Spring 2011

Characterization of normality of chaotic systems including prediction and detection of anomalies

Joseph John Engler
University of Iowa

CHARACTERIZATION OF NORMALITY OF CHAOTIC SYSTEMS INCLUDING PREDICTION AND DETECTION OF ANOMALIES

by

Joseph John Engler

An Abstract

Of a thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy degree
in Industrial Engineering in
the Graduate College of
The University of Iowa

May 2011

Thesis Supervisor: Professor Andrew Kusiak

ABSTRACT

Accurate prediction and control pervades domains such as engineering, physics, chemistry, and biology. Often, it is discovered that the systems under consideration cannot be well represented by linear, periodic nor random data. It has been shown that these systems exhibit deterministic chaos behavior. Deterministic chaos describes systems which are governed by deterministic rules but whose data appear to be random or quasi-periodic distributions.

Deterministically chaotic systems characteristically exhibit sensitive dependence upon initial conditions manifested through rapid divergence of states initially close to one another. Due to this characterization, it has been deemed impossible to accurately predict future states of these systems for longer time scales. Fortunately, the deterministic nature of these systems allows for accurate short term predictions, given the dynamics of the system are well understood. This fact has been exploited in the research community and has resulted in various algorithms for short term predictions.

Detection of normality in deterministically chaotic systems is critical in understanding the system sufficiently to able to predict future states. Due to the sensitivity to initial conditions, the detection of normal operational states for a deterministically chaotic system can be challenging. The addition of small perturbations to the system, which may result in bifurcation of the normal states, further complicates the problem. The detection of anomalies and prediction of future states of the chaotic system allows for greater understanding of these systems.

The goal of this research is to produce methodologies for determining states of normality for deterministically chaotic systems, detection of anomalous behavior, and the

more accurate prediction of future states of the system. Additionally, the ability to detect subtle system state changes is discussed. The dissertation addresses these goals by proposing new representational techniques and novel prediction methodologies. The value and efficiency of these methods are explored in various case studies.

Presented is an overview of chaotic systems with examples taken from the real world. A representation schema for rapid understanding of the various states of deterministically chaotic systems is presented. This schema is then used to detect anomalies and system state changes. Additionally, a novel prediction methodology which utilizes Lyapunov exponents to facilitate longer term prediction accuracy is presented and compared with other nonlinear prediction methodologies. These novel methodologies are then demonstrated on applications such as wind energy, cyber security and classification of social networks.

Abstract Approved: _____
Thesis Supervisor

_____
Title and Department

_____
Date

CHARACTERIZATION OF NORMALITY OF CHAOTIC SYSTEMS INCLUDING PREDICTION AND DETECTION OF ANOMALIES

by

Joseph John Engler

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy degree
in Industrial Engineering in
the Graduate College of
The University of Iowa

May 2011

Thesis Supervisor: Professor Andrew Kusiak

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph. D. thesis of

Joseph John Engler

has been approved by the Examining Committee
for the thesis requirement for the Doctor of
Philosophy degree in Industrial Engineering
at the May 2011 graduation.

Thesis Committee: _____
Andrew Kusiak, Thesis Supervisor

_____
Thomas Schnell

_____
Geb Thomas

_____
Albert Ratner

_____
Juan Pablo Hourcade

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

Accurate prediction and control of systems is an issue which pervades many domains including engineering, physics, chemistry and medicine. Often, it is discovered that the systems under consideration are not well represented as linear, periodic nor random. It has however been shown that these types of systems can be well represented as being influenced by deterministic chaos. Deterministic chaos is a relatively new term used to describe systems which are governed by deterministic equations of motion but whose data may appear to be random or quasi-periodic [1].

While an exact definition for chaos does not yet exist, systems which are governed by deterministic chaos are fundamentally characterized by: (1) an essentially continuous and possibly banded frequency spectrum that resembles random noise; (2) sensitivity to initial conditions such that states initially close to one another often diverge at an exponential rate; and (3) an ergodicity and mixing of the dynamical state trajectories which in essence implies the wholesale visit of the entire phase space by the chaotic behavior [2]. Often, deterministically chaotic systems are misclassified as stochastic systems with noise [3].

Due to the characteristic traits, and the misclassification of systems as stochastic rather than chaotic, prediction of future states of these systems has been problematic. This is especially true for prediction in large time scales. Deterministically chaotic systems can be predicted very accurately if the equations of motion are known, or, to a lesser extent, if the system is analyzed from a chaotic point of view. Unfortunately, it is often the case that the equations of motion are not known or that analysis of the systems is performed external to chaotic phase space.

Detection of normality in deterministically chaotic systems is critical to understanding the system sufficiently for prediction of future states. Due to the sensitivity to initial conditions, the detection of normal operational states for a deterministically chaotic system can be challenging. Complicating the problem further is that the addition of small perturbations to the system which may result in bifurcation of the normal states. The detection of anomalies and the ability to predict future states allows for greater understanding of the system's normal states.

Many theories for prediction and control of nonlinear systems have been posited in the literature. Unfortunately, these methodologies are inefficient and ineffective for determination of the states of normality for a chaotic system and are not applicable in detection and prediction of divergences from the norm of the system. Nonlinear smoothing [4], data mining and machine learning algorithms [5], as well as spatial transformation methodologies [6] have been reported in the literature. Unfortunately, these methodologies have proven to provide only limited accuracy over longer time frames for chaotic systems.

Rapid advances in technology, especially in safety critical domains, have resulted in a greater need for accurate description and fault prediction of deterministically chaotic systems. Domains such as the Smart Grid [7], alternative energy control, advanced cryptographic communications and even the human heart are driven by deterministic chaos and require accurate state characterization for prediction of future anomalies. Current methodologies are appropriate for approximating short term prediction and control [1]. However, these methods are less effective for longer term anomaly and fault detection.

1.1 Motivation and Research Objective

This research is specifically directed towards systems which are deterministically chaotic (examples of these systems and a complete description of chaotic systems is given in Chapter 2 of this thesis). Recently, Chaudhury *et al*. [8] reported the discovery of chaotic characteristics at the quantum level. Should this finding prove to be correct, the statement that chaos is found universally cannot be denied. Until such time as this discovery is validated, it is sufficient to state that deterministic chaos is present in a great many systems. As such, it must be understood and accounted for in system analysis, prediction, control and monitoring. This requirement fuels, in part the motivation of the presently proposed research.

In addition to the ubiquitous nature of deterministically chaotic systems, such systems are often fallaciously classified as stochastic systems or periodic systems heavily influenced by noise. The results of these misclassifications are inappropriate control strategies, incomplete system understanding, and suboptimal prediction and anomaly detection schemas. To achieve results satisfactory to a rapidly changing technological society it is critical for systems to be correctly identified and understood.

A second motivation for this research is the fact that deterministically chaotic systems often have an interesting topology and geometric structure. This geometry is formulated through the complex interactions which make up such systems. As will be presented in this thesis, very simple equations of motion can generate extremely complex behaviors which can be transformed into a number of topologies and geometries. Often the spaces in which these systems are analyzed are of fractional dimension and produce

beautiful natural characteristics, both from a mathematical perspective as well as a qualitative perspective.

The final motivation for this research is the fact that the domain of complexity and deterministic chaos is still a relatively young, open field. The first major discovery of chaotic characteristics was not published until 1963 when Lorenz reported the sensitivity of his systems to initial conditions [9]. The term "*chaos*" was first utilized to describe deterministically nonlinear dynamical systems with the work of Li and Yorke regarding the Logistic equation in 1975 [10]. While there was a great fervor of work performed in the domain of chaos during the 1980's and 1990's there still remain a large number of open problems in this field. Thus, the domain is relatively young and there exists much room for growth.

Based on the above mentioned motivations, this research seeks to fulfill a number of objectives related to the domain of deterministic chaos. These objectives are listed, in no particular order, below.

- The development of a course grained representation of the dynamics of the chaotic system.
- The use of the course grained representation for the detection of normal system states in chaotic systems.
- The development of methodologies for real-time, or near real-time, detection of system state changes and anomalies.
- The development of more accurate prediction techniques for deterministically chaotic systems.

It is hypothesized that the development of a course grained representation of the dynamical system will assist in a clearer understanding of the system states and the

traversal of these states by the chaotic system. Obtaining such an understanding of the chaotic system should lead to the ability to detect system normality. With the detection of system normality is the implied ability to understand abnormal behavior. As such, it is hypothesized that the detection of abnormal, anomalous and shifted (state change) behavior should be achievable in an accurate manner.

The objective of the development of real-time, or near real-time, methodologies for the detection of system state change and anomalous behavior is becoming more critical as deterministically chaotic systems are being discovered in systems which are themselves critical. Chaotic systems, such as the impending Smart Grid, medical prosthetics which make use of real-time computing (e.g. pace makers and brain controlled prosthetic limbs), communications and cyber security, require accurate monitoring for changed behavior. It is hypothesized that the change detection methodology discussed above can be utilized in a real-time, or near real-time, environment for the benefit of such systems.

Finally, there exists a need to develop more accurate prediction methodologies for deterministically chaotic systems. As will be presented, often prediction of the systems being studied here are attempted in the stochastic realm which is suboptimal given that these systems are not derived from such distributions. The need for accurate prediction is also present in the other objectives of this research, if only for the completion of missing data values in the experimental datasets.

1.2 Expected Contributions

This thesis proposes new methodologies for prediction and detection of normality, state changes, and anomalies in deterministically chaotic systems. The expected contributions of this research are listed below.

- A novel course-grained visualization technique for representation of the states of a chaotic system.
- A novel state prediction methodology utilizing the above visualization technique.
- A novel methodology for representing the normality of a deterministically chaotic system
- A novel, computationally simple, mathematical representation of the system state traversal of a deterministically chaotic system which allows for near real-time detection of anomalies and system state changes.
- A new prediction algorithm for more accurate prediction of deterministically chaotic systems.

To validate the accuracy and effectiveness of the novel algorithms, real world case studies are presented. Detection of the normal and anomalous states of systems is preformed on case studies from various domains. Prediction of various attributes of wind turbines and other devices is given to validate the novel prediction methodology.

The remainder of this thesis is organized as follows. The next section offers an overview of deterministic chaos. This overview includes examples of real and theoretical systems which exhibit deterministic chaos. Chapter 3 presents a new methodology for understanding normal system states. Included in this section is the detection of anomalous states of chaotic systems. Chapter 4 presents a novel prediction algorithm and illustrates its accuracy and effectiveness on various real and theoretical datasets. Chapter

5 describes a novel methodology for the detection of system shifts and anomalies within chaotic systems. Validation of the proposed theories on real world applications is presented in Chapter 6 and conclusions are given in Chapter 7.

CHAPTER 2. DETERMINISTIC CHAOS OVERVIEW

Many domains, such as physics, hydrodynamics, biology and medicine, have struggled to develop satisfactory models and methods for complex datasets. Some researchers in these fields have looked towards deterministic chaos to interpret the complex datasets. Fraser [11] suggested that this is due to the fact that phenomena whose explanations were previously thought to require high-dimensional phase space or independent noise sources can be explained in terms of simple low-dimensional deterministic chaos models. But what is deterministic chaos? This section offers an introduction to deterministic chaos. Presented are the basic properties, or characteristics of deterministically chaotic systems (e.g. §2.1), methodologies to determine if a dataset originated from a deterministically chaotic system (e.g. §2.2) as well as an overview of chaotic attractors, the topological spaces in which these objects occur, and the benefits to considering systems in these spaces (e.g. §2.3).

2.1 Properties of Deterministically Chaotic Systems

Deterministic chaos is a term used to represent systems which are governed by deterministic equations of motion but whose datasets resemble random or quasi-periodic (multiply periodic) data [12]. Hidden within this loose definition is the fact that even extremely simple equations of motion can represent very complex behavior. Grassberger *et al* [13] stated that the simplest examples of chaotic systems are represented by systems which map some interval onto itself. The Logistic equation given in Eq. 1 is a well known simple example of such a mapping. The data for the map of Eq. 1 is plotted in Fig.

1 with the horizontal axis representing time increments, *t*, and the vertical axis representing the value of the map, *x*, at time *t*. The value of *a* in Fig. 1 is given as 3.8.

$$x_{t+1} = ax_t(1 - x_t) \tag{1}$$

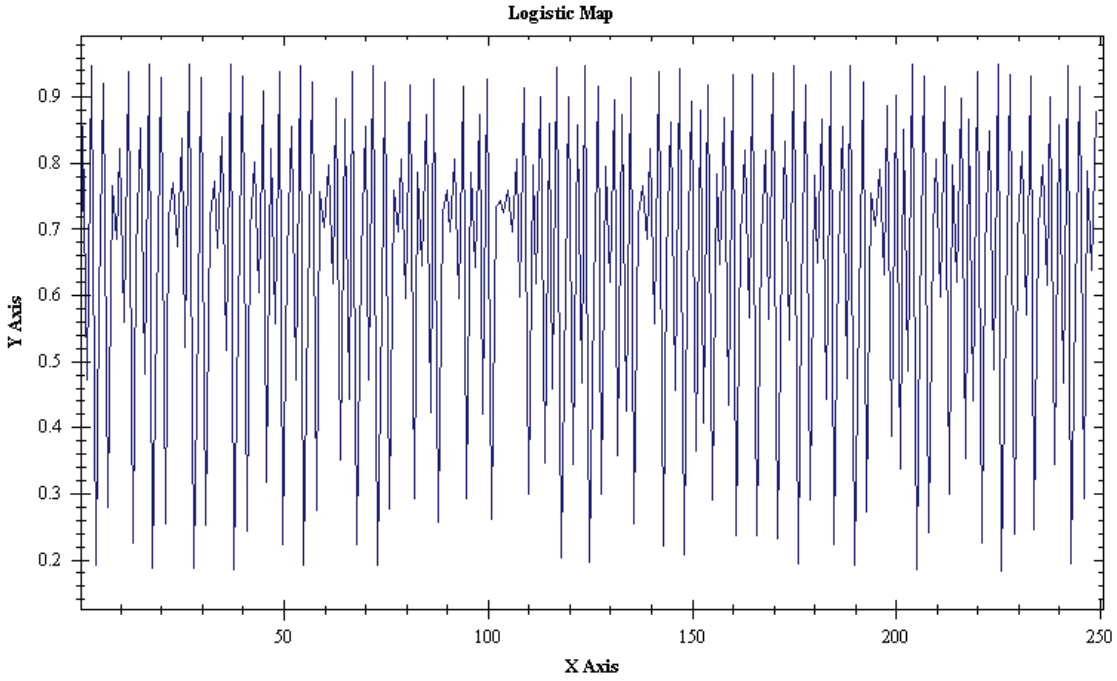

Figure 1. Logistic equation with *a* = 3.8

As can be clearly seen in Fig. 1, the data for the Logistic equation with *a* = 3.8 is very complex and has a similar appearance to random data or data with large amounts of noise. A close examination of the data in Fig. 1 indicates that the dataset is not well represented as periodic or random. It is also clear that the system described in Eq. 1 does not attend asymptotically to a fixed point for the given value of *a*. This space of non-random, non-periodic and non-fixed data is the realm of deterministic chaos.

The Logistics equation is an interesting map in that chaos is not always present in the system. As the values of the system increase from 1 to less than 3, the system will asymptotically approach a fixed value. Increasing the value of *a* from 3 to less than 3.57 causes the system to oscillate, first around two values and then four and so forth. Chaos sets in at $a = 3.57$. The nature of these changes is due to bifurcations experienced in the system. These bifurcations represent a transitional route to chaos which has been traditionally termed the *period doubling route to chaos*. Fig. 2 plots a bifurcation diagram for this route to chaos by representing the value of the parameter *a* on the horizontal axis while the vertical axis represents the distinct values of the system for each value of *a*.

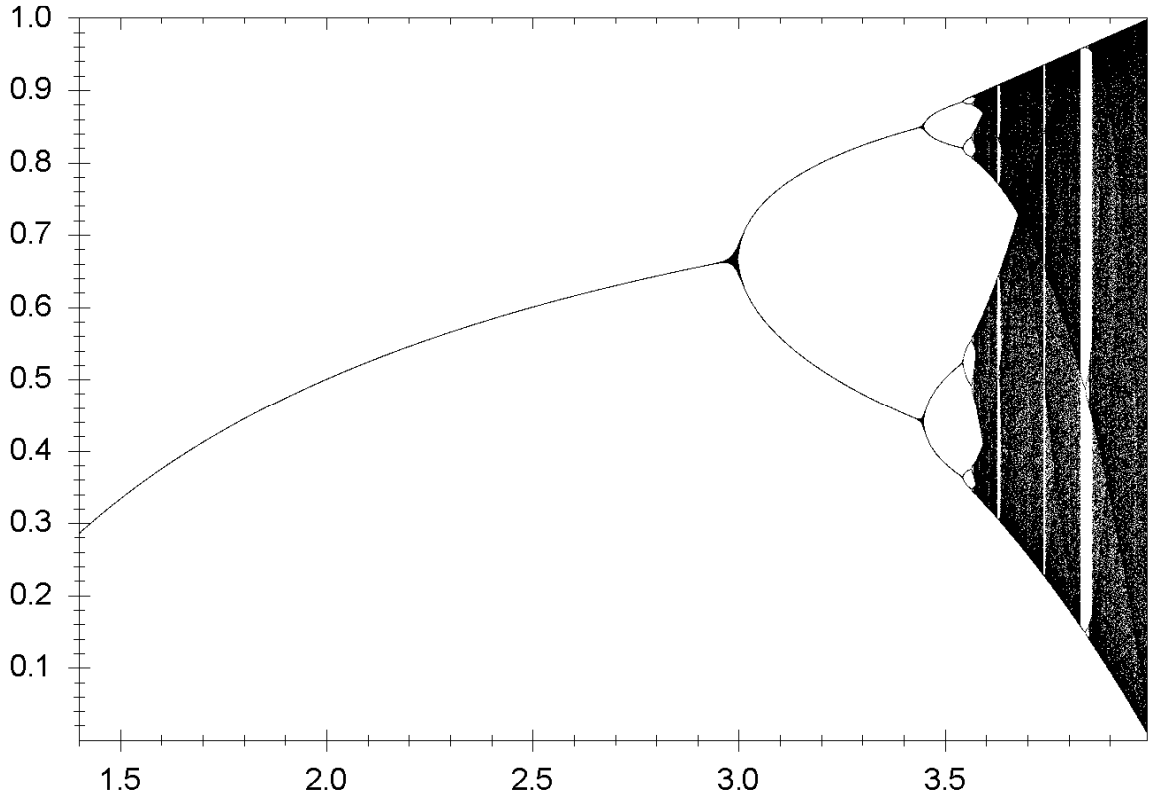

Figure 2. Bifurcation diagram of the Logistics equation.

