12-2015

Predicting Intraday Financial Market Dynamics Using Takens' Vectors; Incorporating Causality Testing and Machine Learning Techniques

Abubakar-Sadiq Bouda Abdulai
East Tennessee State University

Predicting Intraday Financial Market Dynamics Using Takens' Vectors: Incorporating Causality Testing and Machine Learning Techniques

---

A thesis

presented to

the faculty of the Department of Mathematics

East Tennessee State University

In partial fulfillment

of the requirements for the degree

Master of Science in Mathematical Sciences

---

by

Abubakar-Sadiq Bouda Abdulai

December 2015

Jeff Knisley, Ph.D., Chair

Anant Godbole, Ph.D.

Michele Joyner, Ph.D.

Keywords: Takens Vectors, Causality, Financial Market, Machine Learning.

ABSTRACT

Predicting Intraday Financial Market Dynamics using Takens' Vectors:

Incorporating Causality Testing and Machine Learning Techniques

by

Abubakar-Sadiq Bouda Abdulai

Traditional approaches to predicting financial market dynamics tend to be linear and stationary, whereas financial time series data is increasingly nonlinear and non-stationary. Lately, advances in dynamical systems theory have enabled the extraction of complex dynamics from time series data. These developments include theory of time delay embedding and phase space reconstruction of dynamical systems from a scalar time series. In this thesis, a time delay embedding approach for predicting intraday stock or stock index movement is developed. The approach combines methods of nonlinear time series analysis with those of causality testing, theory of dynamical systems and machine learning (artificial neural networks). The approach is then applied to the Standard and Poors Index, and the results from our method are compared to traditional methods applied to the same data set.

DEDICATION

To

My brother, *Inusah Abdulai*

and

his wife, *Anne E. Kye-eebo*,

who taught me good

ACKNOWLEDGMENTS

I thank my advisor, Dr. Jeff Knisley, for all the guidance, encouragement and advice that he provided since I became his student. I have been very lucky to have him as advisor - one who cared so much about the quality the work of his students. His timely response to my questions, and review of this work ensured its timely completion. To paraphrase, he said "it is not only our duty to teach you, but to make sure when you go out there you fit well". Thank you ensuring that, that came through. Thank you to Dr. Joyner for leading me to Dr. Godbole, and to Dr. Godbole for leading me to Dr. Knisley. That is a network that I will forever remember. I thank you all for the support all these years. I also thank all members of staff who worked in diverse ways to help me achieve this feat. A big thank you to Dr. Bob for working so hard to get me to ETSU.

I must express my profound gratitude to my brother, Mr. Inusah Abdulai and his wife, Ms. Anne E. Kye-eebo, to whom I dedicate this work, for all the love and support all these years. No formula can quantify my appreciation. If I have a little understanding of mathematics today, you sowed the seed, and ensured it growth.

Thank you to all my friends, who supported in diverse ways. I appreciate the prayers and best wishes you offered. Thank you so much everybody.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 GRANGER CAUSALITY

1.1 Introduction

"When the Lord created the world and people to live in it – an enterprise which, according to modern science, took a very long time – I could well imagine that He reasoned with Himself as follows : 'If I make everything predictable, these human beings, whom I have endowed with pretty good brains, will undoubtedly learn to predict everything, and they will thereupon have no motive to do anything at all, because they will recognize that the future is totally determined and cannot be influenced by any human action . On the other hand, if I make everything unpredictable, they will gradually discover that there is no rational basis for any decision whatsoever and, as in the first case, they will thereupon have no motive to do anything at all. Neither scheme would make sense. I must therefore create a mixture of the two. Let some things be predictable and let others be unpredictable. They will then, amongst many other things, have the very important task of finding out which is which.' "

From the book "Small is Beautiful: Economics as if people mattered"

by E. F. Schumacher

According to the efficient market hypothesis (EMH), there is no possibility of predicting future prices by analyzing past price information (Weak EMH). It further asserts that, not even by relying on new publicly available information would that provide that prediction advantage (semi-strong EMH), and that combined public and private information would not either (strong EMH) [8]. The reason for this is that EMH presumes that existing share prices incorporate and reflect all relevant information. However, rich evidence in the academic literature [55], and the corporate world casts doubt upon the unpredictability of the financial market. The evidence indicates that there may be complex and largely unexplored dynamics through which information is aggregated in markets which affects price processes [8, 9]. This suggests that extracting and aggregating that relevant information has the tendency to enhance predictions. One approach to achieving this, is to examine the causal relationships between factors and targets through causality testing.

To examine whether information in a lagged time series could enhance the prediction of another time series, Granger [2] came up with a test to assess if such lagged time series significantly explains another time series in a vector autoregressive regression model. Granger causality or what could be called statistical causality as developed by C. Granger is a statistical concept used to assess the influence of one variable on another so they can be incorporated into model building.

Causality as a property first appeared in a paper by Wiener [29] in which he stated that "For two simultaneously measured signals, if we can predict the first signal better by using the past information from the second one than by using the information without it, then we call the second signal causal to the first one."

The basic idea is that in order for a variable X to cause Y, the prediction of future values based on just the past values of Y has to be improved when information about the past of X is also included. In particular, if the variance of the prediction error of Y is reduced by including past measurements of X, then the second variable, X is said to, have a causal influence on the first variable Y in a statistical sense [7, 6].

In general, we say that a first signal causes a second signal if the second signal can be better predicted when the first signal is considered. It is then Granger causality if the notion of time is introduced and the first signal precedes the second one [3, 10].

The context in which Granger defined causality was that of linear autoregressive models of stochastic processes [5]. Evidence of complex dynamics in financial markets, however, makes the use of this traditional method yield poor results in financial applications. In the case of financial data, the lack of stationarity, nonlinear dependence between time series, most likely, the presence of noise, makes it inappropriate to apply an autoregressive model to the time series without first transforming it. Major relaxations in the assumptions underlying the application of an autoregressive model is noted to produce spurious results.

The goal of this thesis is to assess Granger causality in nonlinear financial time series and make predictions thereof. What follows next is a description of case studies illustrating the use of the traditional method in a variety of applications. We also provide definitions and background information on the subject of Granger causality. In chapter two, we describe what constitutes linear and nonlinear systems. We give reasons why linear models are sometimes not suitable for analyzing financial data. We conclude the chapter by introducing Takens theorem. In chapter three, we provide

details of the methodology to be applied. Chapter four is where we apply the method to the Standard and Poor's Index. In five, we give our remarks and conclusion.

1.2 Autoregressive Models

Autoregressive models are based on the idea that the current value of the series, x_t, can be explained as a function of p past values,

$$x_{t-1}, x_{t-2}, ..., x_{t-p},$$

where p determines the number of steps into the past needed to forecast the current value.

Definition 1.1 *An autoregressive model of order p, abbreviated AR(p) is of the form*

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + ... + \phi_p x_{t-p} + \omega_t, \tag{1}$$

where x_t is stationary, and ϕ_1, ϕ_2, ..., ϕ_p are constants ($\phi_p \neq 0$) and ω_t is a sequence of independent (or uncorrelated) random variables with mean 0 and variance σ^2.

The AR model is one of a group of linear prediction formulas that attempt to predict an output of a system based on the previous outputs [6, 11]. In an AR model, the variance of the prediction error is used to test for improvement in model prediction [7].

1.3 White Noise Residuals

A major building block for time series models is the white noise process, which we denote ω_t. A stationary time series ω_t is said to be white noise if Corr(ω_t, ω_s) =

0 for all $t \neq s$. In the least general case,

$$\omega_t \sim \text{ i.i.d } N(0, \sigma^2) \tag{2}$$

A major assumption about ω_t is that

$$E(\omega_t) = E(\omega_t|\omega_{t-1}, \omega_{t-2}, ...) = 0. \tag{3}$$

This property indicates the absence of any serial correlation or predictability.

1.4 Granger Causality Method

The primary method for inferring causality between two time series as developed by Granger uses the given time series to determine whether one predicts, or causes, the other.

If we consider the processes $x(t)$ and $y(t)$, then we observe first that each process admits an autoregressive representation:

$$x_t = \alpha_1 + \sum_{j=1}^{p} \phi_1 x_{t-j} + \omega_t, Var(\omega_t) = \Sigma_1 \tag{4}$$

$$y_t = \beta_1 + \sum_{j=1}^{p} \lambda_1 y_{t-j} + \eta_t, Var(\eta_t) = \Gamma_1 \tag{5}$$

Jointly, they are represented as

$$x_t = \mu_1 + \sum_{j=1}^{p} \phi_1 x_{t-j} + \sum_{j=1}^{p} \lambda_1 y_{t-j} + \omega_t' \tag{6}$$

$$y_t = \mu_2 + \sum_{j=1}^{p} \lambda_1 y_{t-j} + \sum_{j=1}^{p} \phi_1 x_{t-j} + \eta_t' \tag{7}$$

where the noise terms are uncorrelated over time with covariance matrix

$$\Sigma = \begin{bmatrix} \Sigma_2 & \Upsilon_2 \\ \Upsilon_2 & \Gamma_2 \end{bmatrix} \tag{8}$$

The entries are defined as $\Sigma_2 = var(\omega_t')$, $\Gamma_2 = var(\eta_t')$, $\Upsilon_2 = cov(\omega_t', \eta_t')$.

1.5 The F-Test

Consider (4) and (6) from Section 1.4. The value of Σ_1 measures the accuracy of the autoregressive prediction of x_t based on its lagged values, whereas the value of Σ_2 represents the accuracy of predicting the present value of x_t based on lagged values of both x_t and y_t. According to Wiener [29] and Granger [2], if Σ_2 is less than Σ_1 in some suitable statistical sense, then y_t is said to have a causal influence on x_t.

Inferring from (6) and (7) this test is generally carried out by testing the following null hypotheses separately:

$$H_{01} : \beta_1 = \beta_2 = ... = \beta_p = 0$$

$$H_{02} : \lambda_1 = \lambda_2 = ... = \lambda_p = 0$$

From testing these hypotheses, we have four possible results:

1. If both Hypotheses H_{01} and H_{02} are accepted, there is no causal relationship between x_t and y_t.

2. Hypothesis H_{01} is accepted but Hypothesis H_{02} is rejected, then there exists a linear causality running unidirectionally from x_t to y_t.

3. If Hypothesis H_{01} is rejected but Hypothesis H_{02} is accepted, then there exists a linear causality running unidirectionally from y_t and x_t.

4. If both Hypotheses H_{01} and H_{02} are rejected, then there exist feedback linear causal relationships between x_t and y_t).

There are a number of statistics that could be used to test the above hypotheses, with the most common being the standard reduced-versus-full-model F-test.

To test the hypothesis $H_{01} : \beta_1 = ... = \beta_p = 0$ in (6), the sum of squares of the residuals from both the full regression, SSR_F , and the restricted regression, SSR_R, are computed in 6, as well as the F statistic given by

$$F = \frac{(SSR_R - SSR_F)/p}{SSR_F/(n - 2p - 1)} \tag{9}$$

where p is the optimal number of lag terms of y_t in the regression equation on x_t and n is the number of observations. For a given significance level α, we reject the null hypothesis H_{10} if F exceeds the critical value $F_{\alpha(p,n-2p-1)}$ [13].

1.6 Case Studies

Traditional Granger causality has been applied in a variety of settings. Its application using VAR models has yielded quite satisfying results. For instance, Walter and Mark [47] used the method to investigate the causal relationship between chicken and eggs. The examined annual time series of US egg production and chicken population from 1930 to 1983. The study included chickens that are capable of "causing eggs" i.e chickens that lay or fertilize eggs. The eggs also included those that can be fertilized. They concluded that eggs *Granger-cause* chicken.

