and using two unique time periods. As well, the work compares this performance with linear regression models which use the same data.

The new neural network modeling approach delivers better out-of-sample performance compared to regression models using identical risk factors as inputs in both time periods. For example, out of 293 studied funds, a neural network with one hidden layer unit and SW risk factor selection achieved better performance with 216 and 213 of the 293 total funds for the first and second study period, respectively. However, this additional performance comes at a cost of higher variance, as neural networks models consistently had higher standard deviations compared to linear regression models.

Furthermore, stepwise regression backwards elimination risk factor selection consistently improved the performance for all model types across both time periods. For example, for linear regression, better performance was achieved for models that had SW risk factor selection for 227 and 225 out of the 293 total funds compared to models without SW factor selection for the first
and second time periods, respectively. For neural networks, a network with one hidden layer unit achieved better performance with SW risk factor selection for 235 and 213 out of the 293 total funds compared to neural network models without SW factor selection for the first and second time periods, respectively. This result indicates that not all risk factors have explanatory power on every fund, and therefore including all of them would only increase the in-sample performance of models by using the extra regression or neuron parameters to fit to the data points, as opposed to the pattern represented by the data.

Finally, out-of-sample model performance is significantly worse during the second time period compared to the first time period with comparable models. The second time period uses the recent financial crisis as an out-of-sample set to test the models, and compared to the first time period’s out-of-sample testing data, these models have greater mean squared errors and standard deviations in the range of 150%-1600%. These findings indicate that the relatively volatile financial crisis is difficult to model with the chosen risk factors based on historical hedge fund return data. As a result, it is important to understand that while a model may achieve excellent performance over a certain time frame, there is no guarantee that this performance will remain during significant economic upheavals.

8.1 Future Work

We raise additional areas of study that surround efforts to model hedge fund performance. In particular, factor selection optimization, adjusting neural network training parameters and an increasing volume of historical performance data offers different ways to improve neural network model performance.

The process of factor selection in this project, while consistent for all models, was likely not optimized. The utilized SW factor selection is not a globally optimal approach, and as a result
selected subsets of risk factors may not have achieved the best model performance compared to different subsets of risk factors. Therefore, more work should be done to combine a more robust factor selection process with the neural network models. For example, principal component analysis (PCA) is an effective factor selection technique that could be used to select the best subset of the 25 risk factors we studied.

As well, the proposed 25 factors in this thesis are not inclusive of all factors that have been shown to have an explanatory effect on hedge fund performance. A more specific approach to risk factor selection for specific hedge fund trading classifications is a subject of significant literature and, when combined with better predictive neural network models as shown in this work, could lead to potentially impressive modeling results.

Additionally, other than hidden layer units, neural network parameters were not adjusted to increase performance. Significant work can be done testing the effects of varying neuron transfer functions, training methods, and learning rates for the neural network models.

As well, as more historical performance data and unique funds become available, neural networks with greater numbers of hidden layer units should be able to achieve better model performance compared to the ideal neural networks with one hidden layer unit as proposed in this model. These models achieved the best performance likely due to the fact that not enough data was available to prevent overtraining with models with greater numbers of hidden layer units.
REFERENCES