Deterministically chaotic systems exhibit a number of distinct characteristics not normally associated with other systems. One important characteristic is sensitivity to initial conditions. Glasner *et al* [14] offered a topological definition for this characteristic. Call a pair (*X, T*), where *X* is a *d* dimensional compact metric space and *T* is a continuous map from *X* to itself, a system. Then, a system (*X, T*) has sensitive dependence on initial conditions if there exists an $\epsilon > 0$ such that for every $x \in X$ and every neighborhood U of x, there exists $y \in U$ and $n \in \mathbb{N}$ with $d(T^n x, T^n y) \geq \epsilon$.

The definition given in [14] simply states that if two points in a system are initially close to one another, they will be significantly far apart after *n* iterations of the system. Regardless of the selection of the initial configuration of points this phenomena holds true, given the system is experiencing sensitive dependence on initial conditions.

The characteristic of sensitive dependence to initial conditions is easily observed in the Logistic equation. Fig. 1 displays an instance of the Logistic equation with $a = 3.8$. Simply adding an extremely small value to a (e.g. 0.001) produces a completely different dataset as illustrated in Fig. 3.

The data in Fig. 1&3 may not be easily discernable as representing differing systems. However, once the two systems are systematically compared the difference is evident. Fig. 4 displays the absolute difference between the two systems by plotting the absolute difference of the data for the two systems at each time increment. As can be seen in Fig. 4 there is a significant deviation between the datasets of the Logistic equation for $a = 3.8$ (e.g. Fig. 1) and $a = 3.8001$ (e.g. Fig. 3).

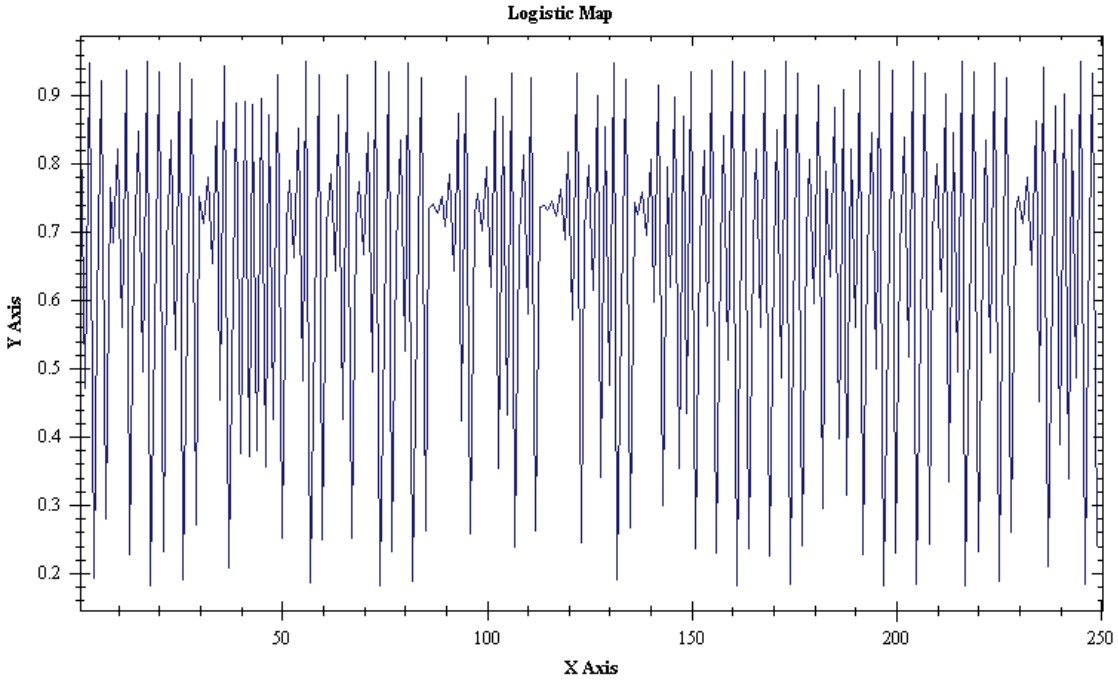

Figure 3. The Logistic equation with $a = 3.8001$

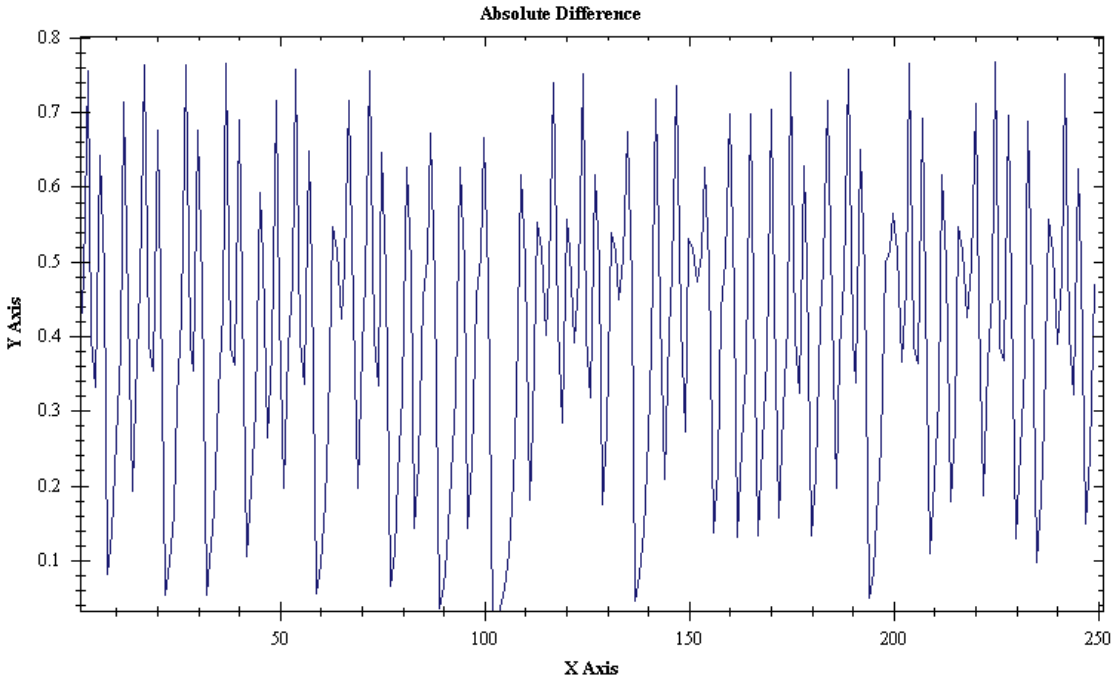

Figure 4. Deviation between two instances of the Logistic equation.

To solidify the difference between the two representations of the Logistic equation two error metrics are calculated. Error, for the present purpose, is simply a representation of the absolute difference between the two systems. The total absolute error (TAE), given in Eq. 2, and the mean absolute error (MAE), as given in Eq. 3, are the calculated metrics. The TAE for the two systems is 100.6307 while the MAE is 0.4025. This difference is significant and illustrates clearly the characteristic of sensitive dependence to initial conditions for chaotic systems. Even slight rounding errors can produce significantly different system dynamics as was discovered by Lorenz in 1963 [9].

$$TAE = \sum_{i=1}^{N} |x_i - x'_i| \tag{2}$$

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |x_i - x'_i| \tag{3}$$

Another characteristic property of deterministically chaotic systems is that of ergodicity. Ergodic systems are those which hold to the Poincare Recurrence Theorem [15] which states, in an interval of significant duration, a system will return to states that are very similar to previous ones. This property is illustrated in the Logistic equation of Fig. 1&3 and represents the quasi-periodicity previously mentioned. To see this property in a clearer manner the data used in Fig. 1 is used to generate the phase portrait in Fig. 5. A phase portrait is a geometric representation of the trajectories of a dynamical system in the phase plane [12]. In the case of Fig. 5, the phase portrait is generated by plotting $x(t)$ versus $x(t+1)$ in a method known as delay embedding which shall be discussed shortly. The ergodicity of the Logistic equation is easily discerned in such a plot. The system

produces dynamics whose trajectories visit very similar spaces with each orbit. The ergodic property of chaotic systems will be discussed at great length and used in assisting the understanding of normality for such systems in Chapter 3.

These chaotic characteristic traits are summed up by Zhang *et al* [2] as discussed in the introduction of this thesis. Implicit is the fact that systems which do not exhibit these characteristics are not deterministically chaotic. Unfortunately, the converse to this statement is not necessarily always true. As such there exist some distinct techniques for determining if a system is deterministically chaotic or not. These techniques are given in Section 2.2.

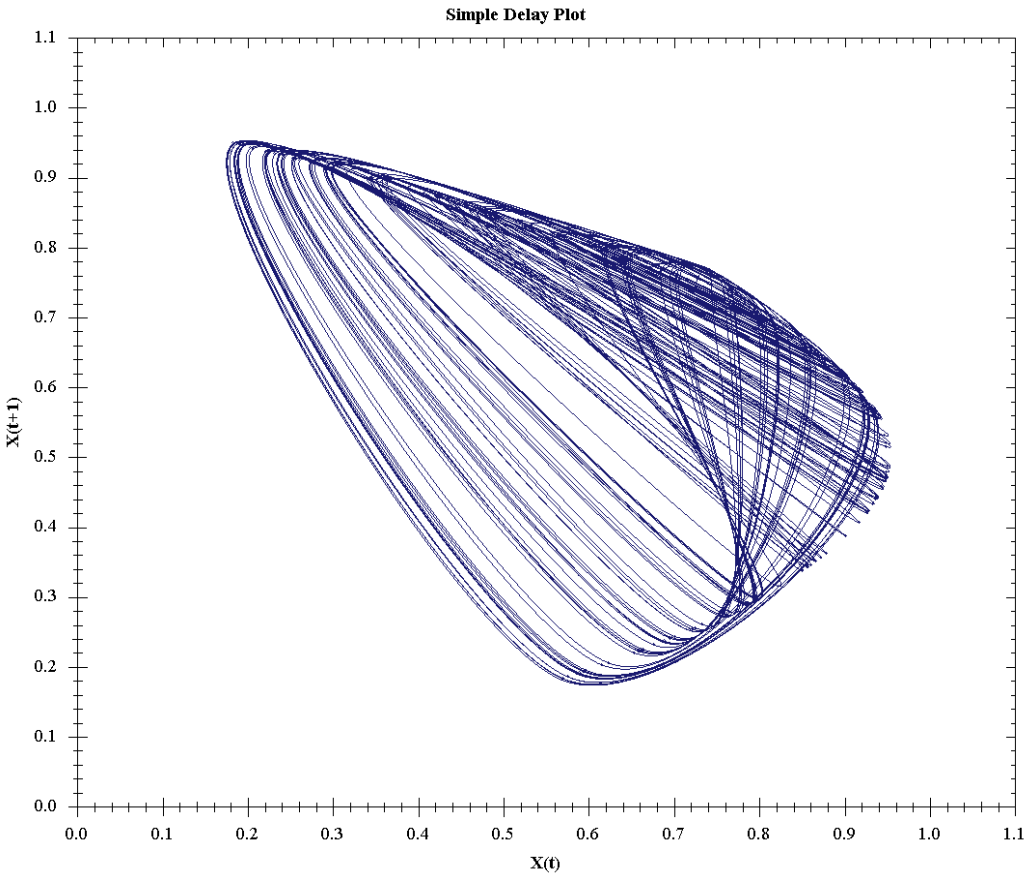

Figure 5. Phase portrait of Logistic equation given in Fig. 1.

It is an intuitive reaction to assume that the data which is transformed to generate the simple delay plot of Fig. 5 is from a periodic or random system and contains a great deal of noise. However, this reaction is obviously incorrect as the plot of Fig. 5 illustrates. Analysis/observation tools in common use, such as the Fourier analysis and Wavelet analysis become of little value for such systems as shall be shown in the next section.

2.1.1. Analysis/Observation Techniques

Many domains are tasked with the analysis/observation of signals. Engineering, physics, chemistry and medical are a few examples of such domains. Due to the ubiquitous nature of signal analysis, there exists a good set of analysis tools which have been developed. Two of the most prominent analysis tools are the Fourier and Wavelet analysis techniques. While these techniques have proven themselves valuable for the analysis of many systems, they are less advantageous when the system under question is chaotic.

This section briefly describes the Fourier and Wavelet analysis techniques as well the delay embedding technique for chaotic systems. For a complete treatment of Fourier analysis the reader is referred to [16] and for a complete treatment of Wavelet analysis the reader is referred to [17]. A comparison of the techniques used to analyze the Logistic equation is presented to further solidify the differences between the techniques. It is hoped that at the end of this section the reader will have an appreciation for why standard analysis techniques fail to accurately and helpfully describe the dynamics of a chaotic system.

2.1.2 Fourier Analysis

Named after Joseph Fourier, the Fourier analysis assumes that general functions may be represented as sums of simpler trigonometric functions [16]. Two types of Fourier analysis are common – continuous time and discrete time. For the purposes of this paper only the discrete time Fourier analysis is considered, however it has been shown that the results presented here hold for continuous Fourier analysis.

The discrete time Fourier analysis considers a signal $x(n)$ which is periodic with period N, that is to say that $x(n + N) = x(n)$. Given this assumption, the signal $x(n)$ can be represented by a series of N harmonically related functions given by Eq. (4) where $j^2 \equiv -1$.

$$e^{j2\pi kn/N}, \; k = 0, 1, \ldots, N-1 \tag{4}$$

Then the signal $x(n)$ can be expressed as in Eq. (5) where c_k is given in Eq. (6) [18].

$$x(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N} \tag{5}$$

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \tag{6}$$

As Gao *et al* [18] stated, it is easy to see that c_k is also periodic in N. Thus, the sequence $|c_k|^2$ for $k = 0,1, \ldots, N\text{-}1$ is the distribution of power as a function of frequency. This is often called the *power density spectrum* of the signal and is a tool often used for the analysis of signals. The power density spectrum indicates frequencies which are most prominent in the signal.

To assist in the understanding of how the power density spectrum is useful in understanding the dynamics of signals the following example is supplied. Consider the Logistics equation with the parameter a = 3.50. As was eluded to earlier, this value of a causes the Logistics equation to be periodic. This periodicity is illustrated in the plot of the raw data in Fig. 6. It is easy to see that there are multiple levels of the periodicity in Fig. 6. It is also clear that the Fourier analysis' power density spectrum effectively captures these levels as shown in Fig. 7.

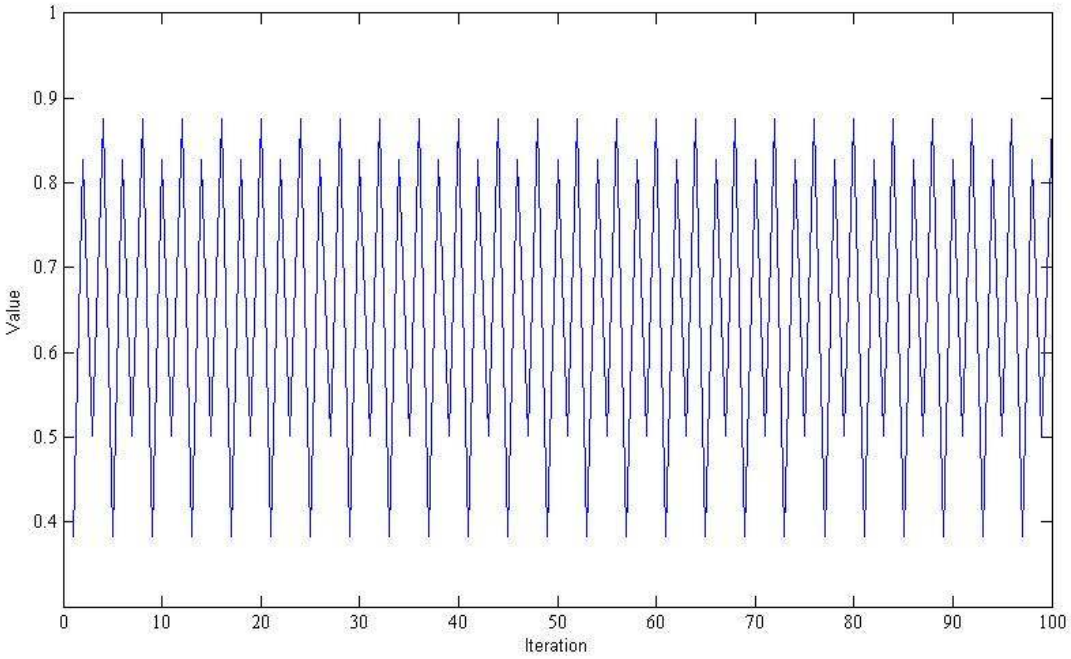

Figure 6. Raw data for the Logistics equation with a=3.50.

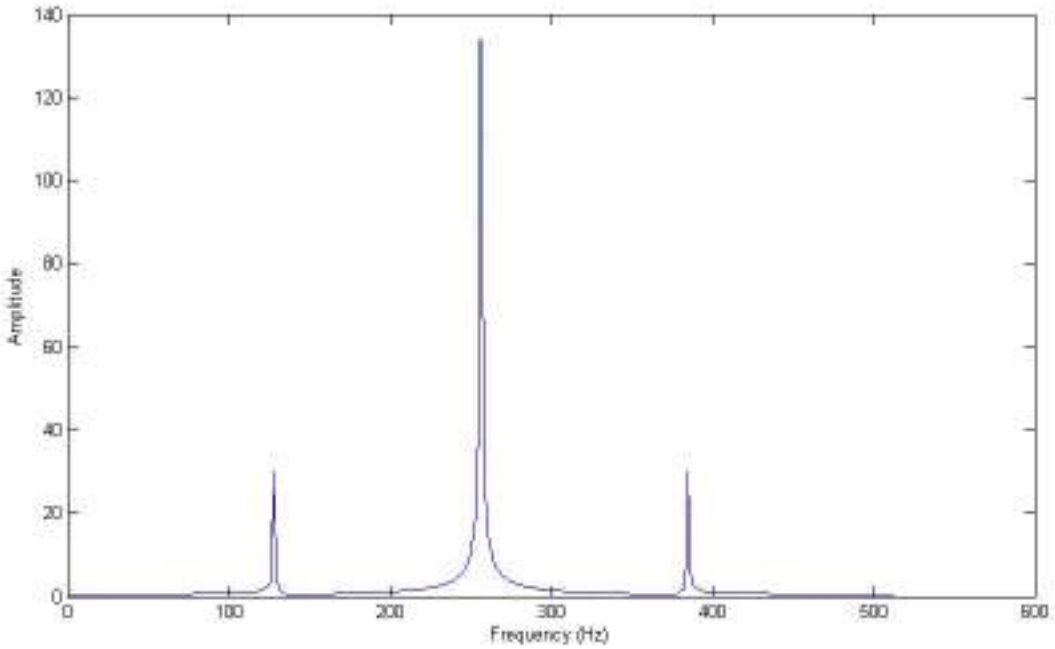

Figure 7. Fourier power density spectrum for the raw data in Fig. 6.