Daniel and Hunter [31], also explored the association between trade and innovation. There has been little explicit study of the direction of causality between these

two variables. Their study used all patents granted in the U.S. between 1987 and 1999, assigned them to probable industries of origin and sectors of use, then tested Granger-causality with trade flows in those same economic sectors. They concluded that causality at the industry level between these variables is more complicated. They found the relationship running in both directions from imports (or exports) to and from innovation.

Not only that, Pasquale Foresti [33], also focused his study on the relationship between stock prices and economic growth. He sought to assess whether there is any potential predictability power of one indicator for the other. He used quarterly time series data on the S&P 500 and GDP that run up to year 2005. The conclusion drawn is that stock market prices can be used to predict growth, but the reverse is not true.

Using data on inflation that spans the time period from the first quarter of 1960 to the third quarter of 1999, Hess and Schweitzer [42] investigated the causal relationship between wage inflation and price inflation. They found little support for the view that higher wages cause higher prices. Rather, the authors found more evidence that higher prices lead to wage growth.

Erdila and Yetkinerb [32] also investigate the Granger-causality relationship between real per capita GDP and real per capita health care expenditure. They do so by employing large macro panel data set with a VAR representation. Their results indicate that the dominant type of Granger-causality is bidirectional. In instances that they found one-way causality, the pattern is not homogenous: The analyses show that one-way causality generally runs from income to health in low- and middle-income countries whereas the reverse holds for high-income countries.

2 GRANGER CAUSALITY AND TAKENS THEOREM

2.1 Why (Linear) Granger Causality Fails

As earlier noted, Granger's original formulation of causality was developed to assess causality in a linear framework, and it is very robust and efficient at doing so. Granger's approach is most likely the best method for assessing Granger causality for data with linear dependence. The measure also works well in nonlinear cases in which the data has a strong linear component. The method has been particularly successful in fields such as neuroscience, climatology and physiology [5, 40]. Linear approximations have been found to work well in many of the applications in neuroscience, especially those involving large scale interactions [4, 5].

The method has been particularly successful in the above mentioned fields because the data from these fields are often characterized by linear dependence and gaussianity [3],

Scheinkman and LeBaron [56] and others [57, 58] find evidence of nonlinear dependence in stock market returns [13]. There are also a number of evidences about nonlinear causal relationships between variables in finance and economics. For example, Hsieh [59] notices that the nonlinear nature of stock price movements is motivated by asset behaviour that follows nonlinear models [13]. Hiemstra and Jones [60] also found evidence of some nonlinear interaction between trading volume and stock returns.

Thus financial data have different characteristics than data most commonly analyzed in fields such as the physical sciences. The key assumptions of linear dependence

or Gaussian distributions, which are often found reasonable to make in other scientific disciplines, are commonly thought to be inappropriate if directly applied to financial time series.

Researchers further point out that stationarity usually does not apply to this kind of data. So, as financial data does not always exhibit stationarity, linearity or Gaussianity, care should be taken in the choice of models used to analyze such data.

2.2 Linear versus Nonlinear Systems

The difference between linear and nonlinear systems, is that, the sum of any two different solutions of a linear system is also a solution to the given linear system; this is referred to as the superposition principle. In contrast to that, solutions to a nonlinear system cannot be added to form another solution. There is therefore no general analytic approach for solving typical nonlinear equations.

2.3 Linear versus Nonlinear Time Series

A time series is a collection of quantitative observations that are evenly spaced in time and measured successively [13]. Examples of time series include measure of a persons heart rate at regular time intervals, periodic recording of body temperature, weekly closing price of a stock, annual rainfall data, and annual death figures.

Definition 2.1 *A given time series is said to be linear if it can be expressed as*

$$x_t = \mu + \sum_{j=-\infty}^{\infty} \phi_j \omega_{t-j} \tag{10}$$

where μ is is the mean of x_t and $\{\omega_t\}$ are iid random variables with finite variance.

Linear time series models are designed to model the covariance structure in a given time series. There are two major groups of linear time series models - the autoregressive (defined in section 1.2) and the moving average models (MA).

A times series $\{y_t\}$ is said to have an MA(q) representation if it satisfies

$$y_t = \sum_{j=0}^{q} \phi_j \omega_{t-j} \tag{11}$$

Any stochastic process that does not satisfy (3) is said to be nonlinear. Mathematically, a purely stochastic time series model for x_t is a function of an *iid* sequence consisting of the current and past shocks - that is,

$$x_t = f(\omega_t, \omega_{t-1}, ...). \tag{12}$$

The linear model in 3 says that $f(\cdot)$ is a linear function of its arguments. Any nonlinearity in $f(\cdot)$ results in a nonlinear model [61]. The general nonlinear model in (12) is usually not directly applicable because it contains so many parameters.

2.4 Stationary and Non-Stationary Time Series

A stationary process is one whose statistical properties do not change over time [75]. In its most strict sense, a stationary process is one where given t_1, ...t_n the joint distribution of x_{t_1}, ..., x_{t_n} is the same as the joint distribution of $x_{t_1+\tau}$, ..., $x_{t_n+\tau}$ for all n and τ. This means that all moments of the process remain the same across time. That, the joint distribution of (x_t, x_s) is the same as for (x_{t+r}, x_{s+r}) and hence does depend only on the distance between s and t. But this definition is usually relaxed since it does not fit any model of nature. Therefore, there is what is referred to as weak stationarity, where the mean and variance of a stochastic process are constant

and do not depend on t. The autocovariance between x_t and $x_{t+\tau}$ also depends only on the lag τ. Basically, where the mean and variance of a time series do not remain constant over time, we say it is non-stationary.

2.5 Takens Theorem and Phase Space Reconstruction

The state of a deterministic dynamical system at a given time t_0 is described by a point $\mathbf{x}$ lying in $\mathbb{R}^d$ and the time evolution of such system is given by the map $\mathbf{x}_{t_0} \to \mathbf{x}_t$, which gives the state of the system at time t given that it was initially in state $\mathbf{x}_{t_0}$ at time t_0. This map is often obtained as a solution of some differential equation.

In many instances, however, we are not able to access the state $\mathbf{x}$, likewise the mapping $\mathbf{x}_{t_0} \to \mathbf{x}_t$. All we have is some function $\psi(\mathbf{x})$ of the state, where $\psi : \mathbb{R}^d \to \mathbb{R}$ is some measurement function. It corresponds to measuring some observable property of the system such as position or temperature. Let the evolution of this quantity be given by the time series $\psi(\mathbf{x}_t)$ which we observe in discrete time intervals in practice. Assume these intervals are equal so we observe the discrete time series $\psi_n = \psi(\mathbf{x}_{n\tau})$ for $n \in \mathbb{Z}^+$ where τ is some sampling interval.

As an example, we consider the Henon map. The henon map is a simple two dimensional mapping exhibiting chaotic behaviour. It has the ability to capture the stretching and folding dynamics of chaotic systems. It further provides a way to conduct more detailed exploration of the chaotic dynamics of a system [73].

It is given by

$$\begin{cases} x_{n+1} = 1 - a(x_n)^2 + y_n \\ \\ y_{n+1} = bx_n \end{cases} \tag{13}$$

where a and b are parameters with a standard choice being $a = 1.4$ and $b = 0.3$. The parameter a controls the amount of stretching and the parameter b controls the thickness of folding.

Notice from above that the time evolution of x_n depends on both x and y. Thus to generate figure 1(a) we have to compute x_n and y_n for n = 1, ..., 1000. Generally, as described above, the state of $\mathbf{x}$ will lie in $\mathbb{R}^d$ and we will need d coordinates to determine the future behaviour of the system. On the other hand, the observed time series ψ_n is only one dimensional.

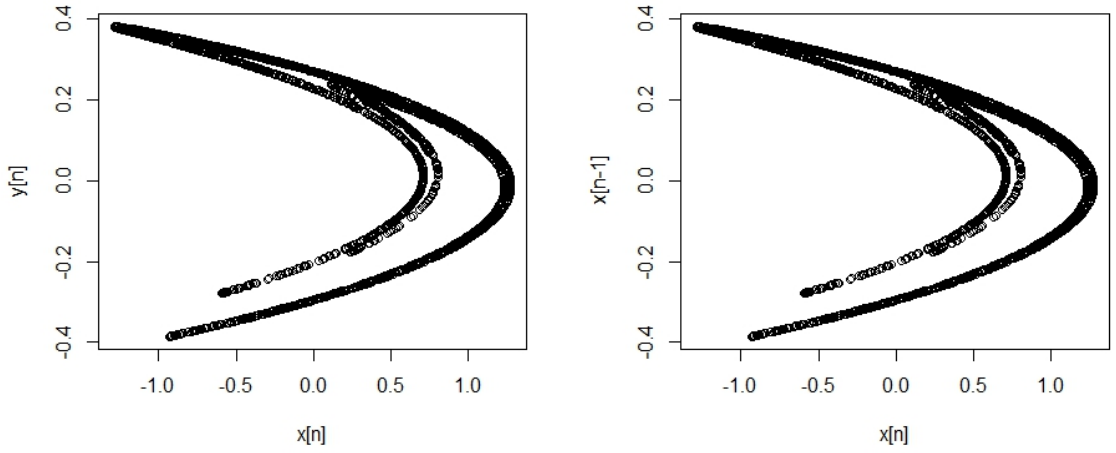
It may seem that ψ_n contains relatively little information about the behaviour of $\mathbf{x}_t$ and that the fact that ψ_n was generated by a deterministic system is of little value when the state $\mathbf{x}$ and the dynamics $\mathbf{x}_{t_0} \to \mathbf{x}_t$ are unknown to us.

Remarkably, this intuition turns out to be false due to a powerful result known as the Takens embedding theorem [1, 25]. In this theorem, Takens proved that for typical measurement function ψ, and typical dynamics $\mathbf{x}_{t_0} \to \mathbf{x}_t$, it is possible to reconstruct the state $\mathbf{x}$ and the map $\mathbf{x}_{(n-1)\tau}) \to \mathbf{x}_{n\tau}$ just from knowledge of the observed time series ψ_n as shown in figure 1(b) below.

Theorem 2.2 (Takens' Theorem) *There exists $d_E > 2d + 1$ and τ such that*

$$[x(t), x(t-\tau), x(t-2\tau), ..., x_{2d\tau}]$$

is an embedding of M into d_E-dimensional Euclidean space.



(a) *Henon Map using x and y*

(b) *Henon Map using Takens Vectors*

Figure 1: The Henon Map

Figure 1(a) above shows 3000 points of the attractor of the Henon map, plotted in the state space given by (x_n, y_n). Figure 1(b) by contrast, shows the reconstructed attractor in (x_n, x_{n-1}) space, using the Takens procedure, detailed in the next chapter. We emphasize that figure 1(b) was plotted using only the time series x_n with no knowledge of y_n. The figures are the same.

Takens' theorem thus gives the conditions under which a chaotic dynamical system can be reconstructed from a sequence of observations of the state of a dynamical system [16, 17]. Whitney showed that a d-dimensional manifold M can be mapped into a $d_E > 2d+1$ dimensional Euclidean space such that M is parametrized (locally) by an d_E dimensional vector-valued function [1]. This is known as an embedding. Takens used Whitney's theorem to show that a manifold topologically equivalent to

an embedding of M can be constructed from a single time series x_t measured from M.

According to Floris Takens, the attractor of a system under investigation can be reconstructed from a time series for one of the coordinates of the original phase space of the system. Let us denote this time series by x_t. Starting from x_t we can construct a new dynamical system of arbitrary dimension d taking as the vector describing the position of the representative point on the attractor in the state space of the system constructed, a d-dimensional vector

$$\mathbf{y_t} = (x_t, x_{t-\tau}, ..., x_{n-(d-1)\tau})$$

where τ is the embedding delay [27].