Often the Fourier analysis of a signal is considered to be a combination of the trigonometric functions of sine and cosine. It is assumed that the real signal, buried within the noise of the data, is some representation of this combination if the signal contains periodicity in *N*. One should also note that the transformation of the Fourier analysis in usually invertible. All of these features, as well as the fact that the power density spectrum is effective a capturing the periodic amplitudes of the system, make the Fourier analysis a desirable technique.

2.1.3 Wavelet Analysis

In a manner akin to the Fourier analysis technique, Wavelet analysis considers that the signal $x(n)$ can be represented as a sum of specific functions. The Wavelet analysis uses a mother wavelet function for its analysis similar to the Fourier analysis' use the sum of trigonometric functions. Unique to the Wavelet analysis though is the fact

that these mother wavelets can be of any nature, not just trigonometric, so long as the sum of the points in the mother function is zero [17]. Two common mother wavelets, the Harr wavelet and the Daubechies 4 wavelet, are shown in Fig. 8.

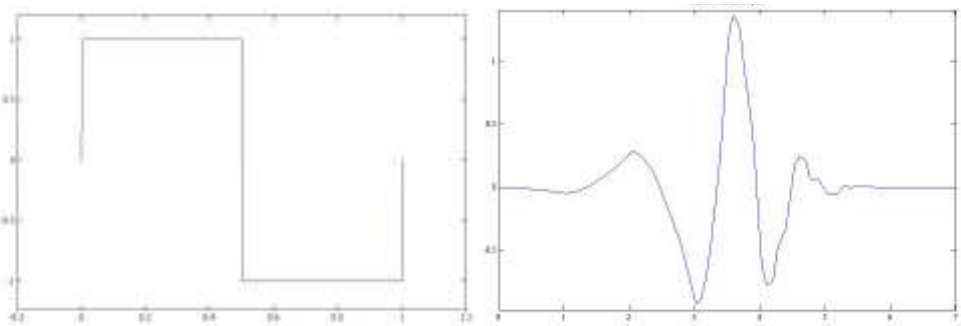

Figure 8. The Harr (left) and Daubechies 4 (right) mother wavelets.

The Wavelet analysis uses a scaled and shifted version of the mother wavelet, $\psi(t)$ to generate approximations and details of the signal $x(n)$. The performance of the Wavelet analysis results in the generation these two, the approximation and details, series at multiple levels of decomposition. When properly combined the approximations and details fairly accurately recreate the signal. To illustrate this, Fig. 9 plots the approximation and details of the first level of a Wavelet analysis on the raw data used in Fig.6 using the common Daubechies 4 wavelet.

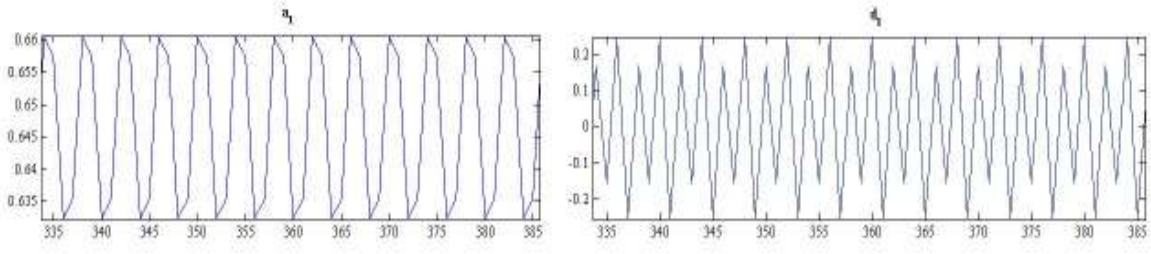

Figure 9. First level Wavelet analysis of the data in Fig. 6 using the Daubechies 4 wavelet. The approximations of the signal are given on the left and the details of the signal are given on the right.

It is easy to see from Fig. 9 that the Wavelet analysis is able to correctly determine that the signal is periodic, as in the approximation, and that it is periodic with a period of 4 for the Logistics equation with $a = 3.50$. The fact that the Wavelet analysis can use many variants of mother wavelet functions makes it a desirable tool for signal analysis. However, as shall be shown next, both the Wavelet and the Fourier analysis are not as effective as the Embedding Phase Space technique for the observation of the dynamics of chaotic signals.

2.1.4 Technique Comparison

This subsection compares the techniques of subsections 2.1.2 and 2.1.3 to the technique of the simple delay plot previously shown and often referred to as Embedding Phase Space. The techniques are compared using the Logistics equation. Due to the ability of the Logistics equation to represent both periodic and chaotic signals, depending on the value of the parameter a, it is an appropriate system to compare the three techniques adequately.

As was seen in the bifurcation diagram of Fig. 2, the Logistics equation undergoes a period doubling route to chaos. Therefore, there are values for a in Eq. (1) for which the Logistics equation represents a single fixed point system, a period two periodic system, a period four periodic system and so forth, until $a = 3.57$ at which time the system becomes chaotic. Here we will compare the three techniques for three values, a = {3.50, 3.56, 3.79}. For each value of a discussion and illustrations will be presented which will highlight the observational differences between the standard Fourier and Wavelet techniques and the Embedding Phase Space technique. Throughout this subsection the Daubechies 4 wavelet is used as the mother wavelet for the Wavelet analysis.

Consider first, the Logistics equation of Eq. (1) where $a = 3.50$. A sample of the output of the Logistics equation, so defined, is given in Fig. 10. As can be seen in Fig. 10, the Logistics equation assumes a periodicity with period $N = 4$ for $a = 3.50$. Due to this periodicity, the Fourier and Wavelet analysis techniques aptly describe the dynamics of the system as illustrated in Fig. 11-12. The Embedding Phase Space technique also does well at describing the system as a system which traverses between four distinct values as shown in Fig. 13. For this sub-section the delay value of the Embedding Phase Space is given as 1 and the dimension of the space is set to 2.

Now consider the system where the Logistic equation has the value of a increased to 3.56 whose raw data is given in Fig. 14. This system is also a periodic system, albeit with a period of $N = 8$. Again, the Fourier and Wavelet analysis techniques are adequate for describing the system as can be seen Fig. 15-16. The Embedding Phase Space technique also illustrates its prowess by showing that the system oscillates between eight fixed points as given in Fig. 17.

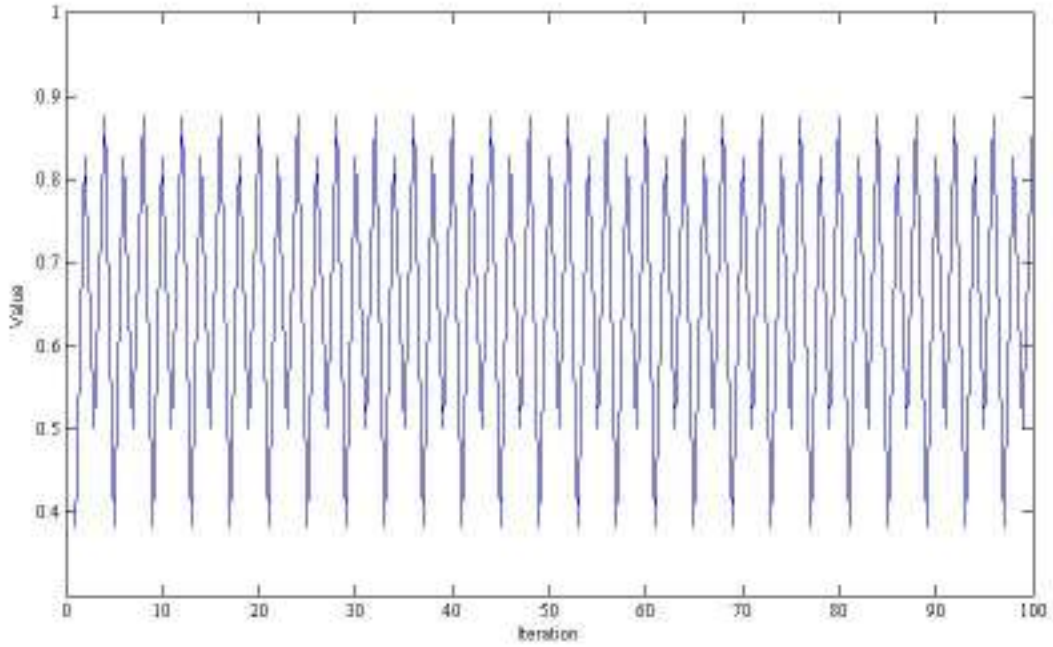

Figure 10.Sample output of the Logistics equation with a = 3.50.

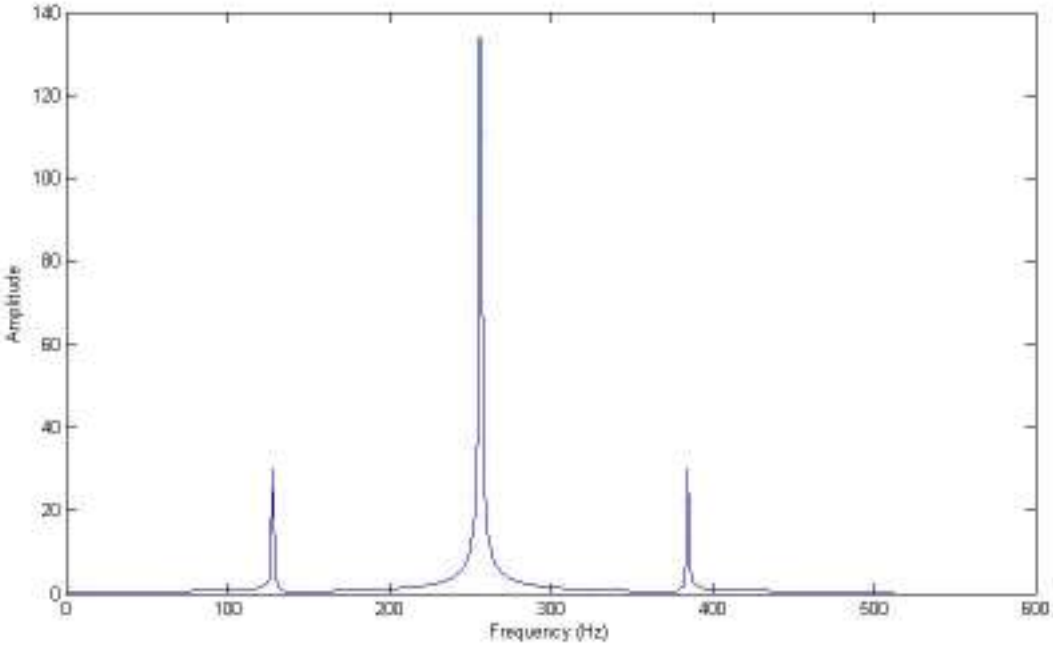

Figure 11. The power density spectrum of the Fourier analysis of the data in Fig. 10.

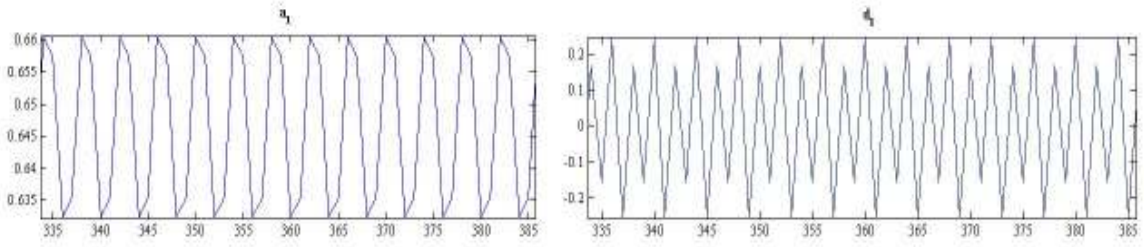

Figure 12. The first level of the Wavelet analysis of the data in Fig. 10.

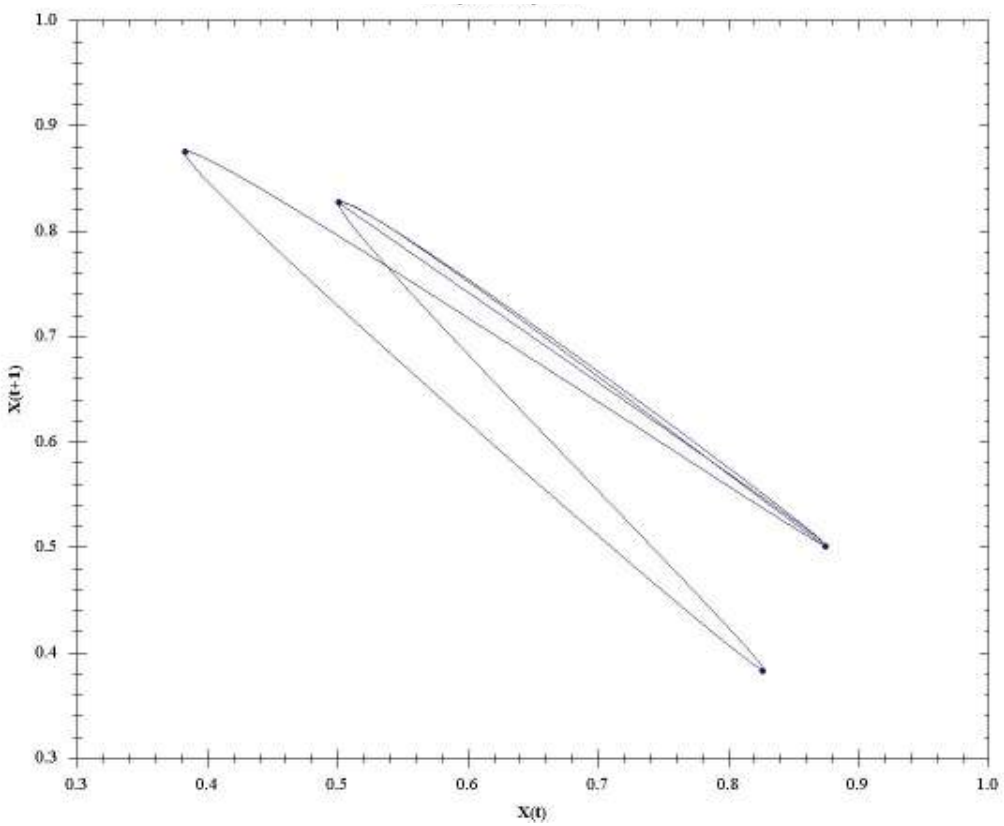

Figure 13. The Embedding Phase Space for the data in Fig. 10.

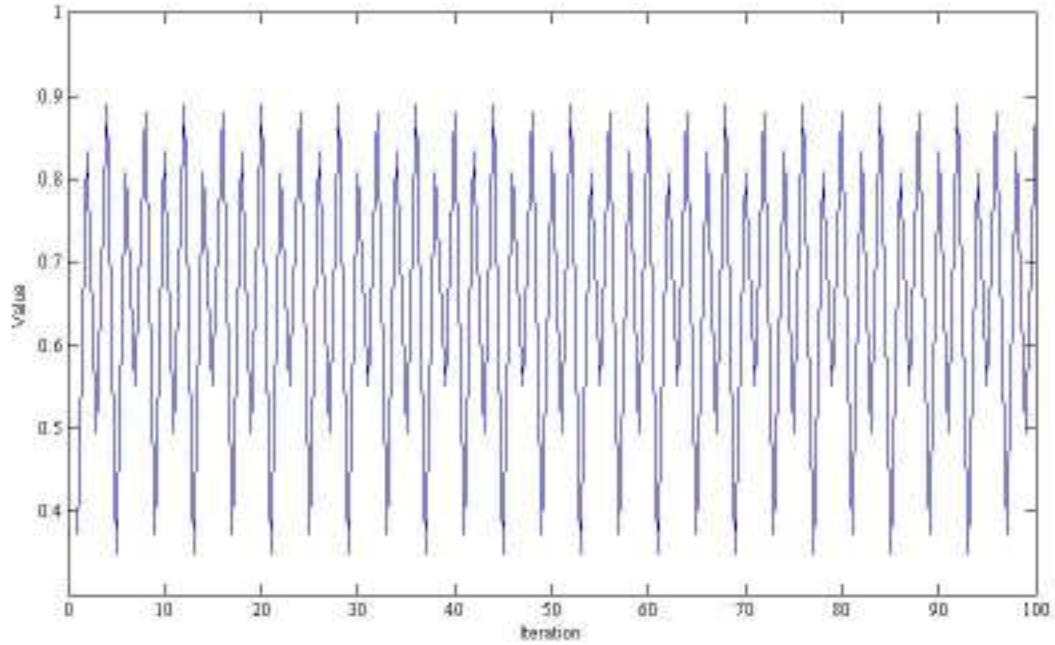

Figure 14. A sample output of the Logistics equation with a = 3.56.

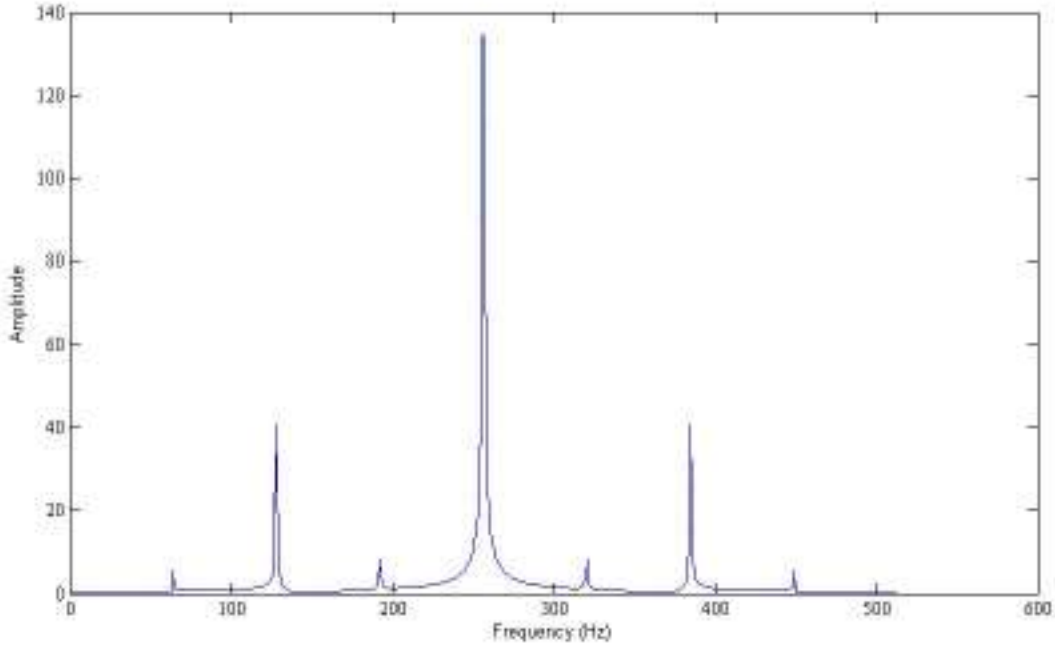

Figure 15. The Fourier power density spectrum for the system whose data is given in Fig. 14.

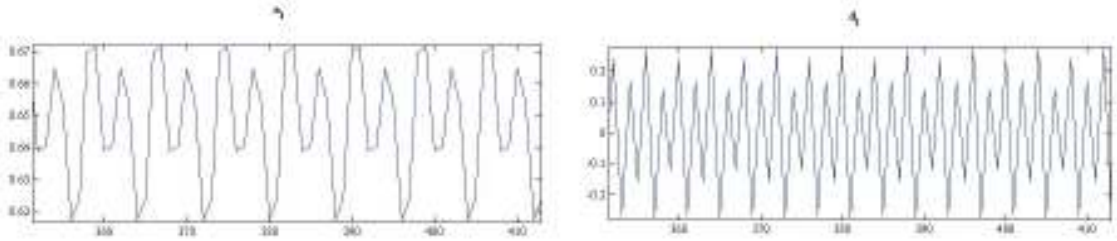

Figure 16. The first level of the Wavelet Analysis of the system whose data is given in Fig. 14.

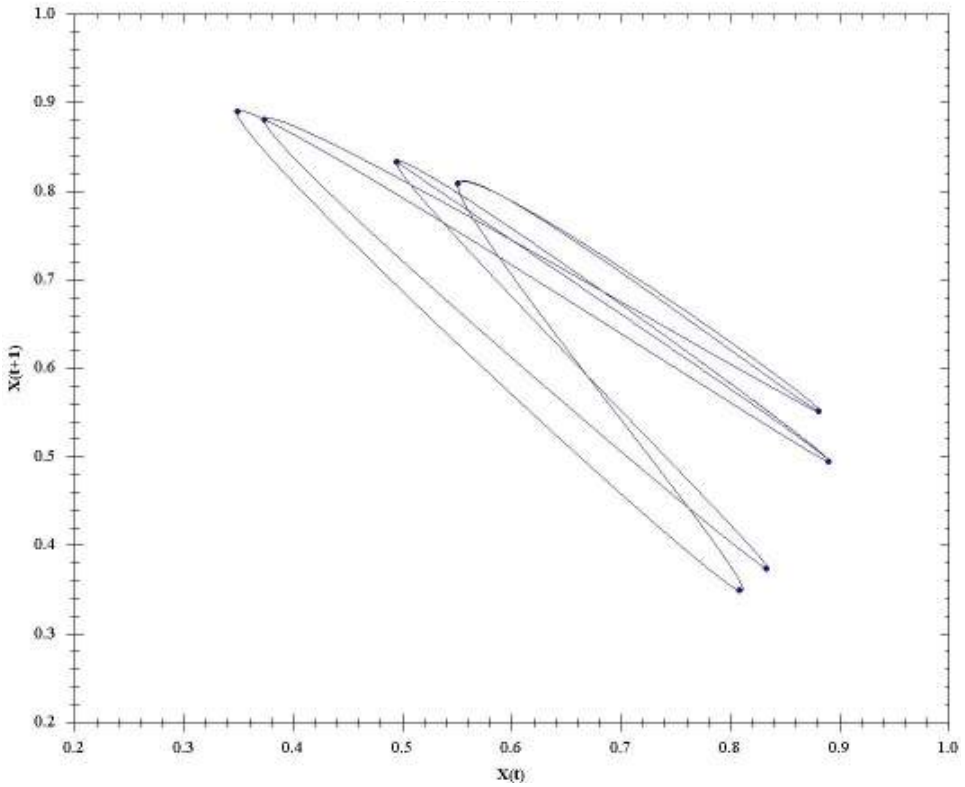

Figure 17. The embedding phase space for the system whose data is given in Fig. 14.

Thus far, all of the techniques have performed well on data from the Logistics equation. Each represents the dynamics of the system in their own manner. However, as the Logistics equation is pressed to the level of chaos, with a = 3.79, a weakness of the two standard techniques is noticed. A sample of the output for the Logistics equation with a = 3.79 is given in Fig. 18. It is easy to see that this data is no longer periodic, or at least

not periodic within a reasonable length period. In fact, this data is deterministically chaotic.

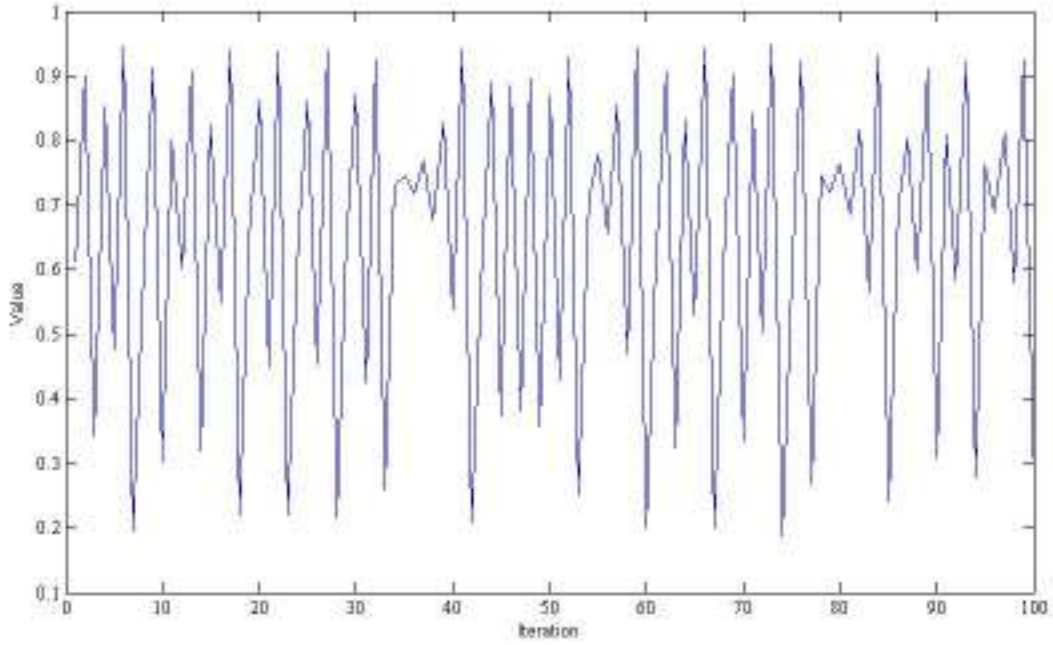

Figure 18. Sample output for the Logistics equation with a = 3.79.

Without inherent periodicity in this version of the Logistics equation the standard analysis techniques of the Fourier and Wavelet analysis are not as beneficial at clearly representing the system dynamics. As can be seen in Fig. 19-20 the standard techniques offer little information as to the true dynamical nature of the system. In fact, taken to the fifth level, the Wavelet analysis is incapable of representing the dynamics in a legible format as shown in the expanded analysis plots of Fig. 20.

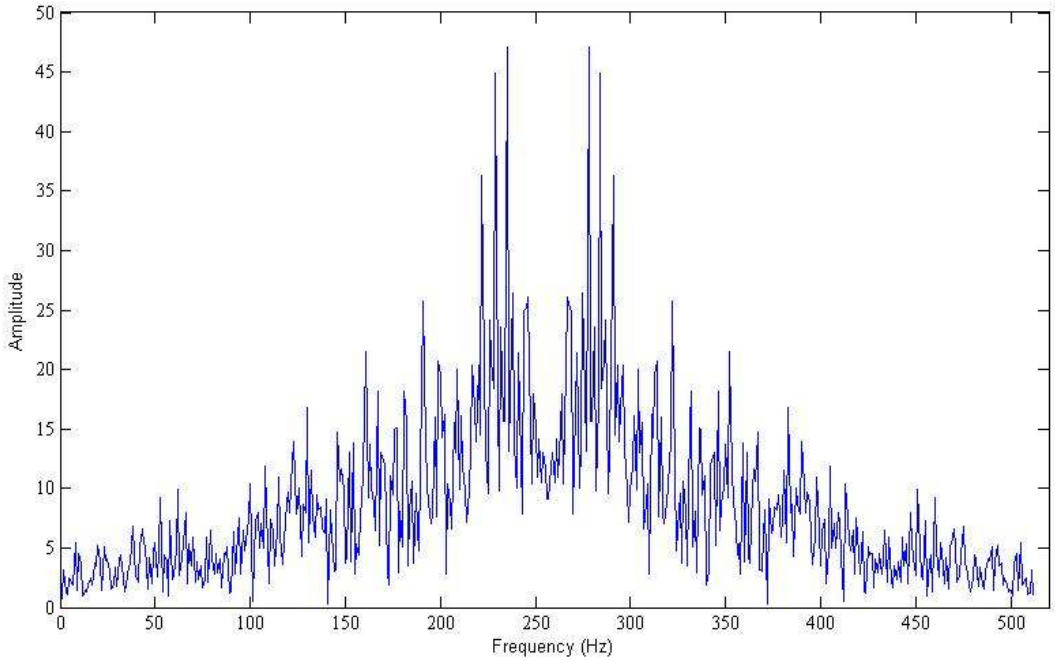

Figure 19. The Fourier power density spectrum of the Logistics equation with a = 3.79.

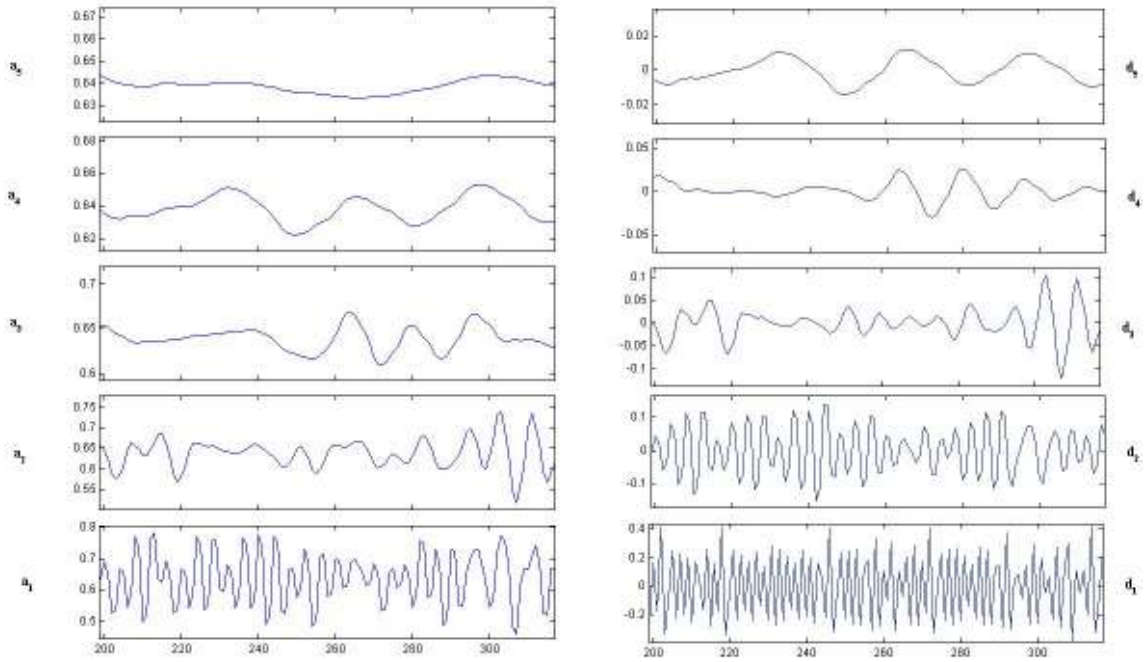

Figure 20. Five levels of Wavelet analysis for the Logistics equations with a = 3.79.

It is easy to see that the Fourier analysis of Fig. 19 does not represent the dynamics of the system well. In fact, it offers very little useful information, aside from asymptotic boundaries. The same can be said for the Wavelet analysis of Fig. 20. In that analysis, the signal is not well approximated or detailed. Further, little useful information, concerning the system's dynamics, can be gleaned from the analysis plots. The Embedding Phase Space technique however does well at representing the dynamics of the system and at highlighting the ergodicity present in the system as can be seen in the delay plot of Fig. 21.

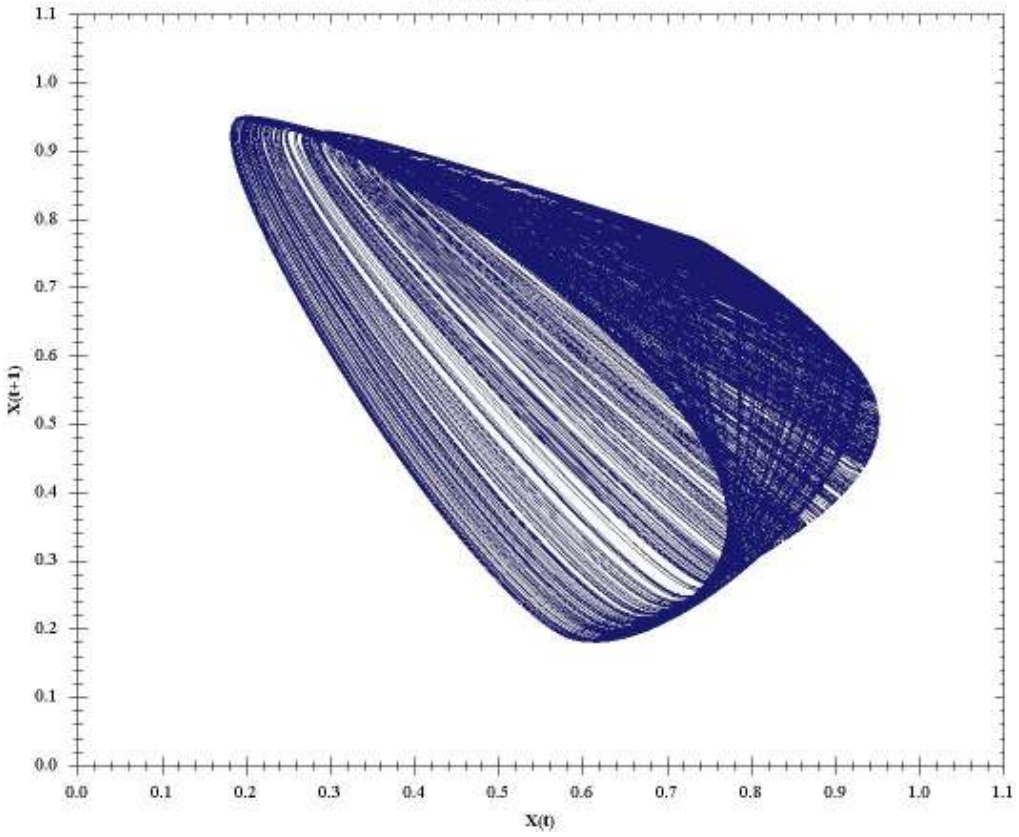

Figure 21. Embedding Phase Space plot of the Logistics equation with a = 3.79.

The chaotic data, which has been used in the above comparisons, varies greatly from data which is periodic, stochastic or attends to a fixed point. The constant revisiting of regions of phase space, as was shown the Embedding Phase Space plot of Fig. 21, cause the system to appear either random or periodic with a great deal of noise. Due to this difference in the nature of the data, the Fourier and Wavelet analysis methodologies suffer.

From the above examples, it is easily seen that for chaotic data the Embedding Phase Space technique offers a much clearer view of the dynamics inherent in chaotic systems than is given by the standard analysis techniques. Further, it was shown that the Embedding Phase Space technique performs equally well at representing systems which are not chaotic but rather are periodic. As such, the Embedding Phase Space technique of system representation is a more desirable analysis/observation tool than the standard frequency analysis techniques for chaotic systems.

Using the standard analysis techniques to analyze chaotic data would lead one to assume that the system is noisy or random. Obviously, this is an incorrect assumption and would account for the issues that have been experienced in the past with poor performance in prediction and normality detection on such systems. Thus, it is critical that such systems be considered in the chaotic realm rather than the standard stochastic domain which is so favored by many researchers.

2.2 Determination of Deterministic Chaos from System Data

This section presents methodologies for determining whether or not a system, represented by some dataset, is deterministically chaotic or not. Most often the datasets that will be considered for analysis come from time dependent sources. Rarely in real

world systems does the data come from a non-time dependent source given correct collection methodologies. As such, this section's presentation is concerned primarily with time series datasets collected in discrete intervals. However, to be complete a brief discussion will first be presented concerning systems whose equations of motion are known apriori of the analysis.

There exist two types of chaotic systems - maps and flows. Maps are mathematical systems which model variables as they change over discrete steps in time [19]. Maps are often referred to as difference equations, iterated maps or recursion relations. Flows are systems of differential equations which, for the study herein, produce chaotic properties[1]. The main difference between maps and flows is continuity. Flows are continuous while maps are generated in discrete time intervals. This continuity obviously effects whether or not the system is smooth. The Logistic equation of Eq. (1) is a sample of a one dimensional map. The time-series of the Logistic equation indicates that points in successive time intervals vary greatly about the map. This is a result of the discontinuity discussed. The Lorenz system, given in Eq. (7), is an example of a chaotic flow as shown in Fig. 22. The time-series of the Lorenz system is continuous and as such points in successive time intervals are extremely close to one another.