Takens' theorem tells us that if we take n consecutive amplitude values of a given time series x_t, and construct d_E-vectors from it (where d_E is greater than twice the dimension of M), then if ψ and Y are generic, these vectors which lie on a subset of $\mathbb{R}^{d_E}$ is an embedding of M [23].

For almost any time series x_t (which must be generic) and almost any time delay τ, the attractor of the d-dimensional dynamical system constructed will be equivalent topologically to the original system's attractor if $d_E \geq 2d+1$, where d is the dimension of the original attractor [27]. That is to say that the reconstruction preserves the properties of the dynamical system that do not change under smooth coordinate changes, but the geometric shape of structures in phase space may not be preserved [17].

A number of linear and nonlinear models are then applied to predict the dynamics of the system via the embedding vectors of one of the processes [34, 41, 49].

3 METHODOLOGY

In this thesis, we use Granger's method to assess causality in a sufficiently long time series to find a section(s) of it that has (have) more predictive power for forecasting an index's intraday movement (up = 1 or down = 0). We then apply Takens' theorem to build Takens vectors which we input together into a neural network, together with the next day's stock index movement. These serve as features for training and prediction. We compare the results from this procedure to the usual predictive modeling approach where features are normalized and fed into the machine learning algorithm for training and prediction.

3.1 Data Pre-Processing and Selection

Data pre-processing is the process of analyzing and transforming the input and output variables to highlight the important relationships among factors [18]. It flattens the distribution of variables and assists neural networks in learning relevant patterns.

Data normalization is known to be one of the major pre-processing applications. It allows for a rescaling of a raw data set into a format that is easily "understood" by the algorithm to be applied. It is known to eliminate the effects of certain influences [71]. If the data is noisy and unreliable, then knowledge discovery during the training phase will be very difficult [19]. It is also known to speed up training time by starting the training process for each feature within the same scale [20]. Some of the more common data normalization techniques are Min-Max, Sigmoidal and Z-score normalizations.

Min-Max Normalization: Min-Max Normalization rescales a variable from one

range of values to a new range of values. Usually, these are features, and they are rescaled to lie within the range [-1,1]. This method of normalization transforms a value A to another value B, to fit in the range $[C, D]$. It is given by the formula

$$B = \left(\frac{A - min(A)}{max(A) - min(A)} \right) \cdot (D - C)) + C$$

This method has the advantage of preserving all relationships in the data.

Sigmoidal Normalization: The sigmoid normalization function is used to non-linearly scale the samples in the range [-1,1]. The original data is first centered about the mean, and then mapped to the almost linear region of the sigmoid. This method is especially appropriate for a noisy data set with outlying values present [70].

Z-score Normalization: This technique uses the mean and standard deviation of each variable in a given training data set to normalize such variable. The transformation is given in the equation

$$x'_i = \frac{x_i - \mu_i(x)}{\sigma_i(x)}.$$

The major advantage of using this statistical norm technique is that it minimizes the effects of outliers in the data.

Normalization ensures that we achieve stationarity and linearity in a data set, which paves the way for the application of Granger causality to select the best set from the long time series for processing. After selecting a section(s) of the long time series that will yield good predictions, we construct the Takens vectors using the section(s) of the original time series that correspond to the selected normalized selection(s).

3.2 Time Delay Embedding and Takens' Vectors

Time delay embedding of a univariate time series involves the reconstruction of a series of vectors $\mathbf{V}_n$, $n = 1, ..., N$ from a time series x_t, $t = 1, ..., T$. The technique, described by Floris Takens, called the method of delays, is quite simple and can be applied to essentially any time series whatsoever [23]. This is done by selecting a few consecutive amplitude values of x_t as the coordinates of the vector. Given a sufficiently long time series, all of the state space will be well sampled, and the information in our reconstructed space will be correspondingly complete and we can learn from it things about the system itself.

Let x_t, $t = 1, ..., T$, be a univariate time series generated by a dynamical system. We assume that x_t is a projection of dynamics operating in high-dimensional state space. If the dynamics is deterministic, we can try to predict the time series by reconstructing the state space. The values of τ and d_E are used to transform the univariate time series into Takens' vectors (also called phase space or delay vectors) $\mathbf{V}_n$ stacked as

$$\mathbf{V}_n = [V_1, V_2, ..., V_N] = \begin{bmatrix} x_1 & x_{1+\tau} & \dots & x_{1+(d_E-1)\tau} \\ x_2 & x_{2+\tau} & \dots & x_{2+(d_E-1)\tau} \\ \vdots & \vdots & \vdots & \vdots \\ x_N & x_{N+\tau} & \dots & x_{N+(d_E-1)\tau} \end{bmatrix} \tag{14}$$

The embedding dimension, d_E, represents the dimension of the state space in which to view the dynamics of the underlying system whiles the time delay, τ represents the time interval between the successively sampled observations used in constructing the d_E-dimensional embedding vectors. If the embedding dimension is "big enough", such that $\mathbf{V}_t$ unambiguously describes the state of the system at time t then there

generically exists an equation for points on the attractor, which is of the form

$$x_{t+n} = f(\mathbf{V}_t) \tag{15}$$

where $f : R^d \to R$, is a function that allows us to predict future values of the time series x_t, given past values, where n is the prediction step. Takens showed that given that $d_E \geq 2d + 1$, then such continuous function f can be found within the bound.

3.3 Embedding Parameters

In reconstructing the phase space, the main problem is how to select the τ and d_E in a way that guarantees the validity of the above function. Thus, a proper phase space reconstruction from a time series that will yield good predictions requires a careful selection of these two key parameters.

Based on Takens' theorem, when $d_E \geq 2d + 1$, an attractor embedding can be obtained. But this is only a condition which serves as a guide for selecting the value of the embedding dimension and time lag. There are a number of criteria for estimating these parameters.

3.4 Determining Time Delay and Embedding Dimension

In real applications, the proper choice of the time delay τ and the calculation of an embedding dimension, d_E, are fundamental for starting to analyze data [36, 39].

It is very important to select a suitable pair of embedding dimension d_E and time delay τ when constructing the Takens vectors since the choice of τ and d_E is directly related to the accuracy of the invariables of the described characteristics of the strange

attractors in state space reconstruction [38]. A "brute force" method would simply be to search over a number of combinations until an attractor comes into focus. However, the typical problems of this method are that, it is time-consuming, unsuited for a short time series, and sensitive to noise. Fortunately, the literature suggests various diagnostic procedures for determining the time delay and the minimum embedding dimension, all of which are heuristic; they yield optimal results for selected systems only, and perform just as average for others.

3.4.1 Time delay

The first step in state space reconstruction is to choose an optimum delay parameter. Determining a good embedding dimension, d_E depends on a judicious choice of time delay, τ. There are many methods for choosing the time delay τ from a scalar time series, the major ones being the autocorrelation function and mutual information methods.

A basic prescribed criteria for estimating a proper time lag is based on the following reasoning: if the time lag used to build the Takens' vectors is too small, the coordinates will be too highly temporally correlated and the embedding will tend to cluster around the bisectrix (diagonal) of the embedding space [68]. In such case, there is almost no difference between the elements of the delay vectors; this is called redundancy in [30] and [48]. A very large value, on the other hand, may cause the different co-ordinates to be almost uncorrelated [68]. In this case, the reconstructed attractor may become very complicated even if the underlying "true" attractor is simple [39]: this is referred to as irrelevance. Thus our goal is to find a lag which falls

in-between these scenarios. And there are a wide variety of methods for estimating an appropriate time lag. The autocorrelation function can be used to estimate an appropriate time lag of a given time series [68]. And there are a wide variety of methods for estimating an appropriate time lag using the autocorrelation procedure:

First Zero Method: One way of doing this, is to choose the first zero of the autocorrelation function. The autocorrelation function of the time series plots the correlation of a time series with itself as a function of time shift. Under this method, we select the time lag where the autocorrelation function decays to zero. Mees et al [62] suggest the use the value of τ for which the autocorrelation function

$$C(\tau) = \sum_{n} [x_n - \bar{x}][x_{n+\tau} - \bar{x}] \tag{16}$$

first passes through zero. This is equivalent to requiring linear independence.

First.e.decay Method: Rosenstein et al [63] proposed choosing a time lag needed for the value of the autocorrelation function to drop to $\frac{1}{e}$ times of the initial value as the time delay. By choosing the lag equal to the time after which the autocorrelation function has dropped to $\frac{1}{e}$ of its initial value, consecutive amplitude values are neither too much nor too little correlated [52].

However, a reasonable objection to the above autocorrelation based procedure is that they are based on linear statistics, not taking into account nonlinear dynamical correlations within a time series [22]; this may not seem a logical choice for determining nonlinear parameters. Therefore, it is usually advocated that one looks for the first minimum of the time delayed mutual information [64].

First Minimum of the average mutual information: This is an estimation method based on the time-delayed mutual information (TDMI), which can be thought of as a nonlinear generalization of the autocorrelation function. This method, proposed by Fraser and Swinney [64], involves plotting the time-delayed average mutual information versus the delay and identifying the first minimum in the resulting curve [35]. The rationale behind the approach is to introduce the needed statistical independence between successive lagged values.

For a given time series, x_t, the TDMI, $I(\tau)$, is a measure of the information about $x_{t+\tau}$ contained in x_t. $I(\tau)$ is therefore a probabilistic measure of the extent to which $x_{t+\tau}$ is related to x_t at a given τ. The first minimum of $I(\tau)$ marks the time lag where $x_{t+\tau}$ adds maximal information to the knowledge we have from x_t, or where the redundancy is least [22, 21, 36]. The τ is returned as an estimate of the proper time lag for a delay embedding of the given time series.

For a discrete time series $I(\tau)$ is estimated as

$$I(\tau) = \sum_{t,t+T} P(x_t, x_{t+\tau}) \log_2 \left[\frac{P(x_t, x_{t+\tau})}{P(x_t)P(x_{t+\tau})} \right] \tag{17}$$

where $P(x_t)$ is the individual probability and $P(x_t, x_{t+\tau})$ is the joint probability density.

3.4.2 Embedding Dimension

Just like the time delay for embedding, an optimal embedding dimension is an important parameter for the phase space reconstruction. Generally, there are three basic methods used in the literature, which include computing some invariant on the

attractor, using singular value decomposition, or the method of false nearest neighbors. The embedding dimension is mostly chosen using the false nearest neighbors method [36]. This method measures the proportion of close neighboring points in a given dimension whose position does not change in the next highest dimension [36]. The minimum embedding dimension capable of containing the reconstructed attractor is that for which the percentage of false nearest neighbors drops to zero for a given tolerance level [36, 28]. However, the false nearest neighbors method, and the two other methods mentioned above contain some subjective parameters or a subjective judgement is required to choose the embedding dimension.

To deal with the problem making an arbitrary selection of the level of tolerance for estimating the embedding dimension, Cao [26] modified the method of false neighbors into a method known as the averaged false neighbors, which does not contain any subjective parameter. All that is required is a properly estimated time-delay. The method takes the estimated time-delay as given, and estimates the embedding dimension that minimizes the prediction error [66].

For a given dimension d, we can get a series of delay vectors V_t defined in equation (14). For each $\mathbf{V}_t$ we can find its nearest neighbor $\mathbf{V}_{t'}$, i. e.,

$$\mathbf{V}_{t'} = argmin\{||\mathbf{V}_t - \mathbf{V}_j|| : j = (d-1)\tau + 1, ..., N, j \neq t\} \tag{18}$$

where the norm

$$||\mathbf{V}_t - \mathbf{V}_j|| = \left(\sum_{i=0}^{d-1} (x_{t-i\tau} - x_{j-i\tau})^2 \right)^{\frac{1}{2}}. \tag{19}$$

Then we define

$$E(d) = \frac{1}{N - J_0} \sum_{t=J_0}^{N-1} |(x_{t+1} - x_{t'+1})|, J_0 = (d-1)\tau + 1, \tag{20}$$

where $E(d)$ is the average absolute prediction error of a zero-order approximation predictor for a given d [66]. Note that a zero-order predictor f is $x_{t+n} = f(\mathbf{V}_t)$ and $x_{t+n} = x_{t'+n}$, where t' is an integer such that $\mathbf{V}_{t'}$ is the nearest neighbor of $\mathbf{V}_t$. Furthermore, note that the N in (20) represents only the number of available data points for fitting, which does not include the data points for out-of-sample forecasting. To choose the embedding dimension, d_E, we simply minimize the E, That is

$$d_E = argmin\{E(d) : 1 \leq d \leq D_{max}\} \tag{21}$$

where D_{max} is the maximum dimension with which we would like to search the minimum value of $E(d)$. More details of the Cao method can be found in [26].

3.5 Artificial Neural Networks (ANNs)

Having obtained statistically relevant variables that could improve the prediction of the target variable, we feed this data into a neural network for training and prediction.

An artificial neural network is a relatively crude network of "neurons" based on the neural structure of the brain [43, 45]. They process records one at a time, figuring out patterns or relationships in a given data set. They compare their classification of the record (which at the outset is arbitrary) with the known actual classification of the record [43]. The errors from this comparison are re-fed into the neural network for modification in subsequent iterations.

The neuron generally has a set of input values X_i with proportional weights W_{ij} and a transfer function Σ that processes the weights and maps the results to an activation function [43].

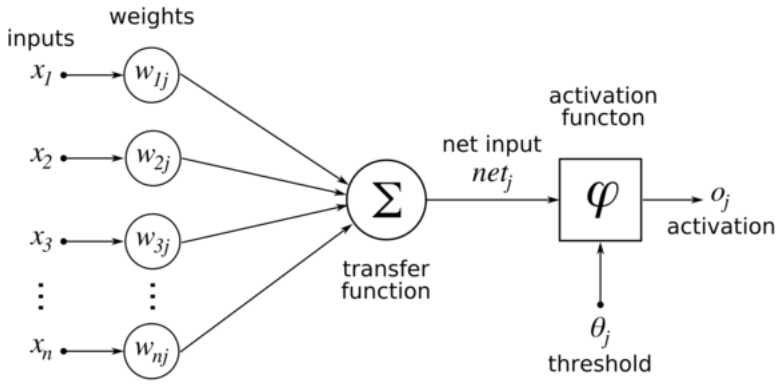


Figure 2: An Artificial Neuron Layout

The neurons in an artificial neural network are basically organized into three layers: input, hidden and output layers. Observed patterns in the training set are presented to the hidden layer through the input layer. The hidden layer, made up of a number of interconnected 'nodes' contains an 'activation function' for processing the weighted records. The hidden layer is in turn connected to the output layer, which outputs the result as predicted by the algorithm.

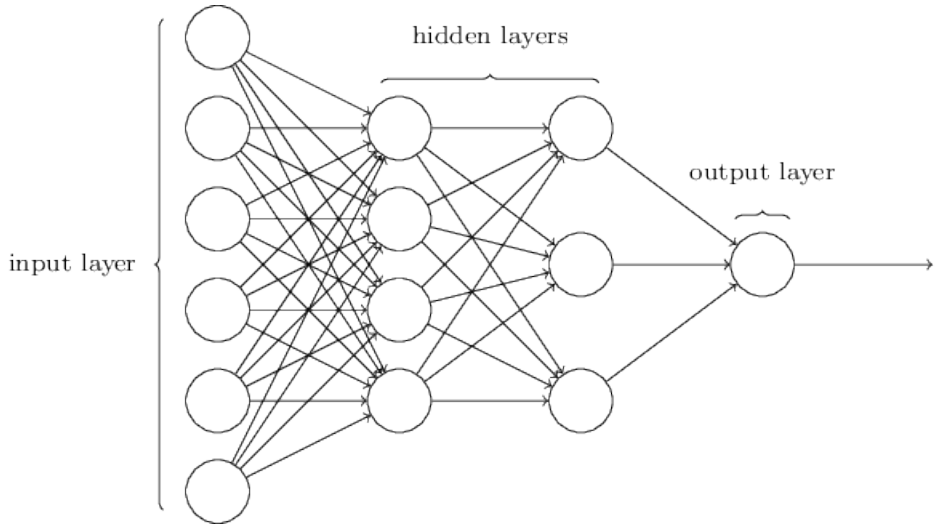


Figure 3: A Multiple Hidden Layer Network

3.5.1 Training an Artificial Neural Network

A neural network must be trained on some input data. During the training phase in, say, a typical classification problem, where the correct class for each record is known, and the output nodes can are assigned "correct" values, it is possible to compare the network's calculated values to the "correct" values, and calculate an error term for each node. The network then uses the error terms to adjust the weights in the hidden layers so that during the next iteration the output values will be closer to the "correct" values.

3.5.2 ANNs for Stock Market Prediction

The ability of neural networks to discover nonlinear relationships in input data makes them ideal for modeling nonlinear dynamic systems such as the stock market [76]. They still remain the most prominent technique for predicting the stock market. Some of the major advantages of neural networks are their ability to tolerate noisy data, as well as their capacity to correctly classify patterns in a data set on which they have not been trained [77].

The most common form of ANN that is used for stock market prediction is the feed forward network which utilizes the backward propagation of errors algorithm to update the network weights. These networks are commonly referred to as backpropagation networks. Currently, the backpropagation architecture is the most popular for complex networks [67].

3.5.3 Model Evaluation

An integral part of modeling is the improvement in predictive accuracy. When a model is built, steps have to be taken to ensure that its application yields very accurate results. One key method that allows for a careful and comprehensive analysis of all possible error severities is the Receiver Operating Characteristic (ROC) curve.

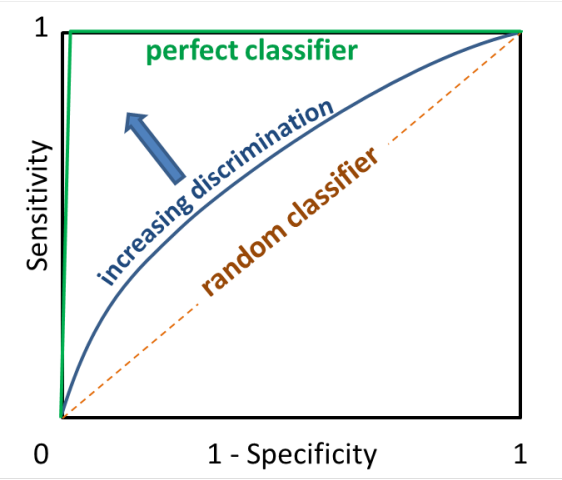


Figure 4: ROC Curve Illustration

The ROC curve plots the true positive rate (called sensitivity) of a model on the vertical axis against false positive rate (1–specificity) on the horizontal axis. An ROC curve always starts at (0,0) and ends at (1,1), and it is always bounded above by 1. A 45 degree diagonal ROC corresponds to a random (aka, "coin toss") classifier, whereas the ROC for a perfect classifier ideally is equal to 1 for all non-zero "false alarm" rates. If the training set is relatively balanced (about same number of positives and negatives), then the area under the curve is the probability that an observation chosen at random will be correctly classified [10].

4 AN APPLICATION

In this section of the thesis we apply the Takens algorithm to predict the movement of the US stock market. Specifically, we will predict the S&P index's intraday movement (up or down) using data from Yahoo! finance. We first apply the Takens procedure to some data set. Next, we apply a normalization procedure, and make a comparison thereof.

The Takens procedure incorporates three key applications:

1. Assessing causality to obtain a more predictive section of the stock data that influences the stock's movement the next day,

2. Extracting this section and constructing Takens' vectors, and

3. Feeding the vectors (which serve as factors) into a neural network for training and prediction.

4.1 *R* Packages Utilized

We make use of a number of packages built for the *R* programming language. The major packages used for data processing and analysis are the *lmtest*, *fNonlinear*, and *nonlinearTseries* packages.

The *lmtest* package is used for Granger causality testing whiles the *fNonlinear* package is used for estimating the time lag for constructing the Takens vectors. The estimation of the embedding dimension and the construction of the Takens' vectors is achieved using the *nonlinearTseries* package.

The *lmtest* package contains the generic function *grangertest* which performs tests for Granger causality in bivariate series. The test is a Wald test comparing the unrestricted model and the restricted model.

The *fNonlinear* package is a collection of functions that allow us to investigate the chaotic behavior of time series processes. For instance the *mutualplot* function graphs the the mutual information index of a given time series for a specified number of lags.

The *estimateEmbeddingDim* function is used to estimate the minimum embedding dimension using the algorithm proposed by L. Cao [26], whiles the *buildTakens* function builds the Takens vectors from the given time series.

4.2 Data Collection and Pre-Processing

The S&P 500 is one of the most commonly used benchmarks for the overall U.S. stock market. The index is computed using data from 500 leading companies. The total market capitalization held by these 500 companies is approximately 80%. These stocks are selected based on factors that include market size, liquidity and industry grouping, among other factors [44]. There is currently over US$ 7.8 trillion benchmarked to the index, with index assets comprising approximately US$ 2.2 trillion of this total [44].

We make use of data from the Yahoo! Finance website. Other sources where same data could be obtained are Google Finance and Morning Star websites. The data set comprises of the daily *open*, *low*, *high*, *close*, *volume*, and *adjusted close* prices for the S&P 500 index. The daily S&P 500 index consecutive values for the last 5 years (January 1, 2010 - December 31, 2014) is used in this study.

4.3 Results and Comparison

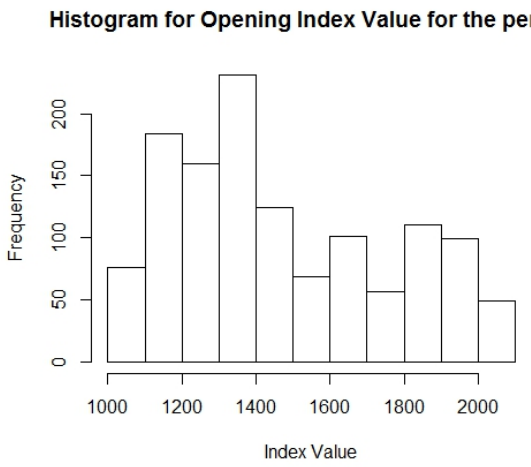
We first normalize the data to put all factors on an equal footing. As earlier mentioned, this:

- enables data mining algorithms to be applied easily
- presents the data in an easily understandable format for both humans and machines
- enhances neural network training and prediction

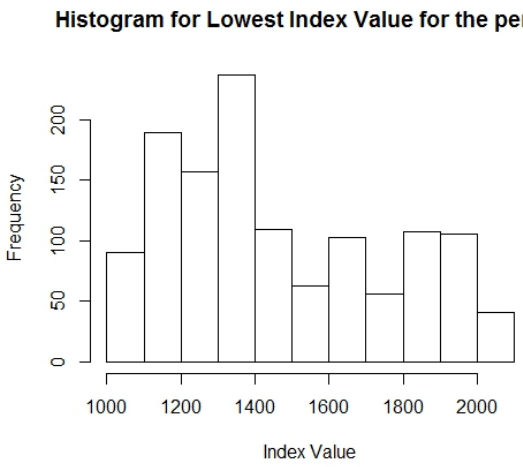
The table and figures below show the summary statistics for the data in its unprocessed form.