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y - x) \\ \frac{dy}{dt} &= x(\rho - z) - y \\ \frac{dz}{dt} &= zy - \beta z \end{aligned} \tag{7}$$

[1] Flows whose dimensions are less than three cannot produce chaos; three or more dimensions are required for a flow to be chaotic.

Chaotic flows and chaotic maps adhere to the principle that points initially close to one another, albeit on different orbits (trajectories), rapidly diverge as their orbits progress. The rate of divergence in a chaotic system is measured through the use of Lyapunov exponents [20].

The Lyapunov exponents of a system indicate a great deal concerning the dynamics of that system. A positive Lyapunov exponent indicates divergence of points initially close to one another in the direction of the Lyapunov exponent (I.E. chaos) while a negative exponent indicates convergence [21]. Further, Lyapunov exponents indicate the predictability of the system in question [22]. The larger the absolute value of the Lyapunov exponent the smaller the window of accurate predictability becomes. Calculation of the Lyapunov exponents of a system is straight forward when the equations of motion are known.

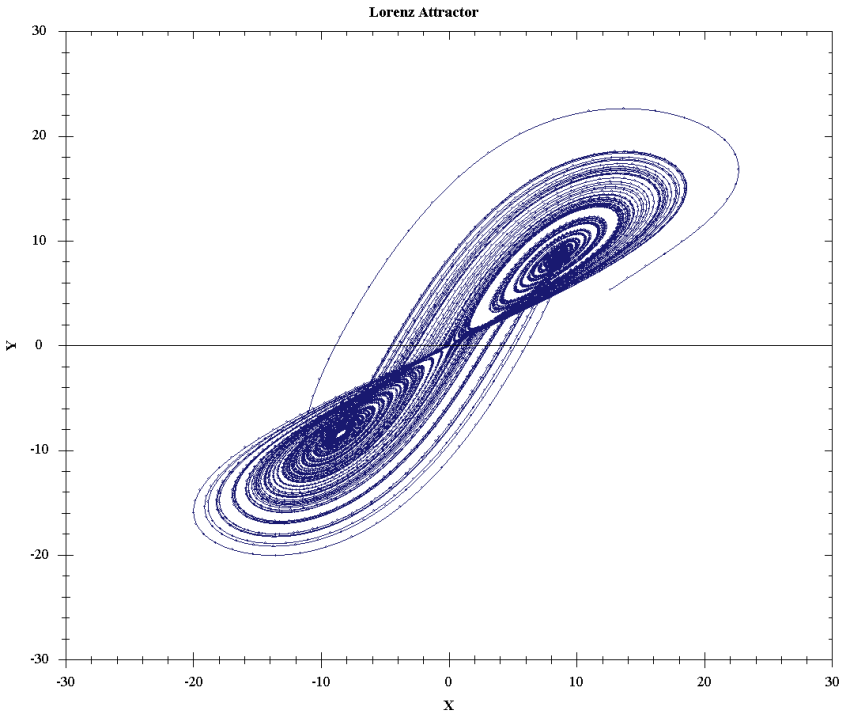

Figure 22. Lorenz Attractor plotted for x and y.

The calculation of the Lyapunov exponents for difference equations is a simple iteration of the map through the discrete time steps and a measurement of the divergence/convergence of points initially close together. Consider two points on a map, initially close together, X_0 and $X_0 + \Delta X_0$, then after one iteration of the map the points are separated as in Eq. (8).

$$\Delta X_1 = f(X_0 + \Delta X_0) - f(X_0) \simeq \Delta X_0 f'(X_0) \tag{8}$$

The local Lyapunov exponent λ at X_0 can be defined as $\lambda = ln\left|\frac{\Delta X_1}{\Delta X_0}\right|$ [14]. The local Lyapunov exponents of a map are the same value as the eigenvalues of that map [12]. The local Lyapunov exponent is helpful in determining the dynamics of a region of phase space while the global Lyapunov exponent is indicitive of the entire space. The calculation of the global Lyapunov exponent, for maps, is simply the average of the local exponent over many iterations.

Calculation of the Lyapunov exponents of a flow proceeds in a similar manner. Consider two points, initially close to one another, on a flow, x_0 and $x_0 + \Delta x_0$. Then after a small amount of time, Δt, the points are seperated as given in eq. (9). As with maps, the local Lypunov exponents are equal to the eigenvalues of the flow. Further the sum of the Lyapunov exponents are equal to the determanit of the Jacobian matrix [23].

$$\Delta x_1 = \Delta x_0 + [f(x_0 + \Delta x_0) - f(x_0)]\Delta t \simeq \Delta x_0[1 + f'(x_0)\Delta t] \tag{9}$$

Calculation of the Lyapunov exponents of a time-series, without the aid of knowing the equations of motion apriori, is a more complicated task. The calculation of this important measure for a time-series dataset requires the calculation of a time delay and an embedding dimension, as in the embedding phase space of the previous section. These metrics are used to transform the time-series dataset into embedding phase space, in which the Lyapunov exponents of the system may be calculated. The remainder of this section discusses the detection of chaos in a time-series data set, the transformation into embedding phase space and the calculation of Lyapunov exponents for such a system.

Consider an *n* dimensional system *X* composed of time increments *T* s.t. $t \in T$ for some discrete measure of time; then $\vec{\boldsymbol{x}}(t) = \{x_1(t), x_2(t), \dots, x_n(t)\}$ represents the state of the system *X* at time *t.* A dataset $D = \{\vec{\boldsymbol{x}}(0), \vec{\boldsymbol{x}}(1), \dots, \vec{\boldsymbol{x}}(i)\}$ representing the system *X* for *i* iterations is possibly deterministically chaotic if $\vec{\boldsymbol{x}}(j)$ is dependent in some manner upon $\vec{\boldsymbol{x}}(k)$ for $j = k + \Delta t$. Using this definition of a system it is possible to determine if the system *X* is non-chaotic through a simple null hypothesis test [3]. To perform this test an ensemble of surrogate datasets are formed through the random reordering of the original dataset *D*. An invariant metric, such as the autocorrelation function, is used to measure the original dataset and the ensemble of surrogate datasets. If the invariant metric results in similar measures for the ensemble and the original dataset then the original dataset is not time dependent and therefore is not from a deterministically chaotic system. However, if there is a significant difference, to the 95% significance level,

between the surrogate dataset and the original dataset then the original dataset may be chaotic[2].

The literature reports variants for null hypothesis testing of deterministically chaotic systems. The simplest of the metrics used for testing is the autocorrelation function as presented by Kantz & Schreiber [3]. The autocorrelation function is the measure of the correlation between two values of the same variable at different times. Given measurements $\vec{x}(t) = \{x_1(t), x_2(t), \dots, x_n(t)\}$ for time $t \in T$ the autocorrelation function is then defined as Eq. (10).

$$r = \frac{\sum_{i=1}^{N-k}(x(i)-\bar{x})(x(i+k)-\bar{x})}{\sum_{i=1}^{N}(x(i)-\bar{x})^2} \tag{10}$$

Other methods to form the surrogate ensemble and other invariant metrics have been suggested as well. Schriber *et al* [24] suggested the use of a simple iteration scheme for forming the surrogate dataset through a Fourier transformation, replacing the squared amplitudes and then transforming back again. The authors [24] made a claim for better surrogate formation through their methodology which results in more precise null hypothesis testing. Similarly, Theiler *et al* [25] proposed another amplitude adjusted Fourier transformation algorithm for surrogate formation. Steuer *et al* [26] suggest the use of the mutual information metric as the invariant metric with which to test the null hypothesis against the original and ensemble data. The mutual information metric describes numerically the information already known about a data point $x(t+1)$ from

[2] It is important to note that systems which do not reject the null hypothesis testing may still be derived from a non-chaotic dataset. Null hypothesis testing only definitively excludes the hypothesis that the system is chaotic; it does not however confirm that the system is chaotic.

the information given from the data point $x(t)$. Denote by p_i the probability that the signal (data) assumes a value inside the *i*th bin of a histogram for the dataset, and let $p_{ij}(x,t)$ be the probability that $x(t)$ is in bin *i* and $x(t+1)$ is in bin *j*. Then the mutual information for the time delayed dataset is given as Eq. (11).

$$I_\epsilon(t) = \sum_{i,j} p_{ij}(x,t) \ln p_{ij}(x,t) - 2\sum_i p_i \ln p_i \tag{11}$$

The methodology chosen for null hypothesis testing, in this thesis, uses the autocorrelation function as the invariant metric and surrogate formation using simple random reordering. This is due to the simplicity of the metric and formulation as well as due to the dataset sizes being addressed herein. Should extremely large datasets, or a large number of datasets be considered then the use of one of the above algorithms may be computationally preferred.

Once the null hypothesis testing has been performed, and the time series dataset has been determined to possibly be from a deterministically chaotic system, a more concrete metric must be used to determine succinctly whether or not the time series is truly chaotic. The best metric for this determination is estimate of the maximal Lyapunov exponent of the system. According to Wolf *et al* [20], calculation of a systems Lyapunov exponents has proven to be the most useful dynamic diagnostic for chaotic systems.

Estimation of the maximum Lyapunov exponent of a system is calculated in embedding phase space. Embedding phase space has been previously introduced but is now defined accurately. Embedding phase space is a metric space into which the data of the system is transformed to better represent the dynamics of the system. The theory of

this transformation was first formalized by Takens [27] in 1981 and requires the calculation of two parameters, a time delay and a minimum embedding dimension.

In order to transform the time series data into the embedding phase space the aforementioned two parameters must be calculated. These parameters are then used to define the points within the embedding phase space. Consider, for example, the two dimensional system of the Duffing Map, given in Eq. (12) where $a = 2.75$ and $b = 0.2$, whose x- axis' data is represented in Fig. 23 for the first 4000 points of the time series.

$$\begin{cases} x_{t+1} = y_t \\ y_{t+1} = -bx_t + ay_t - y_t^3 \end{cases} \tag{12}$$

This data series can be represented very effectively in embedding phase space through the use of a time delay and minimum embedding dimension. For this example, let the time delay, τ, equal 1 and the minimum embedding dimension, d, be 2. Then the embedded phase space can be recreated through the use of the embedding vector $\vec{x} = \{x(t-\tau), x(t)\}$ which produces the phase portrait of the dynamical system as shown in Fig. 24.

Consider a time series which contains a parameter x among other parameters. Then a two dimensional phase space time delay plot for the given variable x can be generated as x_t plotted versus $x_{t-\tau}$ where τ is a delay factor. The time delay, τ, must be selected carefully when reconstructing a phase space. Too small of a τ results in close points which cease to be independent while too large of a τ causes the spatial relationships to break down [3]. Often this is computed as the first coefficient r_τ of the autocorrelation function (e.g., Eq. (10) where $\bar{x}$ is the mean of the data) smaller than $1/e \approx$

0.368 [3], [27]. As an alternative, the first minimum of the mutual information function may also be used.

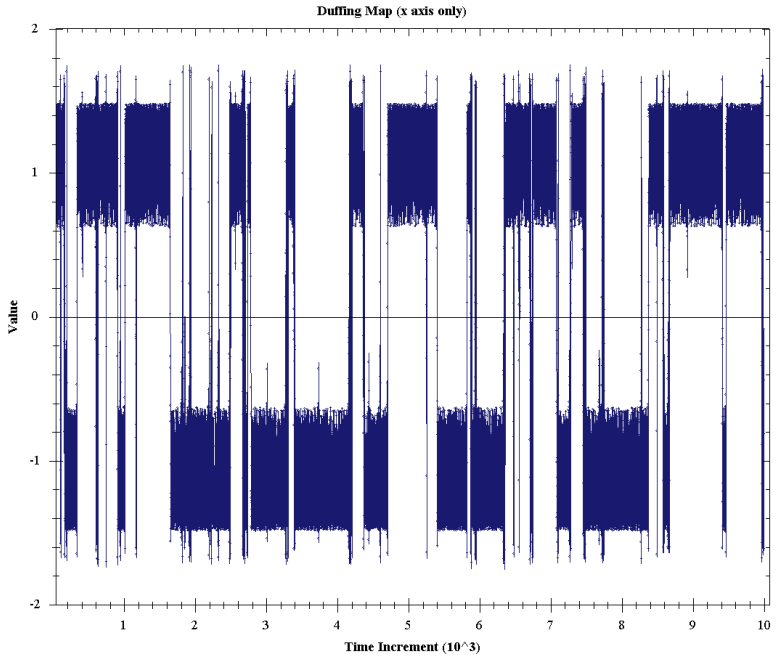

Figure 23. The *x* values of the Duffing Map for 4000 iterations.

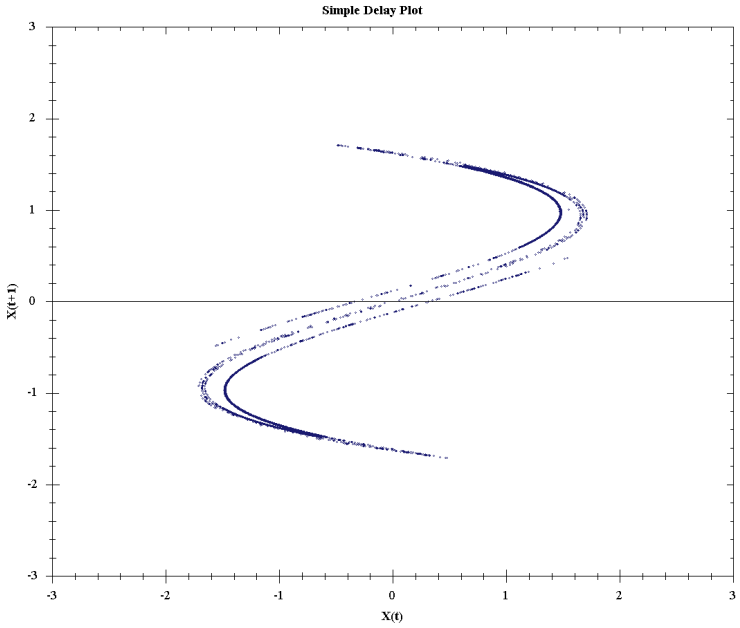

Figure 24. Phase portrait of the Duffing Map whose *x* values are given in Fig. 6.

Table 1. Duffing map autocorrelation.

Delay Factor (τ)	**Autocorrelation Coefficient (r_τ)**
1	0.026
2	0.136
3	0.1186

Once the delay factor τ has been calculated for the given dataset, the remaining parameter to calculate is the minimum embedding dimension for the reconstructed phase space. This dimension indicates the minimum number of dimensions required to reconstruct, or unfold, the scalar phase space back to a multivariate phase space that is representational of the original phase [28], [29].

Consider a scalar time series for a given variable x represented notationally as $x = \{x_0, x_{0+\tau}, x_{0+2\tau}, \dots, x_{0+(d-1)\tau}\}$ in embedding phase space with delay factor τ calculated from the autocorrelation of the variable. We wish to discover a minimal embedding dimension, *d* for which the scalar times series is most representational of the dynamics of the original phase space. To calculate the embedding dimension *d* the False Nearest Neighbors (FNN) methodology is used [28]. The FNN methodology iteratively considers a point $\boldsymbol{x}(n)$ and the point $\mathbf{x}^r(n) \in U(\mathbf{x}(r))$ which is the *r*th nearest neighbor of $\boldsymbol{x}(n)$, in the neighborhood, *U,* of points whose distance to $\boldsymbol{x}(n)$ is below a given threshold ϵ. The distance between these two points in *d* dimensional space is given in Eq. (13).

$$R_d^2(n,r) = \left(\sum_{k=0}^{d-1}[x(n+k\tau) - x^r(n+k\tau)]^2\right)^{\frac{1}{2}} \tag{13}$$

where τ is the time delay from the autocorrelation function [23]. Next the embedding dimension is increased from dimension d to dimension $d + 1$; such that, by time delay embedding, the (d+1)th coordinate is added to each of the vectors $\boldsymbol{x}(n)$. Once the dimension of the embedding space is increased the distance of each point $\mathbf{x}^r(n) \in U(\mathbf{x}(n))$ to $\boldsymbol{x}(n)$ is recalculated using Eq. (13). Those points whose distance to $\boldsymbol{x}(n)$ is larger than ϵ are removed from the neighborhood $U(\boldsymbol{x}(n))$. The removed points are considered false nearest neighbors to $\boldsymbol{x}(n)$ as they were nearest neighbors in the smaller embedding dimension but are not such when that dimension is increased. The minimum embedding dimension for the scalar phase space is given by the smallest d in which no false nearest neighbors are discovered. Takens [27] suggested that the best embedding dimension is given as $2d + 1$ however this is not strictly the case.

To visualize the FNN methodology, consider another two dimensional chaotic system, the Henon Map (e.g. Eq (14) with a =1.4 and b = 0.3), and a reconstruction of the original phase space using only the x variable. For d = 1 the reconstructed scalar phase space is a line. In d = 1 dimension embedding space some points a and b are close to c as given in Fig. 25. If the embedding dimension is increased to d = 2 it is then clear, from Fig. 26, that while in this example a is close to c, b was a false neighbor in d = 1 dimension embedding space due to the constriction of the embedding space.

$$\begin{aligned} x_{t+1} &= y_t + 1 - ax_t^2 \\ y_{t+1} &= bx_t \end{aligned} \tag{14}$$

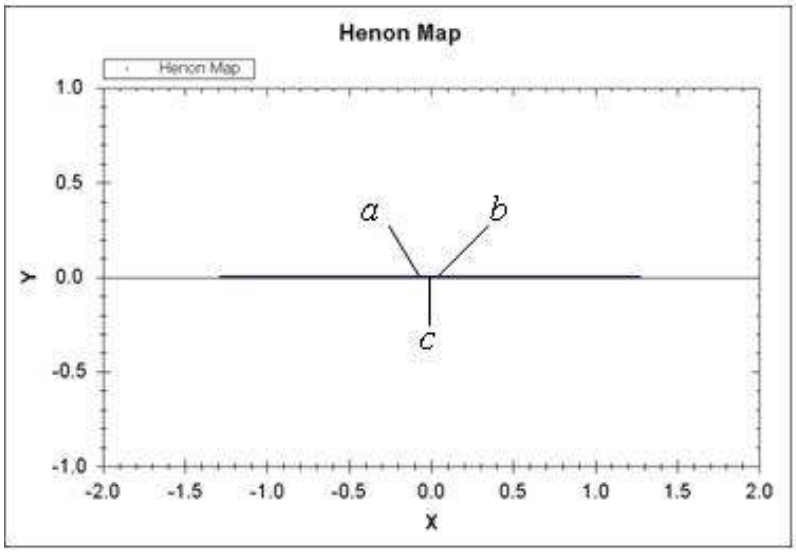

Figure 25. 1 Dimensional reconstruct of Henon Map with 2 nearest neighbors to *c*.

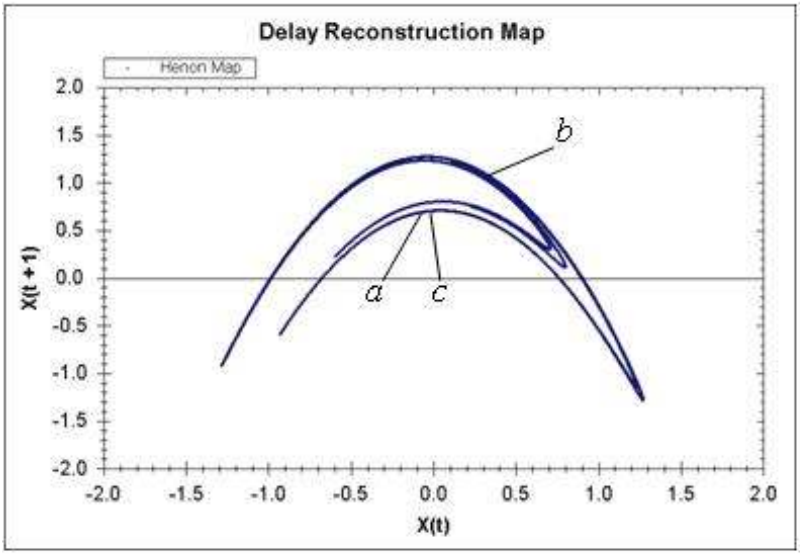

Figure 26. 2 dimensional reconstruction of the Henon map with 1 nearest neigh to *c* and one FNN *b*.

Once the time delay and minimum embedding dimension are calculated it is possible to estimate the maximum Lyapunov exponent of the time series. It is important to note that there exist as many Lyapunov exponents as dimensions of a given system. The estimation of the maximum Lyapunov exponent proceeds as follows. Consider two

trajectories, in reconstructed phase space, as given in Eq. (15) starting respectively from $\boldsymbol{x}(n_1)$ and $\boldsymbol{y}(n_1)$.

$$\begin{aligned}\vec{\boldsymbol{x}}(n_i) &= \{x_0, x_{0+\tau}, x_{0+2\tau}, \dots, x_{0+(d-1)\tau}\} \\ \vec{\boldsymbol{y}}(n_i) &= \{y_0, y_{0+\tau}, y_{0+2\tau}, \dots, y_{0+(d-1)\tau}\}\end{aligned} \tag{15}$$

Let the distance between $\boldsymbol{x}(n_1)$ and $\boldsymbol{y}(n_1)$ be some small value ϵ. Then denote by Eq. (16) the distance between the trajectories at some time, Δt, in the future.

$$\epsilon_{\Delta n} = \left\| \vec{\boldsymbol{x}}_{n_1+\Delta t} - \vec{\boldsymbol{y}}_{n_1+\Delta t} \right\| \tag{16}$$

Then the Lyapunov exponent, λ, of the trajectories for the selected variable is determined by Eq. (17) [30].

$$\begin{aligned}\epsilon_{\Delta t} &\cong \epsilon_0 e^{\lambda \Delta t} \; or \\ \lambda &= \lim_{t\to\infty} \frac{1}{t} \sum_{n=0}^{\infty} ln\left(\frac{\left\|\vec{\boldsymbol{x}}_{n_1+\Delta t} - \vec{\boldsymbol{y}}_{n_1+\Delta t}\right\|}{\vec{\boldsymbol{x}}_{t_1} - \vec{\boldsymbol{y}}_{t_1}}\right)\end{aligned} \tag{17}$$

Eq. (17) utilizes the limit as t approaches ∞. It would be a rare event in which the data collected from a given system represents an infinite time. Therefore, a time series analysis of the Lyapunov exponent is required to determine the degree of non-linearity within a constrained system. Consider a time series T and a neighborhood $U(\boldsymbol{x}(n_1))$ about a point $\mathbf{x}(\mathrm{n}_1) \in T$, such that each point $\mathbf{y}_{n1} \in \mathrm{U}(\mathbf{x}(\mathrm{n}_1))$ has a distance from $\boldsymbol{x}(n_1)$ of some small value less than or equal to ϵ. Then the maximum Lyapunov exponent of the time series T can be calculated as given in Eq. (18) [20].

$$\lambda \approx \frac{1}{N} \sum_{n_0=1}^{N} \ln\left(\frac{1}{\left|U(\mathbf{x}(n_1))\right|} \sum_{\mathbf{y}_n \in U(\mathbf{x}(n_1))} \left| x_{n_0+\Delta n} - y_{n+\Delta n} \right| \right) \tag{18}$$

The maximum Lyapunov exponent serves to determine if the given system is non-linear. Any system containing at least one positive Lyapunov exponent is defined to be chaotic [20]. Further, the magnitude of the maximum exponent indicates the degree of chaos involved in the system. The appendix of this thesis contains a sample method (using the C# programming language) for the calculation of the maximum Lyapunov exponent as discussed here. As has been shown above, the determination of chaos within a system involves a number of calculations but does result in a very clear metric for the evaluation of the system under question. The benefit of using the null hypothesis testing to eliminate non-chaotic systems should be clear due to the number of calculations required to declare a system chaotic.

The set of all Lyapunov exponents form the Lyapunov spectrum of the system. It may be desirable to calculate the Lyapunov spectrum to better understand the non-linearity's within the system. Unfortunately, calculation of the complete Lyapunov spectrum for a time series data set is extremely computationally intense and is therefore rarely attempted. The reader is referred to [30] for details on calculation of the complete Lyapunov spectrum for a given system.

2.3 Chaotic Attractors

Clearly, the chaotic systems under consideration in this thesis vary greatly from systems which are linear, random, or periodic. But what benefits are gained from considering systems as chaotic rather than using standard statistical or machine learning methods? This section further details the space of chaotic systems and highlights the benefits of using the embedded phase space for data analysis, prediction and control.

In §2.1 the Logistic equation was presented as an example chaotic system for specific values of *a*. Included at the end of that section was a phase portrait of that map. From the phase portrait it was clear that the system displayed distinct ergodicity. If ergodicity is defined in an exclusive context, such that systems may return to states that are very similar to previous ones but may not repeat exactly any of the previous states[3], then these systems are known as chaotic attractors.

Chaotic attractors are systems which revisit portions of the phase space in a ergodic fashion, with mixing, resulting in orbits (trajectories) of the system [31]. More formally, in a system with *D* degrees of freedom, an attractor is a subset of *D*-dimensional phase space towards which almost all sufficiently close trajectories are *attracted* asymptotically [13], [32]. Obviously, there are many attractors for which the behavior of the attractor is very simple and anticipated. Consider, for example, the system represented by Eq. (19). This system is asymptotically attracted to the fixed point 1.

$$x_{t+1} = \sqrt{x_t} \tag{19}$$

Similarly, the system defined by the cosine function (e.g. Eq. (20)) attends to a periodic cycle.

$$x_{t+1} = \cos(\mathrm{x_t}) \tag{20}$$

There is little remarkable about either of the systems in Eq. (19) or Eq. (20). However, there are systems which do produce remarkable attractors. Systems which exhibit chaotic dynamics, such as those of the focus of this thesis, produce very remarkable attractors. These types of attractors have been termed *strange* attractors due

[3] Revisiting exactly a previous state would cause a deterministic system to be periodic rather than chaotic.

to the chaotic dynamics of the systems they represent [33]. Confirming this, Eckmann *et al* [30] stated that the notion of strangeness refers to the dynamics on the attractor and not just to its geometry and that this applies whether the time is discrete or continuous.

The transformation of a time series dataset into phase space through the time delay embedding [27] discussed in §2.2 provides for a representation of the attractor of the system. As Cross *et al* [34] stated, each embedding is a representation of the original phase space and its attractor. Interestingly, this attractor will form regardless of the parameter upon which the transformation is built, given the parameter was pertinent in the dataset. Eckmann *et al* [30] stated that an attractor is by definition invariant under a dynamical evolution. Further, all geometric measures of embeddings are diffeomophism invariants [34]. Thus for the Henon map given in Eq. (14) the attractor which appears when the x axis of the system is transformed into embedding phase space will also appear if the y axis is transformed in a similar manner. This fact is illustrated in Fig. 27 & 28.

Attractors allow for a simpler representation of the dynamics of a chaotic system. Often the original (experimental) data of a chaotic system is represented in a high dimensional space (e.g. more than 100 dimensions/parameters for wind turbines [35]). However, it is often the case that a number of these parameters do not specifically represent the dynamics of the system, nor do they represent the topology of the dynamics properly. Attractors, usually, have a significantly simpler topology [36] thus, they represent the dynamics of the system in a simpler, clearer fashion.

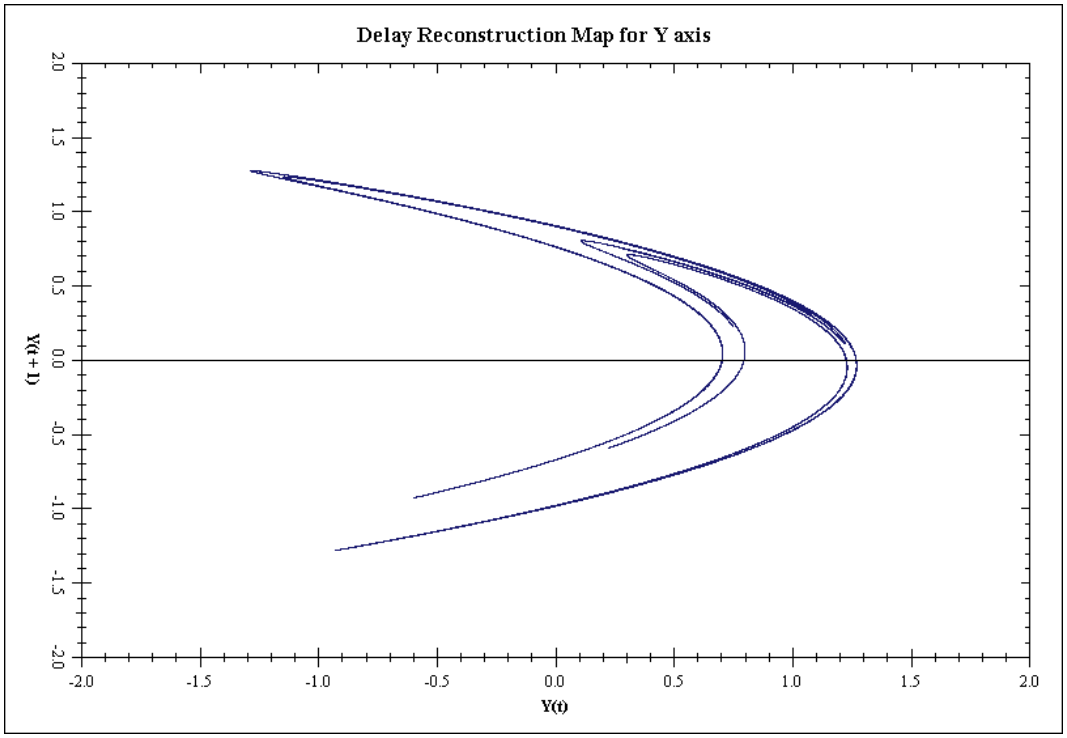

Figure 27. Reconstructed phase space for Henon Map using the *y* axis data.

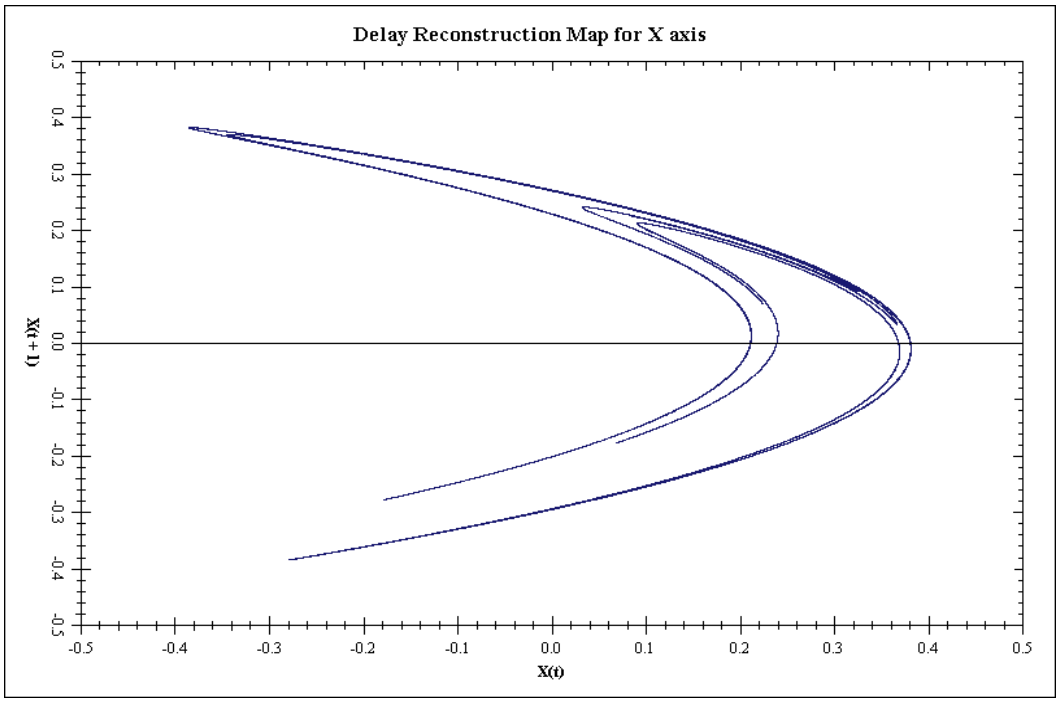

Figure 28. Reconstructed phase space for Henon Map using the *x* axis data.

Embedding of a dynamical system into phase space, and therefore describing its attractor, offers many benefits for analysis, prediction and control over working with the original dataset. Perhaps the largest benefit is found in clear understanding of the divergence/convergence of initially close points (e.g. the system dynamics). Obtaining this understanding will greatly assist in the prediction of future states of the system based upon current states. This guidance is not clearly discovered in standard data mining, machine learning or statistical methodologies that are often employed to predict, classify and control these types of systems.

In addition to the understanding of divergence/convergence, embedding a dynamical system into phase space, and thus recreation of its attractor, offers a greater understanding of system variable interactions. Often in machine learning, statistical algorithms such as singular value decomposition or principle component analysis are employed specifically to determine important parameters of a system [5]. Through the use of embedding phase space these interactions are automatically discovered and are implicit to the embedding therefore requiring no addition algorithmic steps to complete this understanding.

This section has offered an overview of deterministic chaos. The nature of chaotic dynamics has been presented. The transformation of time series datasets into embedding phase space has been described. The calculation of important metrics for the embedding phase space has been presented. Finally, the nature, use and benefit of chaotic attractors for analysis and prediction have been discussed. The next section shall propose a method

for the detection of normality in chaotic systems, using the above mentioned chaotic dynamics.

CHAPTER 3. NORMALITY IN CHAOTIC SYSTEMS

Understanding normal activity, or states, of a chaotic system is a nontrivial task. Determining the normal system dynamics requires a transformation into embedding phase space, accomplished through techniques such as the delay embedding theorem of Takens [27] discussed in the previous section. There exist cases, such as with data derived from an experimental time series rather than exact equations of motion, where the embedding phase space can prove to be of limited assistance to the visualization of the dynamics of the system in general. This is due, in part, to the "bird's nest" quality of the phase space often experienced in visualization of traditional embedding phase space for such data. This section seeks to present a methodology for determining normality within deterministically chaotic systems and to better represent the dynamics in a course-grained fashion for those scenarios where the normal embedding phase space is of little assistance.

The literature indicates a large amount of work has been performed in the domain of anomaly detection, which is closely related to normality detection. Chandola *et al* [37] presented a comprehensive survey of research performed in this area, in general. Specific to deterministically chaotic systems, work has been reported on chaotic anomaly detection by Aydin *et al* [38] using a chaos-based negative selection algorithm. Geisel *et al* [39] discussed the diffusion of anomalies in chaotic systems. These are but two of a great many works specific to anomaly detection in chaotic systems. Application oriented works are also present in the literature such as the work of Xiong *et al* [40] which presented research on computer network anomaly detection.

Curiously, the literature is largely silent on the specific detection and analysis of normal operating states of deterministically chaotic systems. While much work has been illustrated on the analysis of specific chaotic equations of motion (see for example [41], [42], and [23]) little has been mentioned of a generalized understanding of normality for such systems. This is especially true for the analysis of experimental data of a chaotic system.

3.1 Ergodicity Plots

This section presents a novel methodology for determining normal system states. Additionally, a new visualization technique is presented that represents a significant advance over current visualization techniques for the general understanding of the dynamics of deterministically chaotic systems. The methodology proposed in this section is used to detect anomalous behavior of the system as well as to predict future states of the system. A short case study concerning the weather in Cedar Rapids, IA, USA, is presented as illustration of the efficacies of the methodology. Further case studies are reserved for later sections of this thesis.

Consider a system exhibiting deterministic chaos. The mathematical space best used for understanding the dynamics of this system is that of embedding phase space as presented in section 2. Due to the nature of the system, as deterministically chaotic, there exists a wholesale revisiting of the entire phase space by the chaotic behavior [43]. This revisiting previous regions of the phase space is a representation of the property of ergodicity and closely follows the Poincaré Recurrence Theorem.

Due to this ergodicity there exists an ability to understand the manners by which a chaotic system revisits certain regions of phase space. Such an understanding enhances

the overall system understanding in general and is the basis for the understanding of the general dynamic states of the chaotic system being proposed in this section.

Eckmann *et al* [44] utilized the property of ergodicity to create recurrence plots (RP), described below. The RP is effective in describing the timing of revisitation of regions of the phase space, especially for larger dimensional systems. In fact, the purpose of RP is to indicate, for a given moment in time, the times at which a phase space trajectory revisits the same area in the phase space as the point under consideration. Taken as a matrix the RP can indicate the wholesale revisiting of the phase space spoken of previously.