Table 1: Summary statistics from unprocessed data

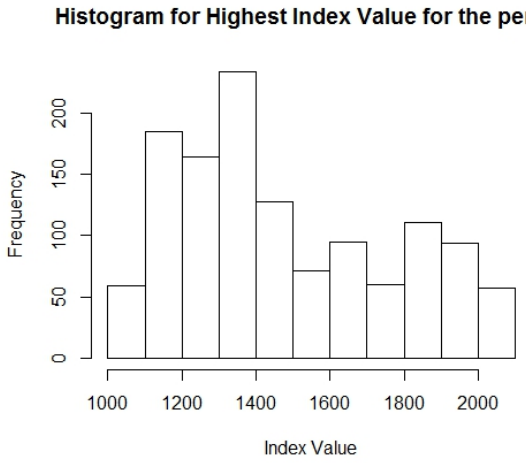
Statistic	N	Mean	St. Dev.	Min	Max
Open	1,258	1,471.89	292.18	1,027.65	2,088.49
High	1,258	1,479.77	291.82	1,032.95	2,093.55
Low	1,258	1,463.70	292.85	1,010.91	2,085.75
Close	1,258	1,472.58	292.38	1,022.58	2,090.57
Volume	1,258	3,805,030,525	925,364,614	1,025,000,000	10,617,809,600
Adj.Close	1,258	1,472.58	292.38	1,022.58	2,090.57



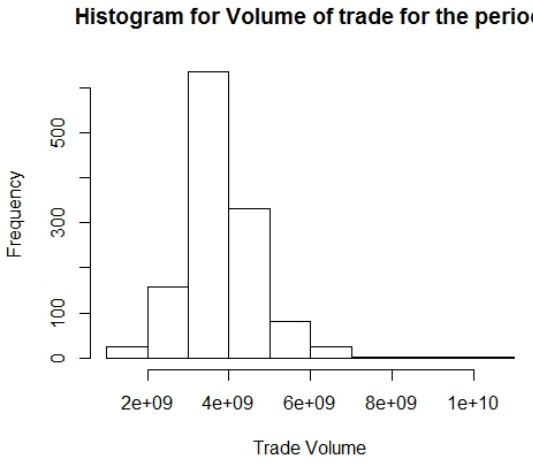
(a) *Daily Open*



(b) *Daily Low*

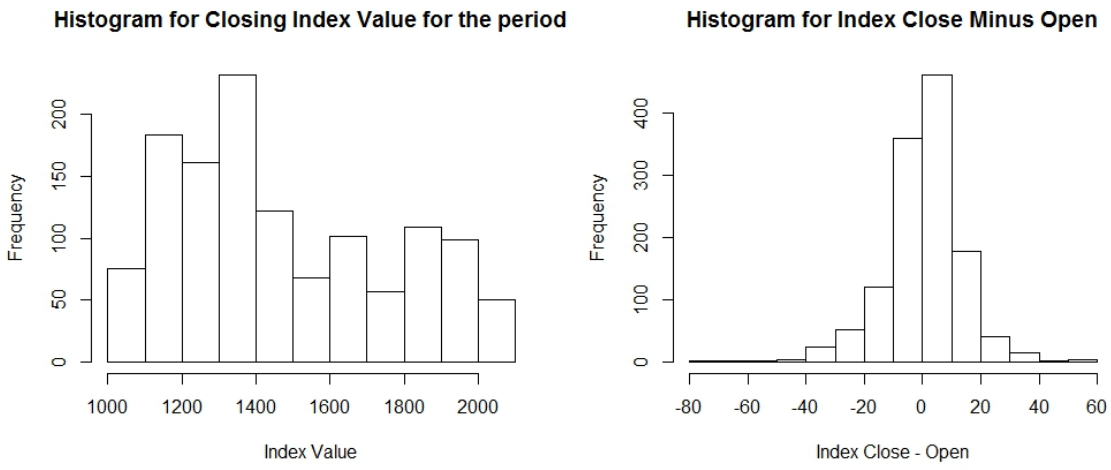


(c) *Daily High*



(d) *Daily Volume*

Figure 6 continued on next page.



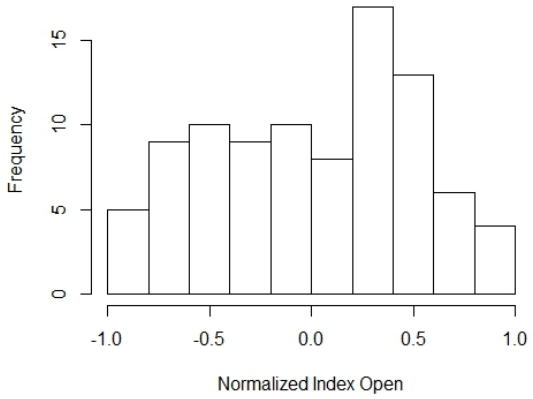
(a) *Daily Close* (b) *Difference in Close and Open*

Figure 6: Histograms of unprocessed data

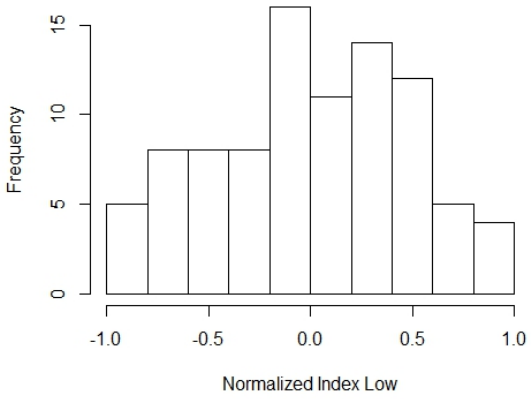
Clearly, we see that the data requires normalization before Granger testing can be applied. The table and figures below show the summary statistics for the normalized data.

Table 2: Summary statistics from normalized data

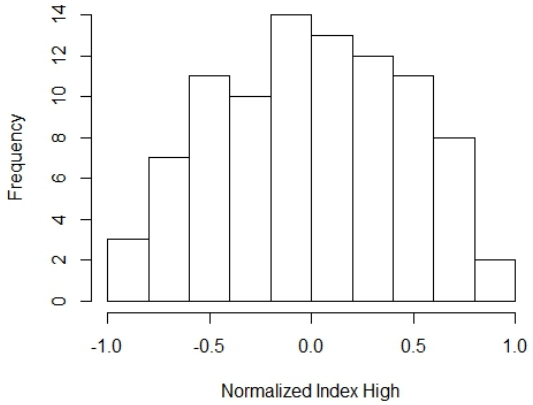
Statistic	N	Mean	St. Dev.	Min	Max
Open	1,258	0.00	0.47	−0.72	1.00
Low	1,258	0.00	0.47	−0.73	1.00
High	1,258	0.00	0.48	−0.73	1.00
Volume	1,258	0.00	0.14	−0.41	1.00
Close	1,258	0.00	0.47	−0.73	1.00



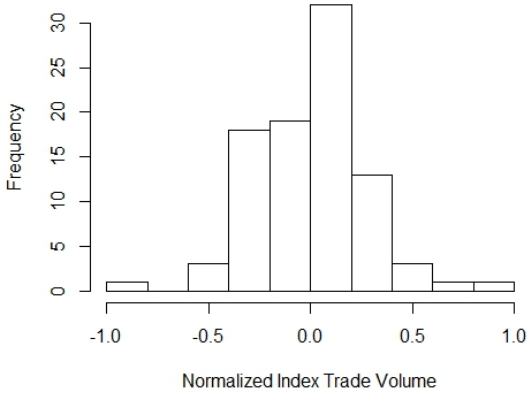
(a) *Normalized Daily Open*



(b) *Normalized Daily Low*

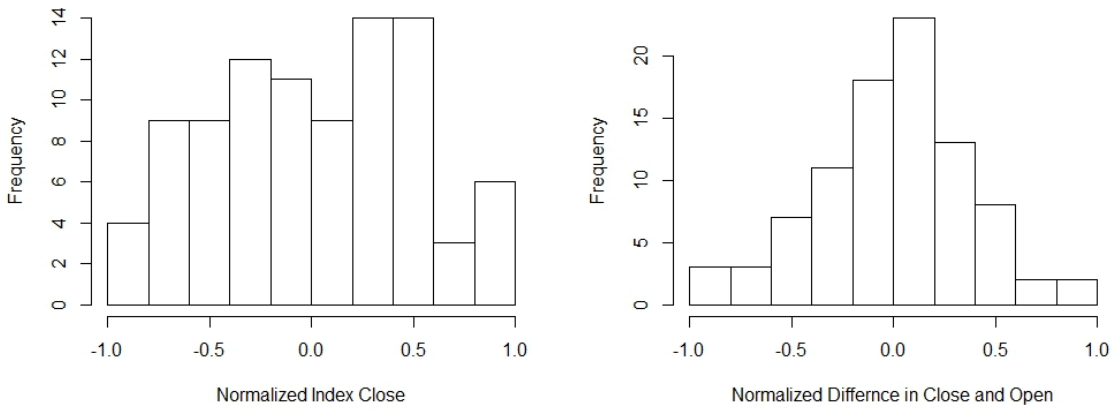


(c) *Normalized Daily High*



(d) *Normalized Daily Volume*

Figure 8 continued on next page.



(a) *Normalized Daily Close* (b) *Normalized difference in Close and Open*

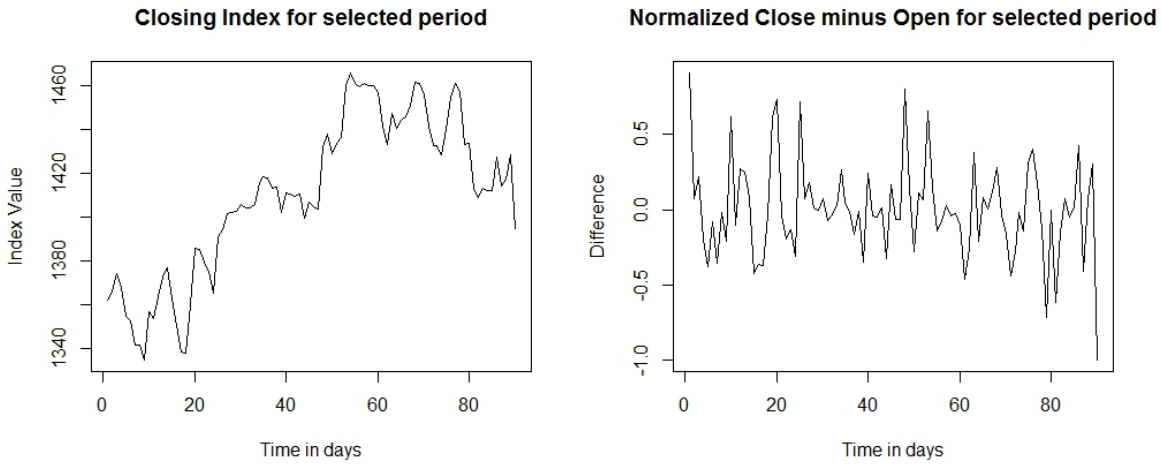
Figure 8: Histograms of normalized data

Now, we run a *grangertest* to select the most predictive sections of the data set. Using the difference between the index close and open, hereafter referred to as "difference", we run a test of causality of the differenced time series with the index close. Thus we seek to find sections of the difference that influence how the dynamics of the index's close on a given day. The results indicate that the underlisted sections have more predictive power based on the p-value.

Table 3: Sections of data with more predictive power

Time in Days	From	To	Granger test p-value
1 - 91	1/4/2010	5/13/2010	0.046
94 - 184	5/18/2010	9/24/2010	0.044
258 - 348	1/6/2011	5/19/2011	0.036
377 - 467	6/30/2011	11/7/11	0.047
629 - 718	6/28/2012	11/7/2012	0.015
1118 - 1208	6/12/2014	10/20/2014	0.048

From the table above, it can be seen that the period (6/28/2012 - 11/7/2012) has the most predictive power. Below are graphs for the closing index and Index difference for the selected period.



(a) *Index Close for selected period* (b) *Index difference for the period*

Figure 9: Plots of Selected Section of data

We first fetch the section of the unprocessed "difference" that corresponds to the selected period. With this data in hand, it is time to estimate the time delay and embedding dimension, τ and d_E. We used the whole sampled data to determine the time lag and embedding dimension with which to construct the Takens vectors. The values of τ and $E(d)$ for the "difference" time series is shown in the figures 10 and 11 respectively. The time lag is estimated using the method of averaged mutual information whiles the embedding dimension is computed by the method of averaged false neighbors. They are 1 and 9 respectively.

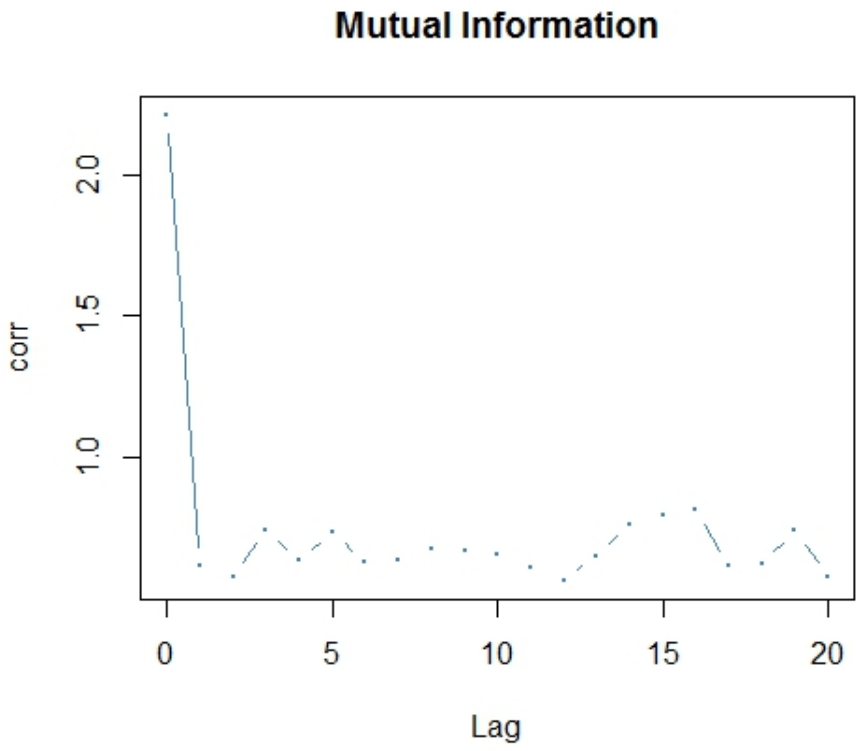


Figure 10: Estimating time lag for difference using Mutual Information

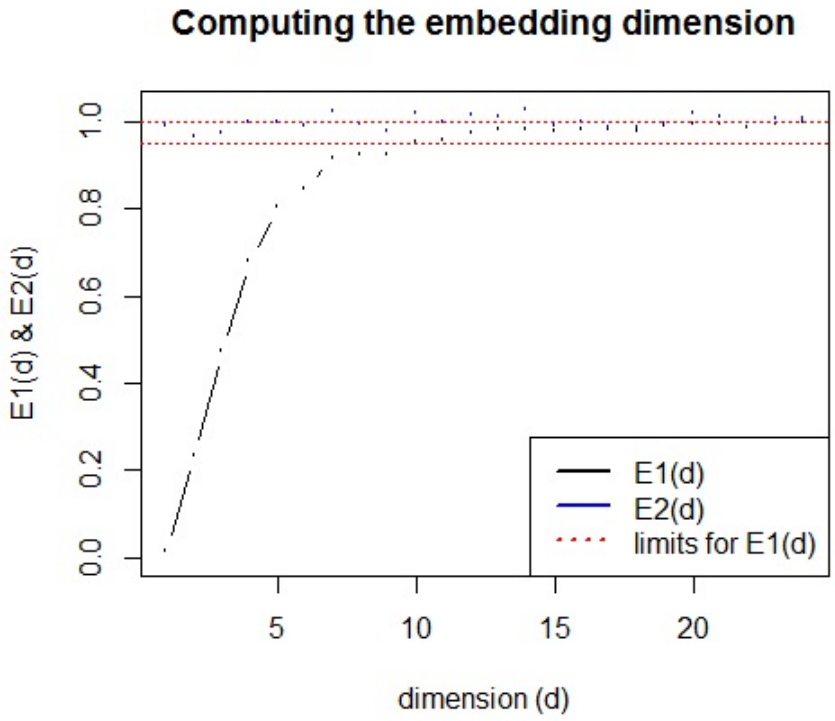


Figure 11: Embedding dimension for difference

With the estimated time delay τ and embedding dimension d_E, we can reconstruct the Takens vectors of for the diffference for the given period as

$$V_t = [x_t, x_{t-\tau}, ..., x_{t-(d_E-1)\tau}] \in \mathbb{R}, i = 1, ..., n$$

where $n = T - (d_E - 1)\tau$ = 91 - (9 - 1)x1 = 83

This implies that the matrix containing the reconstructed vectors has a dimension of 83×9. Each of the 9 vectors become the factors which we feed into the ANN model for training and prediction.

The next step involves feeding the reconstructed vectors into the neural network for training and prediction. The predictive algorithm divides the selected time series

into three parts. The first part, made up of 75% of the total observations, is used to fit the predictive model; the second part, made up of 15% of total observations, is referred to as the internal validation part, is used to search for the optimal values of model parameters, and the third part, with 15% of total observations, is used to test the fitted model.

The figure below shows an AUC of 0.86, an indication of good predictive capability using the Takens vectors as factors.

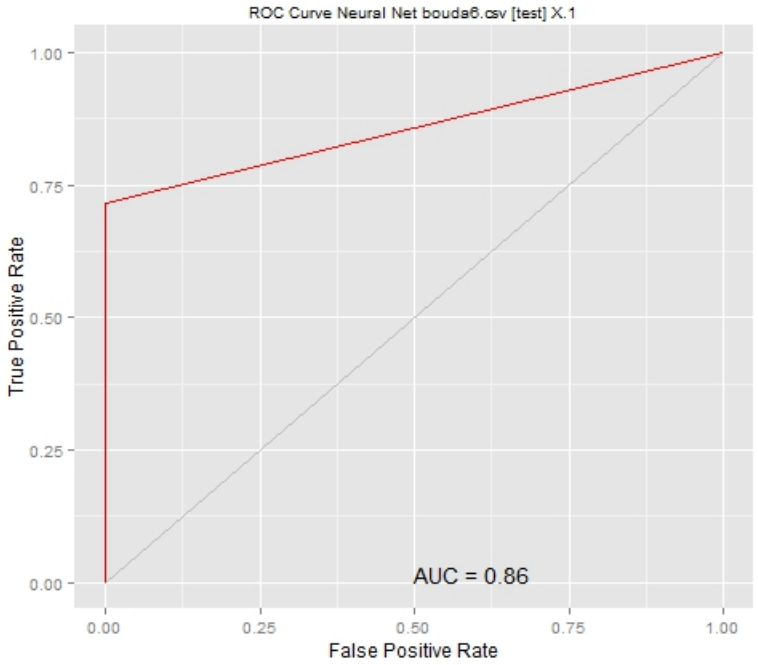


Figure 12: ROC curve from Taken's vectors approach

To make a comparison, we apply the normalization technique to same section of the data spanning 6/28/2012 - 11/7/2012. Using the normalized data of the index *open, low, high, close*, and *volume*, for the said period, we train our model on all five

factors.

The model for the 90 days yields an AUC of 0.57 whiles that for all five years worth of data yields an AUC of 0.56. The AUC of 0.57 and 0.56 shows that the model in this case does a little above random classification. The use of all five years worth of data is to show the impact of using granger test to select a more predictive portion of the 5-year long time series for our analysis. Modeling with only 91 days yields an AUC of 0.57 whiles using all 5 years yields an AUC of 0.56.

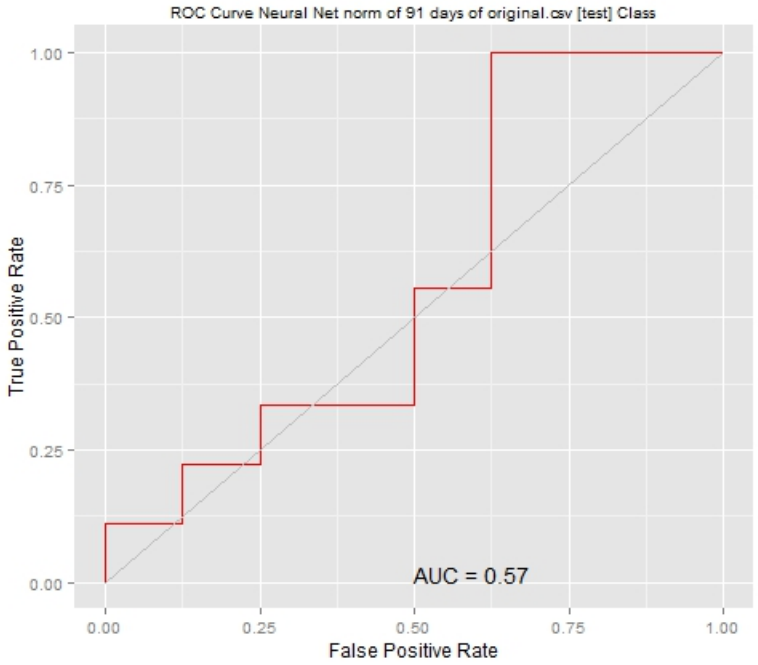


Figure 13: ROC for Model 1(a): 91 days

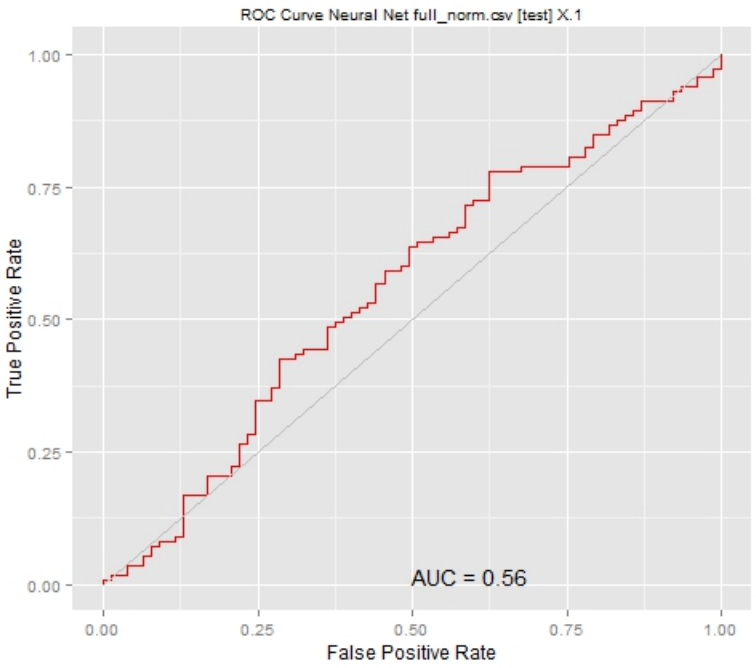


Figure 14: ROC curve for Model 1(b): 5 years

5 CONCLUSION AND REMARKS

Stock market prediction has recently gained much attention. One of the major reasons being the financial gains that accrue from undertaking profitable stock investments.

Over the years, many new methods for the modeling and predicting the stock market, most of which rely on linear statistics. However, increased complexity in financial market dynamics has made it inherently difficult to apply only linear statistics to make predictions. Thus the need to resort to new methods capable of unearthing the dynamics in the market to allow for accurate predictions. We therefore incorporated both linear and nonlinear methods to make predictions in this thesis.