Consider a point $\vec{\boldsymbol{x}}(i)$ on the orbit describing a dynamical system in *d*-dimensional embedding phase space, for i=1,…,N. The RP is an array of pixels, in a $N \times N$ square, where a pixel is colored at (i,j) if $x(j)$ is within some small distance ε of $x(i)$ [44]. The RP should be viewed as a binary $N \times N$ matrix where element (i,j) is assigned 1 if $x(j)$ is within some small distance ε of $x(i)$ otherwise it is assigned 0. This matrix is fairly symmetric with respect to the identity diagonal; however this symmetry is not a necessary condition of the RP. The RP indicates times in which points revisit the embedding phase space in an ergodic fashion. This can be beneficial in determining system dynamics such as quasiperiodicity. In fact, diagonal lines parallel with a central diagonal of the RP matrix indicate periodicity within the dynamic system; and the longer these diagonal lines are the greater the periodic component of the system. This can be seen in Fig. 29 &30 which plot the RP for the periodic system of Eq. (21) and the chaotic system of Eq. (22) respectively.

$$x_t = \sin(t) \tag{21}$$

$$x_t = \sin(2\pi x_{t-1}) \tag{22}$$

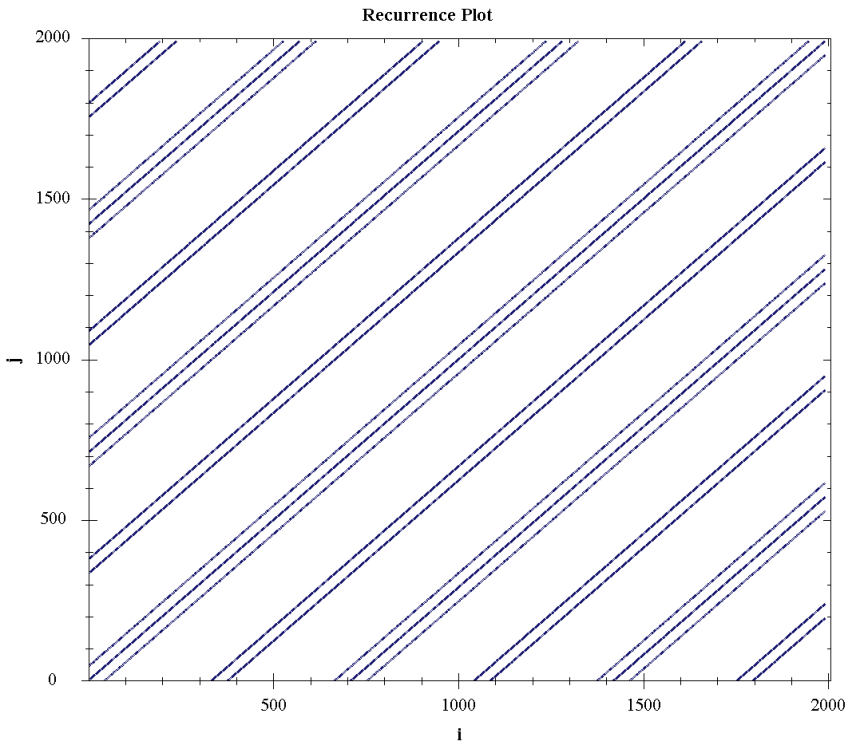

Figure 29. Recurrence Plot for the system in Eq. (21).

Unfortunately, the RP does not offer a clear understanding of the normal activity of visiting regions of phase space, rather just what times points are in similar regions. It is difficult to determine clearly from the RP which regions of phase space are being revisited and at what frequency these regions are visited. Further, the RP does not give a clear indication of normal local dynamics of the system. It is well known that certain regions of chaotic systems are more divergent while other regions tend to converge [45]. This information is not clearly portrayed in the RP.

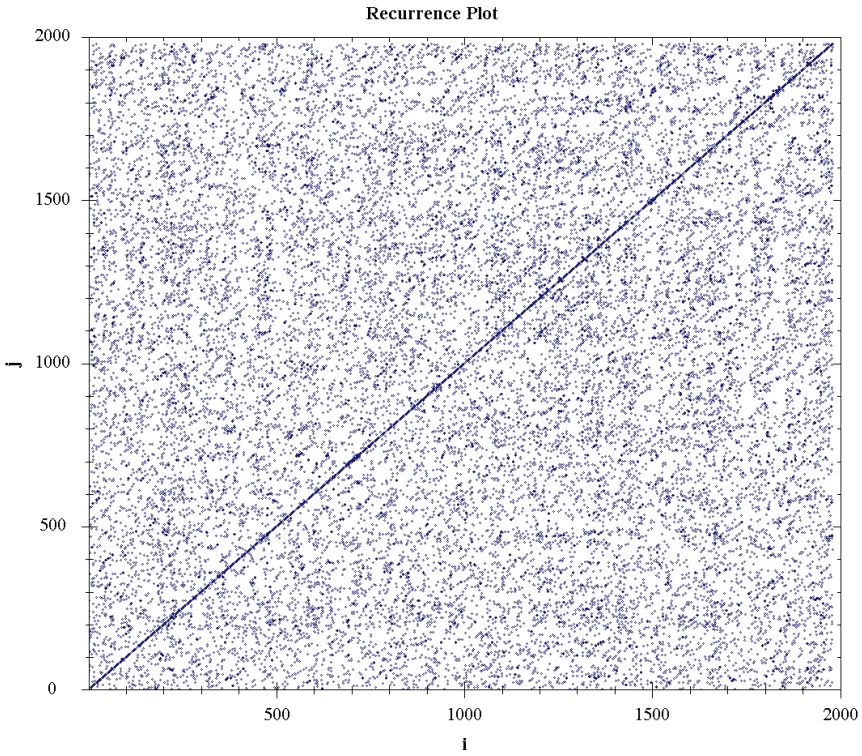

Figure 30. Recurrence Plot for the system in Eq. (22).

To understand which regions, or states, are being revisited by the trajectories of a system, a new visualization algorithm is suggested. Consider the embedding phase space of a given chaotic system. Impose on the embedding phase space an $N \times N$ grid such that regions of phase space are equally divided into square cells of size $\frac{1}{N} \times \frac{1}{N}$ where N is a user defined parameter. Then plot the embedding phase space by assigning to each point the centroid of the grid cell representing the region of phase space which that point is a member of. Each centroid can be considered a region of phase space, or more importantly a state of the dynamical system. This plot is herein termed the Ergodicity Plot (EP) of the embedding phase space. The appendix of this thesis contains a pseudocode method for computer software for the generation of an EP as described here.

The EP of a system allows for better visualization of the regions of phase space being dynamically visited by the system and therefore the states in which the dynamical system finds itself at given times. Fig. 31 & 32 graph the EP for the systems in Eq. 21 & 22 respectively. From Fig. 31 & 32 it is clear that the systems revisits specific regions of the phase space in a recurring fashion, a fact that was not abundantly clear in the RP of the systems. Additionally, the Ergodicity Plot indicates the states that are traversed during the ergodic revisiting of previous states.

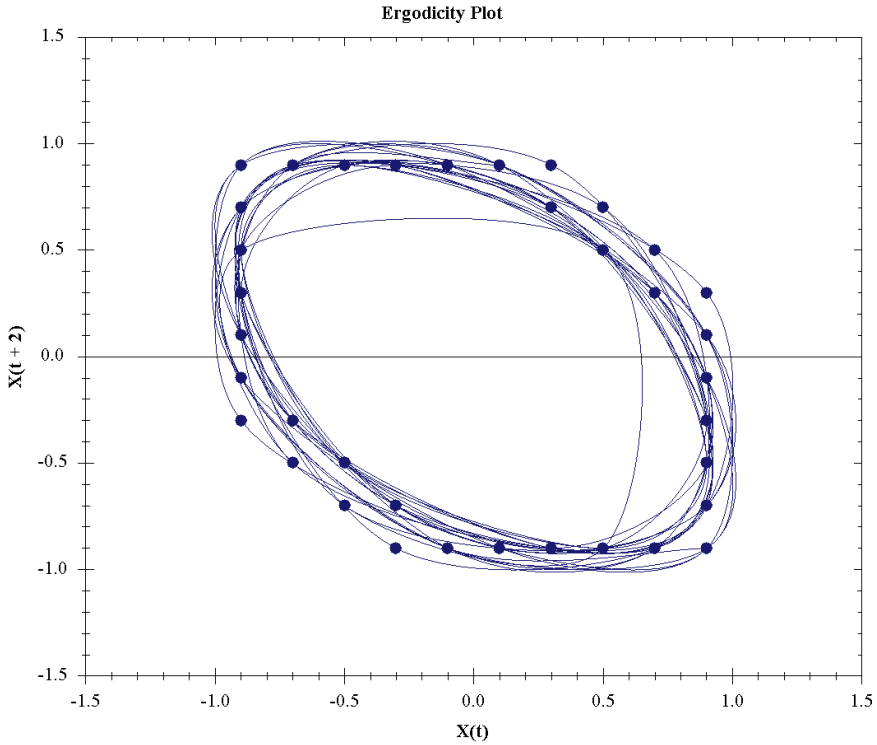

Figure 31. Ergodicity Plot for the system in Eq. (21) with *N*=10.

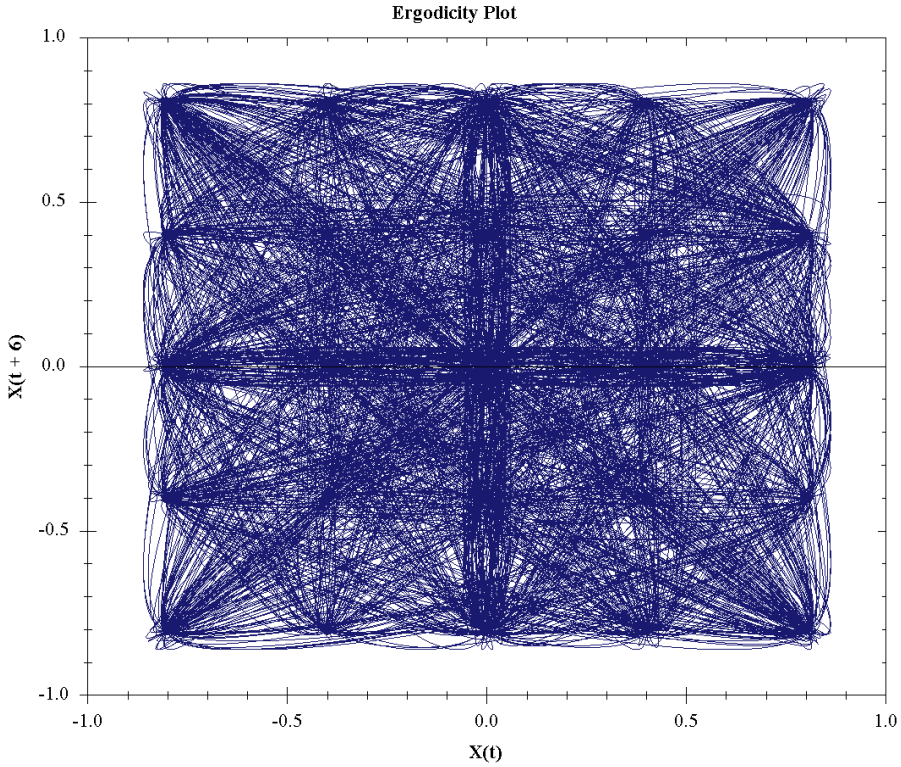

Figure 32. Ergodicity Plot for the system in Eq. (22) with $N = 5$.

To clearly demonstrate the effectiveness of the EP over the RP the following example is given. Consider the raw data for the monthly mean temperature (F) for Cedar Rapids, IA, from 1902 through 2009, given in Fig. 33 where the horizontal axis represents time (months since 1902) and the vertical axis represents the mean temperature in degrees Fahrenheit. Generating an EP for this data with a time delay of 1 and an embedding dimension of 2 it is possible to detect system states as shown in the EP of Fig. 34. These states are rendered less clearly in the RP of Fig. 35, although the quasiperiodicity is discovered in the slight diagonal lines displayed by the RP upon magnification.

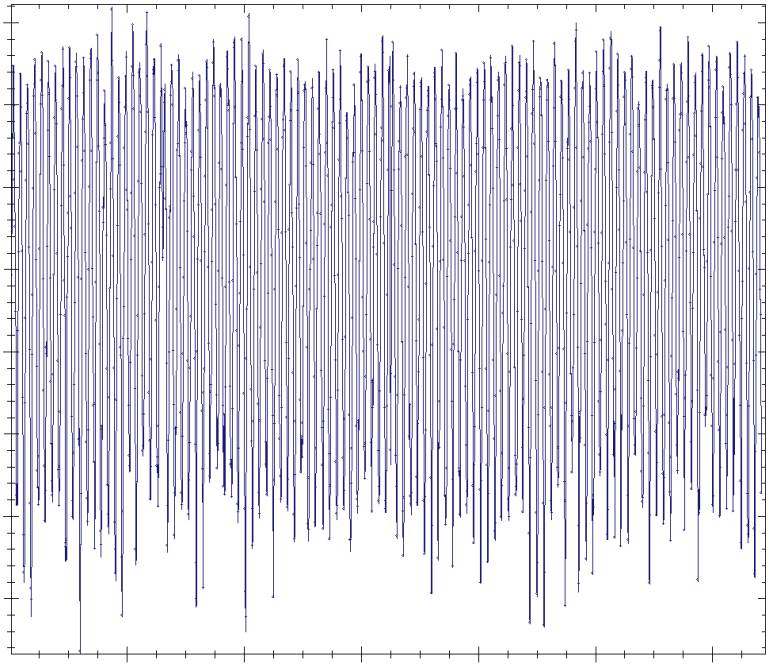

Figure 33. Monthly mean temperature (F) for Cedar Rapids, IA from 1902 through 2009.

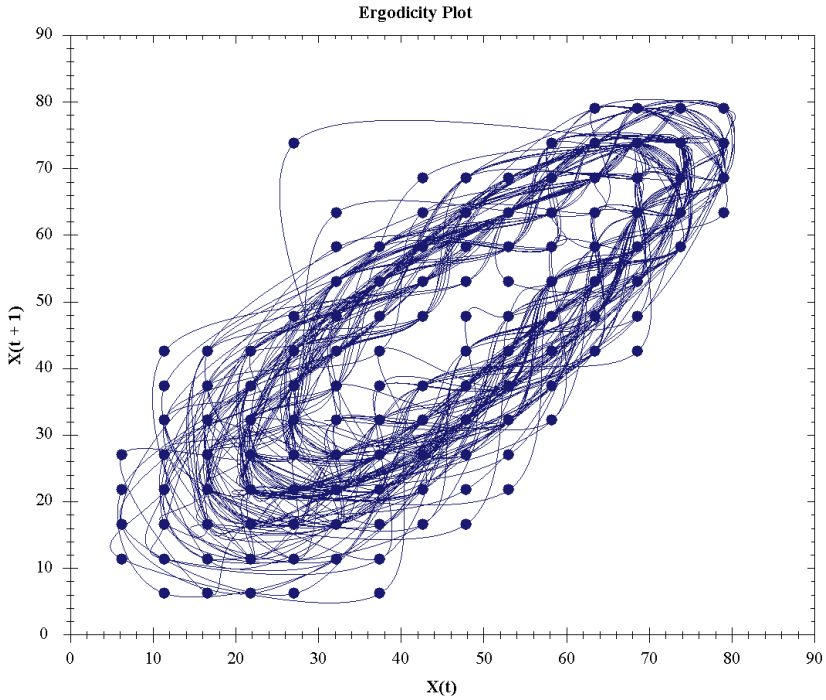

Figure 34. EP for the data in Fig. 33 with $N = 15$.

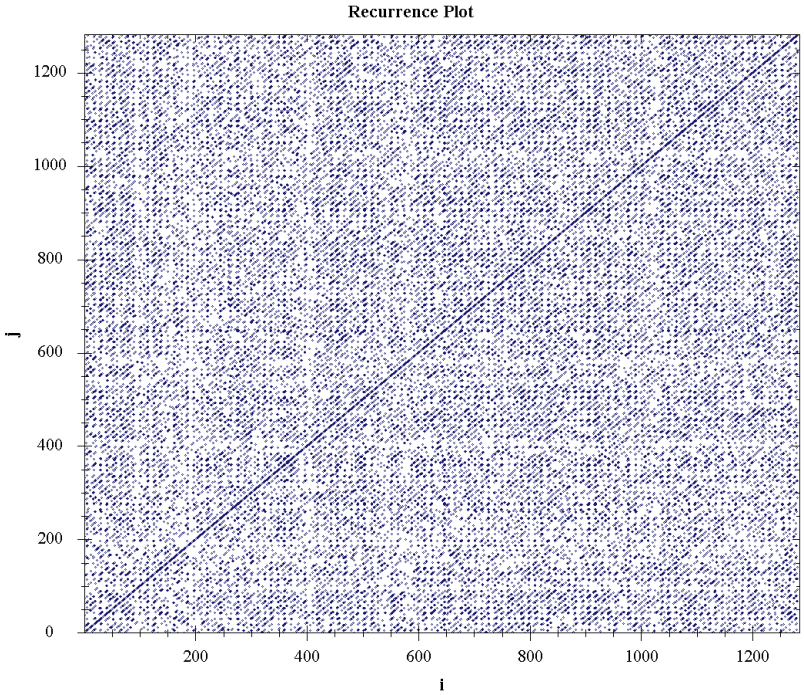

Figure 35. RP for the data in Fig. 33.

The EP is a powerful new tool for determining the activity of a deterministically chaotic system. Consider such a system operating under normal conditions. Here normal conditions are considered such when the system is experiencing no unique or rare perturbations or other external influences, aside from the normally present external influences that may cause noise. Through the use of the EP, it is easy to determine the normal state transitions of the given system as discussed in the next subsection. Coupling the normal state transitions with the invariant metrics of the Lyapunov exponent and the system entropy give a more complete picture of the system's normal operation.

Due to the enhanced understanding of the normal operations of a deterministically chaotic system, there also exists a new ability to detect anomalies that may occur within the system. Anomalies may be automatically detected when a change in the Lyapunov exponent, $\delta_L > \varepsilon$, is detected to be greater than some threshold. Anomalies may also be

detected by a change in the normal ergodicity of the system resulting in new or rare ergodic paths being traversed with a greater probability than under normal conditions. Finally, anomalies can be detected as new points in embedding phase space which vary from the anticipated orbit of the system by some value greater than a set threshold.