We applied the famous Takens theorem to predict the market' daily movement. We started out applying the usual method of data transformation which only yielded a little above average results. We next, applied our outlined method, which is, given daily stock data for today, we predict whether the market will "go up" or "go down" the following day. The upward or downward movement of the S&P Index is usually seen as a measure of the overall performance of the US stock market, and thus informs many of the decisions taken in the business world.

The results from using the Takens algorithm proved better, with an AUC of 0.86 compared to 0.57 from just using normalization.

We put together an R script for the Takens approach. The script accepts stock data and performs the outlined methodology. All that is needed is the raw data in a proper format. This can be very beneficial for individuals, corporate investors, and financial analysts. With such a model, one can predict before hand, the day-head-

behaviour of a stock market or even for individual stocks. Knowing in advance, the stock's movement leads to making more informed decisions that will rake in monetary gains.

BIBLIOGRAPHY

[1] Takens, F. (1981), *Detecting strange attractors in turbulence*, In: Springer Berlin Heidelberg. (pp. 366-381).

[2] Granger, C. W. J., *Investigating Causal Relations by Econometric Models and Cross-spectral Methods.* Econometrica 1969, 37, 424 438.

[3] Zaremba, Anna, and Tomaso Aste, *Measures of Causality in Complex Datasets with application to financial data.*, Entropy, 16.4 (2014): 2309-2349.

[4] David K., *Nonlinear science.* by Campbell, Los Alamos Science 15 (1987): 218-262.

[5] Bressler, Steven L., and Anil K. Seth., *Wiener-Granger causality: a well established methodology.*, In: Neuroimage 58.2 (2011): 323-329.

[6] Ding, Mingzhou, Yonghong Chen, and S. L. Bressler., *Granger causality: basic theory and application to neuroscience.*, arXiv preprint q-bio/0608035. 2006.

[7] Ladroue, Christophe, et al., *Beyond element-wise interactions: identifying complex interactions in biological processes.* PLoS One 4.9 (2009): e6899.

[8] Kleinberg, Samantha, and Bud Mishra., *The temporal logic of causal structures.*, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. AUAI Press, 2009.

[9] Samantha Kleinberg, Petter Kolm, And Bud Mishra, *Investigating Causal Relationships in Stock Returns with Temporal Logic Based Methods*, arXiv preprint arXiv:1006.1791 (2010).

[10] Garland, Joshua, and Elizabeth Bradley., *Predicting computer performance dynamics.*, Advances in Intelligent Data Analysis X. Springer Berlin Heidelberg, 2011. 173-184.

[11] Singh, Jyoti Prakash, and Paramartha Dutta., *The temporal effect of mobility on path length in MANET.*, International Journal of Wireless Information Networks 19.1 (2012): 38-48.

[12] Erdil, Erkan, and I. Hakan Yetkiner, *The Granger-causality between health care expenditure and output: a panel data approach.*, Applied Economics 41.4 (2009): 511-518.

[13] Bai, Zhidong and Wong, Wing-Keung and Zhang, Bingzhi, *Multivariate linear and nonlinear causality tests*, Published by Elsevier, Mathematics and Computers in Simulation 81.1 (2010): 5-17.

[14] Bai, Zhidong, et al., *Multivariate causality tests with simulation and application."*, Statistics & Probability Letters 81.8 (2011): 1063-1071.

[15] Fontbona, Joaquin, and Benjamin Jourdain., *On the long time behavior of stochastic vortices systems.*, arXiv preprint arXiv:1308.1060 (2013).

[16] Su, Liyun, *Multivariate local polynomial regression with application to Shenzhen component index.*, Discrete Dynamics in Nature and Society 2011 (2011).

[17] Vialar, Thierry, *Complex and chaotic nonlinear dynamics.*, Advances in Economics and Finance, Mathematics and Statistics, Springer (2009).

[18] Sibarama Panigrahi, H. S. Behera, *Effect of Normalization Techniques on Univariate Time Series Forecasting using Evolutionary Higher Order Neural Network*, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 8958, Volume-3, Issue-2, December 2013

[19] Abraham, Ranjit, Jay B. Simha, and S. Sitharama Iyengar, *Effective Discretization and Hybrid feature selection using Nave Bayesian classifier for Medical datamining.*, International Journal of Computational Intelligence Research 5.2 (2009): 116-129.

[20] T.Jayalakshmi, Dr.A.Santhakumaran, *Statistical Normalization and Back Propagation for Classification*, In: International Journal of Computer Theory and Engineering, Vol.3, No.1, February, 2011, 1793-8201

[21] Qing-Fang, Meng, Peng Yu-Hua, and Xue Pei-Jun, *A new method of determining the optimal embedding dimension based on nonlinear prediction.*, Chinese Physics 16.5 (2007): 1252.

[22] Holger Kantz & Thomas Schreiber, *Nonlinear Time Series Analysis*, Cambridge University Press - 2004

[23] J. P. Huke, *Embedding Nonlinear Dynamical Systems: A Guide to Takens' Theorem*, Defence Research Agency, March 2006

[24] Norman H. Packard, James P. Crutchfield, J. Doyne Farmer and Robert S. Shaw, *Geomtry from a Time Series*, Physical Review Letters 45 (1980): 712–716

[25] Sauer et al., *Embedology*, J. Stat. Phys. Vol. 65 (1991) 579

[26] Cao,L., *Practical method for determining the minimum embedding dimension of a scalar time series.*, Physica D 110, 4350 (1997).

[27] Landa, Polina S., *Nonlinear Oscillations and Waves in Dynamical Systems*, Vol. 360. Springer Science & Business Media, 2013.

[28] Huffaker, R., *Phase Space Reconstruction from Time Series Data: Where History Meets Theory.*, Proceedings in Food System Dynamics (2010): 1-9.

[29] Norbert, W., Beckenbach, E.F., Ed., *The Theory of Prediction. In Modern Mathematics for Engineers*; McGraw-Hill: New York, NY,USA, 1956; Volume1.

[30] Casdagli, M., S. Eubank, J. Farmer, and J. Gibson, *State space reconstruction in the presence of noise*, in Physica D: Nonlinear Phenomena 51.2-3, 52-98, (cit. on pp. 18).

[31] Daniel K. N. Johnson and Hunter Phoenix Van Wagoner, *The Chicken or the Egg: Granger-Causality between Trade and Innovation.*, Colorado College Working Paper No. 2014-04

[32] Erkan Erdila, and I. Hakan Yetkinerb, *The Granger-causality between health care expenditure and output: a panel data approach*, Applied Economics 41.4 (2009): 511-518.

[33] Foresti, Pasquale *Testing for Granger Causality Between Stock Prices and Economic Growth.* (2006).

[34] Faes, L., A. Porta, and G. Nollo (2008), *Mutual nonlinear prediction as a tool to evaluate coupling strength and directionality in bivariate time series: Comparison among diferent strategies based on k nearest neighbors.*, In: Physical Review E 78.2, p. 026201 (cit.on p. 18).

[35] Basharat, Arslan, and Mubarak Shah, *Time Series Prediction by Chaotic Modeling of Nonlinear Dynamical Systems*, In: Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009.

[36] K, Radko, *Finding chaos in Finnish GDP.*, International Journal of Automation and Computing 11.3 (2014): 231-240.

[37] Cao, Liangyue, and Abdol S. Soofi., *Nonlinear deterministic forecasting of daily dollar exchange rates.*, In: International Journal of Forecasting 15.4 (1999): 421-430.

[38] Ma, Hong-guang, and Chong-zhao Han., *Selection of embedding dimension and delay time in phase space reconstruction.*, In: Frontiers of Electrical and Electronic Engineering in China 1.1 (2006): 111-114.

[39] Strozzi, Fernanda, Jos-Manuel Zaldvar, and Joseph P. Zbilut, *Application of nonlinear time series analysis techniques to high-frequency currency exchange data.*, In: Physica A: Statistical Mechanics and its Applications 312.3 (2002): 520-538.

[40] Amblard,P.O.; Michel,O.J.J., *On directed information theory and Granger causality graphs.*, In: J. Comput. Neurosci. 2011, 30, 716.

[41] Chen, Y., G. Rangarajan, J. Feng, and M. Ding (2004)., *Analyzing multiple nonlinear time series with extended Granger causality.*, In: Physics Letters A 324.1,pp. 2635 (cit. on pp. 18, 35).

[42] Hess, Gregory D., and Mark E. Schweitzer., *Does Wage Inflation Cause Price Inflation?*, In: FRB of Cleveland Policy Discussion Paper 1 (2000).

[43] Smith, Leslie., *An introduction to neural networks.*, Center for Cognitive and Computational Neuroscience, Department of Computing & Mathematics, University of Stirling (1996).

[44] S&P 500, S&P Dow Jones Indices by McGRAWHILL Financial at https://us.spindices.com/indices/equity/sp-500, accessed 06/16/2015

[45] Neural Network Classification by Fronline Solvers, at http://www.solver.com/xlminer/help/neural-networks-classification-intro, accessed 06/16/2015

[46] Knobloch, Edgar (1981)., *Chaos in the segmented disc dynamo*, In: Physics Letters A 82 (9): 439440. Bibcode:1981PhLA...82..439K. doi:10.1016/0375-9601(81)90274-7

[47] Thurman, Walter N., and Mark E. Fisher. *Chickens, eggs, and causality, or which came first.* American Journal of Agricultural Economics 70.2 (1988): 237-238.

[48] Gibson, J. F., J. D. Farmer, M. Casdagli, and S. Eubank (1992). *An analytic approach to practical state space reconstruction.* In: Physica D 57.1-2,pp. 130 (cit. on p. 18).

[49] Schiff, S. J.,P. So, T. Chang, R. E. Burke, and T. Sauer, *Detecting dynamical interdependence and generalized synchrony through mutual prediction in a neural ensemble.* In: Physical Review E 54.6, p. 6708 (cit. on p. 18) (1996).

[50] Martin Castagli *Chaos and Deterministic versus Stochastic Nonlinear Modeling* SFI working paper: 1991-07-029

[51] Michael Small, C.K. Tse *Optimal embedding parameters: A modeling paradigm*

[52] Stam, Cornelis J. *Nonlinear brain dynamics.* New York: Nova Science Publishers, 2006.

[53] S. Nargundkar and J. Lewis, *Assessment of Evaluation Methods for Prediction and Classification of Consumer Risk in the Credit Industry*

[54] *Excerpt from Robert Hagin, Modern Portfolio Theory, Dow Jones-Irwin, 1979, pp. 11-13 & 89-91*

[55] Burton G. Malkiel *The Efficient Market Hypothesis and Its Critics* Journal of Economic PerspectivesVolume 17, Number 1Winter 2003Pages 59 82

[56] Scheinkman and LeBaron, *Nonlinear Dynamics and Stock Returns* The Journal of Business, Vol. 62, No. 3. (July., 1989), pp. 311-337

[57] M. Hinich and D. Patterson, *Evidence of Nonlinearity in Daily Stock Returns* Journal of Business & Economic Statistics Vol. 3, No. 1 (Jan., 1985), pp. 69-77

[58] David G. McMillan, *Nonlinear predictability of stock market returns: Evidence from nonparametric and threshold models* International Review of Economics and Finance 10 (2001) 353 - 368

[59] Hsieh, D. A., *Chaos and nonlinear dynamics: Application to financial markets, 1991*, The Journal of Finance 46, 1839-1887

[60] Hiemstra, C. and Jones, J. D. *Testing for linear and nonlinear Granger causality in the stock price-volume relation.* Journal of Finance, 49, number 5, 16391664. (1994)

[61] Ruey S. Tsay, *Analysis of Financial Time Series, Second Edition by Wiley, 2005*, ISBN: 0-471-69074-0

[62] Mees, A. I., Rapp, P. E. and Jennings L. S., 1987, *Singular value decomposition and embedding dimension.* Phys. Rev. A 36, 340.

[63] Rosenstein MT, Collins JJ, De Luca CJ. *Reconstruction expansion as a geometry-based framework for choosing proper delay times.* Physica D 1994;73:8298

[64] Fraser, A.M. and Swinney, H.L. (1986) *Independent Coordinates for Strange Attractors from Mutual Information.* Physical Review A, 33, 1134-1140.

[65] Ye Yuan, Yue Li, and Danilo P. Mandic, *Comparison Analysis of Embedding Dimension between Normal and Epileptic EEG Time Series*

The Journal of Physiological Sciences Advance Publication by J-STAGE doi:10.2170/physiolsci.RP004708

[66] Abdol S. Soofi, Andreas Galka, Zhe Li, Yuqin Zhang and Xiaofeng Hui, *Applications of Methods and Algorithms of Nonlinear Dynamics in Economics and Finance*, Complexity in Economics: Cutting Edge Research New Economic Windows 2014, pp 1-30

[67] M. khalil Sari Alsmadi, Khairuddin Bin Omar and Shahrul Azman Noah, *Back Propagation Algorithm: The Best Algorithm Among the Multi-layer Perceptron Algorithm* IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.4, April 2009

[68] Constantino A. Garcia (2014). *nonlinearTseries: R package for nonlinear time series analysis.* R package version 0.2.1. http://CRAN.R-project.org/package=nonlinearTseries

[69] Diethelm Wuertz, many others and see the SOURCE file (2013). *fNonlinear: Nonlinear and Chaotic Time Series Modelling.* R package version 3010.78. http://CRAN.R-project.org/package=fNonlinear

[70] Hann, J., Kamber, M. (2000). Data Mining: Concepts and Techniques. Morgan Kaufman Publishers.

[71] Priddy, Kevin L. and Paul E. Keller. *Artificial neural networks : an introduction* p. cm

[72] M. Henon, *Numerical study of quatratic area-preserving mappings,* Quart. Appl. Math. 27 (1969) 291.

[73] Haoran Wen *A review of the Henon map and its physical interpretations* Georgia Tech PHYS 7224: Nonlinear Dynamics, course project, spring semester 2014.

[74] Hartshorne, Robin. *Graduate Texts in Mathematics. Vol. 52, Algebraic Geometry.* New York: Springer Science+Business Media, Inc., 2010. Accessed July 27, 2015.

[75] *chapter11: Stationary and non-stationary time series from Statistics in volcanology.* by G. P. Nason, Geological Society of London, 2006.

[76] Khan, Asif Ullah, T. K. Bandopadhyaya, and Sudhir Sharma. *"Stocks selected using SOM and Genetic Algorithm based Backpropagation Neural Network gives better returns."* International Journal of Computer Science 8.2 (2011): 664-668.

[77] Shital Patil and Prof. G. K. Andurkar, *Analysis of Two Intermediate Layers in Neural Network on Optical Character Recognition: A Result Analysis* International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 5, May 2014)

[78] Schumacher, Ernst Friedrich. *Small is beautiful.* Rowohlt Taschenbuch Verlag, 1985.

APPENDIX

0.1 R Code for Replicating the Methods

```
install.packages("nonlinearTseries")
library(nonlinearTseries)
install.packages("fNonlinear")
library(fNonlinear)
library(lmtest)
library(rattle)
install.packages("clusterSim")
library(clusterSim)
library(ROCR)


stock<-read.csv("stock10.csv", header = TRUE)
names(stock)
plot.ts(stock$Close)
plot.ts(stock$Open)
close<-stock$Close[2:nrow(stock)]
n<-length(close)
open<-stock$Open[1:n]
ch<-(stock$Close-stock$Open)
```

```
norm<-data.Normalization(ch,type="n5",
                         normalization="column")

norm_close<-data.Normalization(close,type="n5",
                         normalization="column")
plot.ts(norm, col="Blue")
plot.ts(norm_close, col="red")

for ( t in 91:length(close) ) {
  data = data.frame( prior = norm[(t-90):t],
                     post = norm_close[(t-90):t] )
  tst = grangertest( post ~ prior, order=3, data=data )
  p = tst$"Pr(>F)"[2]
  if( p < 0.05 ) {
    print(table(p,t))
  }
}

f<-718
```

```
s<-f-90
section<-ch[s:f]
mutualPlot(section, partitions = 16, lag.max = 20,
                                        doplot = TRUE)
d_E = estimateEmbeddingDim(section,
                           number.points = length(section),
                           time.lag = 1,
                           max.embedding.dim = 25,
                           threshold = 0.95,
                           max.relative.change = 0.1,
                           do.plot = TRUE)

Takens_dif = buildTakens(section, embedding.dim = d_E,
                                        time.lag = 1)
class<-stock$Class[(s+1):(f-d_E+2)]
feednet<-data.frame(Takens_dif, class)
write.csv(feednet, file = "feed_net.csv")
dataset<-read.csv("feed_net.csv", header = TRUE)

building <- TRUE
```

```
scoring  <- ! building

library(colorspace)

crv$seed <- 42

crs$dataset <- dataset

set.seed(crv$seed)

crs$nobs <- nrow(crs$dataset)

crs$sample <- crs$train <- sample(nrow(crs$dataset),
                                  0.7*crs$nobs)

crs$validate <- sample(setdiff(seq_len(nrow(crs$dataset)),
                               crs$train), 0.14*crs$nobs)

crs$test <- setdiff(setdiff(seq_len(nrow(crs$dataset)),
                            crs$train), crs$validate)


crs$input <- c("X", "X1", "X2", "X3", "X4", "X5",
               "X6", "X7", "X8", "X9")


crs$numeric <- c("X", "X1", "X2", "X3", "X4", "X5",
                 "X6", "X7", "X8", "X9")

crs$categoric <- NULL

crs$target  <- "class"
```

```
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- NULL
crs$weights <- NULL


require(nnet, quietly=TRUE)
set.seed(199)
crs$nnet <- nnet(as.factor(class) ~ .,
                 data=crs$dataset[crs$sample,c(crs$input,
                 crs$target)], size=10, skip=TRUE, MaxNWts=10000,
                  trace=FALSE, maxit=100)
cat(sprintf("A %s network with %d weights.\n",
                    paste(crs$nnet$n, collapse="-"),
                    length(crs$nnet$wts)))
cat(sprintf("Inputs: %s.\n",
                 paste(crs$nnet$coefnames, collapse=", ")))
cat(sprintf("Output: %s.\n",
                 names(attr(crs$nnet$terms, "dataClasses"))[1]))
cat(sprintf("Sum of Squares Residuals: %.4f.\n",
                 sum(residuals(crs$nnet) ^ 2)))
```

```
require(ggplot2, quietly=TRUE)
crs$pr <- predict(crs$nnet, newdata=crs$dataset[crs$test,
                                    c(crs$input, crs$target)])
no.miss   <- na.omit(crs$dataset[crs$test, c(crs$input,
                                    crs$target)]$class)
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL

if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}

pe <- performance(pred, "tpr", "fpr")
au <- performance(pred, "auc")@y.values[[1]]
pd <- data.frame(fpr=unlist(pe@x.values), tpr=unlist(pe@y.values))
```

```
p <- ggplot(pd, aes(x=fpr, y=tpr))
p <- p + geom_line(colour="red")
p <- p + xlab("False Positive Rate") + ylab("True Positive Rate")
p <- p + ggtitle("ROC Curve Neural Net dataset [test] class")
p <- p + theme(plot.title=element_text(size=10))
p <- p + geom_line(data=data.frame(), aes(x=c(0,1), y=c(0,1)),
                                        colour="grey")
p <- p + annotate("text", x=0.50, y=0.00, hjust=0, vjust=0, size=5,
                                         label=paste("AUC =", round(au, 2)))
print(p)


##NORMALIZATION PROCEDURE
select<-stock[s:f, c('Open', 'Low', 'High',
                                             'Close', 'Volume')]

norm_open<-data.Normalization(select[,1],type="n5",
                                          normalization="column")
norm_low<-data.Normalization(select[,2],type="n5",
                                         normalization="column")
```

```
norm_high<-data.Normalization(select[,3],type="n5",
                                        normalization="column")
norm_close<-data.Normalization(select[,4],type="n5",
                                         normalization="column")
norm_vol<-data.Normalization(select[,5],type="n5",
                                        normalization="column")

frame<-data.frame(norm_open, norm_low, norm_high,
                                          norm_close, norm_vol)

class2<-stock$Class[(s+1):(f+1)]
combined<-data.frame(frame, class2)
write.csv(combined, file = "combined.csv")

combined2<-read.csv("combined.csv", header = TRUE)
crs$dataset <- combined2
set.seed(crv$seed)
crs$nobs <- nrow(crs$dataset)
crs$sample <- crs$train <- sample(nrow(crs$dataset),
                                        0.69*crs$nobs)
```

```
crs$validate <- sample(setdiff(seq_len(nrow(crs$dataset)),
                               crs$train), 0.14*crs$nobs)
crs$test <- setdiff(setdiff(seq_len(nrow(crs$dataset)),
                            crs$train), crs$validate)

crs$input <- c("X", "norm_open", "norm_low", "norm_high",
                           "norm_close", "norm_vol")

crs$numeric <- c("X", "norm_open", "norm_low", "norm_high",
                             "norm_close", "norm_vol")
crs$categoric <- NULL
crs$target  <- "class2"
crs$risk    <- NULL
crs$ident   <- NULL
crs$ignore  <- NULL
crs$weights <- NULL

require(nnet, quietly=TRUE)
set.seed(199)
crs$nnet <- nnet(as.factor(class2) ~ .,
```

```
                    data=crs$dataset[crs$sample,
                    c(crs$input, crs$target)],size=10,
                    skip=TRUE, MaxNWts=10000,
                    trace=FALSE, maxit=100)

cat(sprintf("A %s network with %d weights.\n",
            paste(crs$nnet$n, collapse="-"),
            length(crs$nnet$wts)))
cat(sprintf("Inputs: %s.\n",
            paste(crs$nnet$coefnames, collapse=", ")))
cat(sprintf("Output: %s.\n",
            names(attr(crs$nnet$terms, "dataClasses"))[1]))
cat(sprintf("Sum of Squares Residuals: %.4f.\n",
            sum(residuals(crs$nnet) ^ 2)))

require(ggplot2, quietly=TRUE)
crs$pr <- predict(crs$nnet, newdata=crs$dataset[crs$test,
                                c(crs$input, crs$target)])

no.miss   <- na.omit(crs$dataset[crs$test, c(crs$input,
```

```
                                        crs$target)]$class2)
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL

if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}

pe <- performance(pred, "tpr", "fpr")
au <- performance(pred, "auc")@y.values[[1]]
pd <- data.frame(fpr=unlist(pe@x.values),
                                    tpr=unlist(pe@y.values))
p <- ggplot(pd, aes(x=fpr, y=tpr))
p <- p + geom_line(colour="red")
p <- p + xlab("False Positive Rate") +
                                    ylab("True Positive Rate")
```

```
p <- p + ggtitle("ROC Curve Neural Net combined2 [test] class2")
p <- p + theme(plot.title=element_text(size=10))
p <- p + geom_line(data=data.frame(), aes(x=c(0,1), y=c(0,1)),
                                          colour="grey")
p <- p + annotate("text", x=0.50, y=0.00, hjust=0, vjust=0,
                  size=5, label=paste("AUC =", round(au, 2)))
print(p)
```

VITA

ABUBAKAR-SADIQ BOUDA ABDULAI

Education:	B.S. Actuarial Science, Kwame Nkrumah University Kumasi, Ghana, 2007 – 2011 M.S. Mathematical Science, East Tennessee State University Johnson City, Tennessee, 2015
Professional Experience:	Graduate Teaching Assistant, East Tennessee State University, Johnson City, TN, 2014 – 2015