Not every anomaly that is detected can be considered negatively. Many chaotic systems will evolve over time and the detection of these changes will initially be perceived as anomalous. In example, the cyber activity of a given user is well represented by deterministic chaos (see the cyber security case study in Chapter 5). However, the activity of that user may change over time such that the user is performing new duties related to a new project that would not fit into a representation of their normal cyber activity. Therefore, it is beneficial to study anomalies over time and if needed evolve the EP and Lyapunov exponents based on new normal data.

Another important benefit of the EP is that it allows for classification of systems. Through the more course grained representation of the dynamics of the system, given by an EP, it is possible to plot the EP for various systems and to use these plots as data for a classification or pattern detection algorithm (In Chapter 6, presentation is given of the use of the EP to classify users of a massive multiplayer online environment.). The EP can be used by standard pattern detection and image analysis algorithms to classify, or even cluster, multiple chaotic systems. This data mining advantage is not accomplished easily using the standard attractor reconstruction or the use of RPs.

3.2 Future State Prediction

The EP can be considered akin to symbolic dynamics of mathematics. The study of Symbolic dynamics offers a course-grained description of systems [46]. Symbolic dynamics typically considers smooth topological systems and represents them in a discrete space consisting of sequences of abstract symbols. Unfortunately, not all deterministically chaotic systems can be considered "smooth". Nor are the states of the systems generally considered abstractly. Regardless, the relation between the EP and symbolic dynamics exists and is close.

Similar to the relation between the EP and symbolic dynamics, a relation is found between thermodynamic formalism and the EP. Gupta *et al* [47] posited the use of Potts models (e.g. extensions of the standard Ising model) for detection of patterns in complex systems. This methodology utilized the discovery of Markov symbol chains in the system as it was described using symbolic dynamics [47].

Due to the similarities between the EP and the theories of symbolic dynamics and thermodynamic formalism it is possible to use the EP to describe mathematically state visitation of the system as well as to predict the order and time of the visiting of future states. The deterministically chaotic systems under consideration here can be viewed as a matrix of regional state spaces that are visited in a specific order, ergodically, over time. Therefore, similar to symbolic dynamics, it is possible to generate Markov style chains of the states visited by the system and to determine, with a given probability which states will be visited next. Crutchfield *et* al [48] indicated that the Markov style models (IE: Hidden Markov Model) were insufficient for expressing the dynamics of chaotic systems.

The state prediction algorithm proposed in this section is not a Markov chain, nor a Hidden Markov Model, however, as it more closely follows an Apriori style algorithm.

Using the proposed EP it is possible to determine frequent system state change motifs for any given system. System state change motifs indicate the traversal of the system states by the orbits/trajectories of the systems under consideration. The system state change motifs further assist in understanding the dynamical system represented by its time series data. Frequent system state change motifs are generated from the EP using an Apriori style analysis as follows.

While generating the EP, for the given system, an ordered list of regions of the phase space (system states) visited may be maintained. From this ordered list it is possible to generate a list of frequent 1 region sets; these are sets in which the number of times the region has been visited is above a certain threshold. After the formation of the frequent 1 region sets, all records containing regions not deemed frequent can be removed and the frequent 1 region sets may be combined, in order, to form 2 region sets. The iterative removal of the non-frequent records and the combining of the remaining records continue until such time as no records remain to combine. This combination and evaluation of the region sets closely follows the Apriori style analysis of data sets [5].

The frequent patterns generated in the above Apriori style algorithm form system state change motifs for the system under consideration. Associated with each system state change motif is a support and confidence metric which indicate the strength of the motif. The greater the support metric for a motif, the more often it is discovered in the system. These motifs offer a deeper understanding of the traversal of the system orbits and indicate the changes taking place in the system as it evolves over time. These motifs not

only offer substantial insight into the system but may be used to assist in prediction of future system states.

A mathematical representation of the generation of frequent state change motifs can be described follows. Let Ω be the set of all system states σ_i created through the imposition of the $N \times N$ grid on the embedding phase space of the system. Then each initial state $x_o \in \Omega$ of the system generates a sequence of visited states defined by a mapping M such that $M(x_0) = \{\sigma_0, \sigma_1, \dots, \sigma_n\}$ indicates the regions of phase space that have been visited in order. $M(x_0)$ is frequent if, and only if, the probability of the sequence is greater than the frequency threshold, $Prob\big(M(x_0)\big) > \epsilon$. It is, therefore, easy to see that the traversal of frequent system states forms a Markov type chain with a given probability of occurrence.

Prediction of future system states is based upon the traversal of the orbits about the phase space in the EP. Using the system state change motifs described above, it is a relatively straight forward task to predict the next states of a system, given a set of the most recent states of the system. Given a system has most recently traversed a small set of ordered state regions; it is simply a matter of comparing the most recent traversals with the detected system state change motifs. From the system state change motifs it is possible to assign a probability to predict the next set of states a system will traverse with a given accuracy.

In the example of the mean temperature data above it was discovered that the system frequently moved from the state represented by the centroid 68,73 to the state represented by the centroid 63,68 and then to 53,63 then to 37,53 and finally to 21,37. This motif was found to exist in 67% of the orbits. Thus, 67% of the orbits moved from

68,73 to 63,68 to 53,63 and so forth. It was also discovered that 18% of the orbits traversed from the state represented by the centroid 68,73 to the state represented by the centroid 63,68 and then to 47,63. Therefore, it is possible to state that if a system was recently in the region of 68,73 and then 63,68, there is a 67% probability that it will next be in the region 53,63 then in region 37,53 and finally be in region 21,37 and there is an 18% probability that the system will next visit the region 47,63.

The results above are intuitive for the system of mean temperature. It is often discovered that the mean temperature, of the considered region, cycles from warm to cold to warm again. This is easily seen in the system state motifs that were discovered. The system traversed from a warm state (68,73) to a cooler state (63,68) to an even cooler state (56,63) and then to (37,53) and finally to (21,37). This traversal was discovered in 67% of the orbits of the system. However, in 18% of the trajectories there was a path from the same initial starting point (68,73) to a similar second state (63,68) and then to a different third state of (47,63).

There may exist cases where the simple list of probabilities may be insufficient as a result. In such cases it is possible to form a weighted mean of the resulting states to enhance the prediction of the next system state. The weighted mean utilizes the probabilities discovered in the system state motif detection algorithm as described in Eq. (23) where $Pred(x_i)$ is the predicted value for the *i*th state coordinate, ρ_k is the probability of the x_k state discovered in the motif detection.

$$Pred(x_i) = \frac{\sum_{k=1}^{N} \rho_k x_k}{\sum_{k=1}^{N} \rho_k} \tag{23}$$

Using this mean state prediction technique the first three states of the weather system in the above example would be 68,73 to 63,68 and then to 54,63. In addition to allowing for a single predictive point on the third state in this example, it is also instructive to see that the single predictive point forms a norm of sorts for the cyclical action of the chaotic system. The state predictive algorithm suffices for system generalization when actual single point prediction is not necessary. Further, the state predictive algorithm allows for prediction of longer periods than obtainable by single point prediction using chaotic prediction techniques, discussed in Chapter 4 of this thesis, or stochastic prediction methods.

In conclusion, a new methodology for visualizing the states of a chaotic system has been introduced. The traversal of these system states was represented by the novel Ergodicity Plot (EP). The proposed EP represents the system dynamics in a much clearer fashion than is given in the standard RP and can be used to detect system normality. Further, the use of the EP allows for detection of frequent system state change motifs which are then used to predict the next system states. The motif detection algorithm presented closely followed the Apriori algorithm for discovery of frequent sets and was used to discover frequent state change patterns or motifs. The frequent state change patterns allow for efficient prediction of future states of the system without the need for individual value prediction discussed in Chapter 4 of this thesis.

CHAPTER 4. PREDICTION OF CHAOTIC SYSTEMS

Time series prediction plays an important role in many domains including medical diagnosis, energy planning, weather prediction, and warning of catastrophic events. Unfortunately, many of these domains are represented by systems which exhibit deterministic chaos, making accurate prediction of future data values difficult.

Prediction of future values in a system which appears random, but operates from deterministic equations is a challenging task that has been addressed by many researchers over the past two decades. Farmer *et al.* [49] addressed this topic in 1987 using a *k*-nearest neighbors methodology in embedding phase space to predict the next states of the system. The prediction technique of [49] is considered a local technique based upon a given neighborhood about the point being predicted. The nearest neighbor technique, as it is often called, simply selects the nearest neighbor to the last point in the time series (on a separate orbit) and follows that point's orbit for *n* time increments. Casdagli [50] proposed a nonlinear prediction algorithm which derived the equations of motion for the chaotic system through interpolation of the chaotic time series data. Zhang *et al.* [2], in 2004, utilized the idea of Lyapunov exponents and local neighborhoods to predict values *k* iterations in the future.

Other researchers have sought to enhance chaotic time series prediction rather than introducing new prediction techniques. Karunasingha *et al.* [51] proposed the enhancement of prediction through the use of non-linear noise-reduction techniques to improve data quality. It was proposed by [4] that the reduction of noise in the chaotic time series improved the prediction accuracy. As recent as 2009, Zhao *et al.* [43] introduced the technique of using multiple axes of data to predict another axis. Damle *et*

al. [52] sought to enhance time series prediction techniques through data mining in the nonlinear phase space.

While the literature contains various prediction techniques for chaotic time series data, it is widely accepted that the accurate prediction of deterministically chaotic time series is a relatively open problem. This section seeks to offer a solution which is aimed, at least partially, at solving the problem of accurate prediction. The proposed methodology makes use of Lyapunov exponents and the calculation of a globally optimal constant through the use of a modified evolutionary algorithm to predict values of a chaotic system in embedded phase space. As will be shown, this methodology holds promise for accurate prediction to the limit of the Lyapunov exponent of the considered variable. The proposed methodology is also very useful in predicting points which may be missing from a chaotic dataset in use by the other algorithms of this thesis.

4.1 A New Chaotic Prediction Algorithm

This section highlights a new algorithm for predicting future values of a chaotic system. The proposed chaotic prediction methodology makes use of the embedding phase space and the Lyapunov exponents discussed in Chapter 2. These elements are coupled with a modified evolutionary algorithm for the calculation of an optimal constant for the prediction schema. As described in Chapter 2, Lyapunov exponents determine the rate of divergence or convergence of points in embedded phase space. This fact can be exploited to the benefit of a prediction algorithm. Let s_1 and s_2 be two points in embedding phase space with distance $\|s_1 - s_2\| = \delta_0 \ll \varepsilon$, some small value. Denote by δ_n the distance between the evolution of s_1 and s_2 at some point n ahead in phase space.

Then define the Lyapunov exponent, λ, used here as given in Eq. (24) where *c* is some constant and *e* is the exponential [3].

$$\delta_n = \delta_0 e^{c\lambda} \tag{24}$$

This important definition forms the basis of the proposed prediction methodology. From this definition it is possible to accurately predict a new point given the current state (last point) of the time series and another point in the series which is close to the current state as follows. From the current state (last point) of the time series, x_{cur}, the nearest neighboring point, x_i, on a different orbit from the current state is found. The distance, δ_0, between these two points is calculated. The the distance, δ_1, is caluclated using Eq. (24). The system is then advanced one time step forward from the nearest neighboring point, x_{i+1}, and the predicted point, x_{pred}, is calculated as given in Eq. (25).

$$x_{pred} = x_{i+1} + \delta_1 \tag{25}$$

The prediction methodology above is described intuitively in Fig. 19.

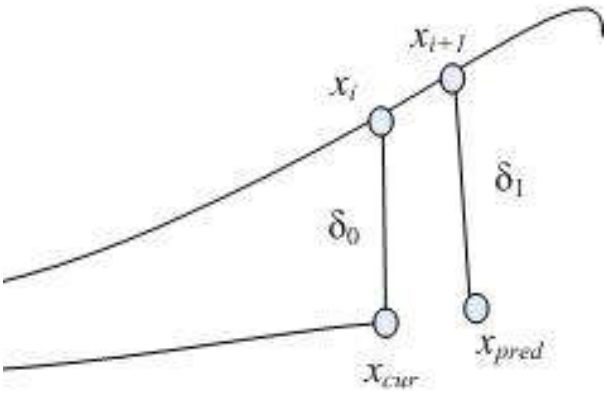

Figure 36. Proposed prediction schema

The prediction methodology then continues iteratively with the newly predicted point becoming the current state (last point) of the time series in embedding phase space. The predictions are easily transformed back into the original data space for the variable which the embedding phase space was created. Each prediction is appended to the end of the real space for the variable under consideration.

The methodology above assumes that the constant *c* in Eq. (24) has been calculated optimally. For this the use of a modified evolutionary algorithm is considered. Evolutionary algorithms are population based optimization algorithms which utilize the evolutionary processes of mutation and crossover [53]. The algorithm suggested here uses a population of individuals which are randomly generated as variants of an initial seed given to the algorithm for the constant *c* of Eq. (24). The individuals have a representation of the single number *c* and are being evolved to discover the optimal constant for the predictions schema. The mutation operator of the modified evolutionary algorithm mutates the individual by randomly adding/subtracting a small value to/from that individual's value respectively. The crossover operator is the major modification of the algorithm and produces a single child which is the mean value of the two parent's values.

The modified evolutionary algorithm takes an initial seed of *c* as input. The initial *c* is calculated from the current state (last point) of the time series and its closest neighbor as described above. The distance, δ_0, is calculated as before; then the distance, δ_{-1}, between the points preceding these two points is also calculated. The two distance measures are then used to estimate *c* as in Eq. (26)

$$c = \frac{ln|\delta_0 - \delta_{-1}|}{\lambda} \tag{26}$$

where λ is the, possibly local, Lyapunov exponent of the time series for the variable being predicted.

The fitness function of the modified evolutionary algorithm uses a training set of data from the time series and the *c* value of the individual to calculate the individual's error on predicting the training set. The algorithm progresses until the best individual's (the individual with the lowest error on the training data) error value has not changed for 100 iterations of the algorithm. The best individual of the algorithm is selected and its *c* value is used in the proposed prediction methodology.

For some datasets, especially those with larger Lyapunov exponents, the discovered constant may need to be updated throughout the prediction run. Larger Lyapunov exponents indicate greater divergence and therefore the constant may need updating to maintain good prediction results. Alternatively, local Lyapunov Exponents may be used in specific regions to enhance the prediction accuracy. Those datasets with smaller positive Lyapunov exponents do well with the single optimization of the constant for relatively larger numbers of prediction steps. The next section shall illustrate the use of this prediction methodology on three chaotic systems.

4.2 Prediction in Practice

This section illustrates the accuracy of the proposed prediction methodology on two well known chaotic systems, the Henon Map and the Duffing Map. Additionally, the methodology is applied to prediction of the generator speed of a wind turbine from a time

series containing historic values of the generator speed and wind speed. The accuracy of the proposed prediction methodology is indicated in each case by the relative and absolute errors for specific prediction points and the mean absolute error for 100 prediction points.

4.2.1 Prediction with the Duffing Map

The Duffing Map, as described in Eq. (12) is a well known difference equation with well understood chaotic dynamics. For the validation of the proposed prediction methodology 10,000 data points are used as the training set of the prediction algorithm. The prediction algorithm is then used to predict the next 100 points. The Lyapunov exponent for the 10,000 points of the Duffing Map was calculated to be 0.50493 and the optimal constant for the prediction algorithm given in Eq. (24) was generated by the modified evolutionary algorithm to be 0.760247. Table 2 describes the prediction accuracy for the first 10 points.

Table 2. Prediction Accuracy for the First 10 Predictions for the Duffing Map

Actual	Predicted	Absolute Error	Relative Error
0.088	0.086	0.002	0.030
0.260	0.253	0.007	0.028
0.681	0.662	0.018	0.027
1.505	1.237	0.268	0.178
0.592	1.376	0.783	1.322
1.120	0.946	0.174	0.155
1.555	1.474	0.081	0.052
0.288	0.650	0.362	1.257
0.457	1.218	0.761	1.667
1.103	1.413	0.311	0.282

As can be seen from Table 2 the proposed prediction algorithm does well until the 5th prediction step at which time it produces larger errors but then recovers. This is due to the region of phase space the prediction is in and its compression. This indicates a need for the use of the local Lyapunov Exponent. Fig. 37 shows the relative error for the 100 predictions of the Duffing Map. Clearly, the prediction schema does fairly well with most of the 100 predictions but there are some points in which the prediction schema performs poorly. As will be seen in §4.2.2, this is dataset dependent. The mean absolute error for the predictions of the Duffing Map was calculated to be 0.99687.

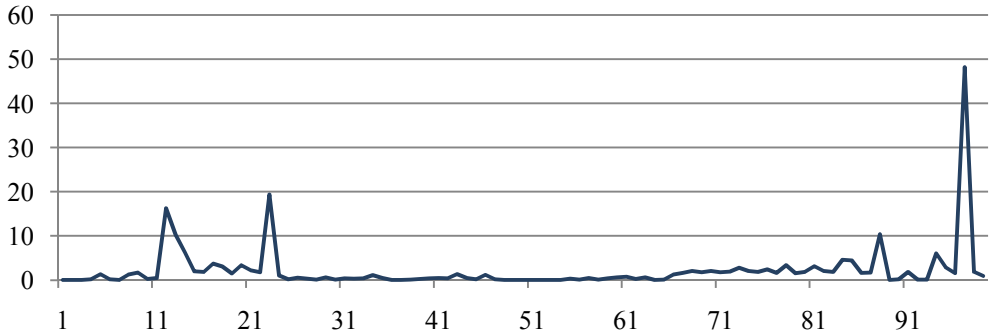

Figure 37. Relative error for the proposed prediction technique on the Duffing Map.

To facilitate a complete analysis of the new prediction technique a comparison against existing techniques is given. Two existing techniques are chosen for comparison. The neural network technique is chosen as it is a very well known and accepted technique often used to perform prediction of nonlinear systems. The neural network used in this comparison consisted of a topology with an input layer of 5 nodes, a hidden layer of 10 nodes and an output layer, the prediction. Standard sigmoidal activation functions were used throughout the network and were updated in a back propagation methodology [5].

The second existing technique chosen is the nearest neighbor prediction algorithms put forth by Farmer *et. al.* [49] and described above.

Applying the neural network prediction schema to the Duffing Map produced a result which was significantly better than the currently proposed technique. The neural network predictor predicted with a mean absolute error of only 0.3278. Fig. 38 plots the relative error for the neural network predictor on the Duffing Map.

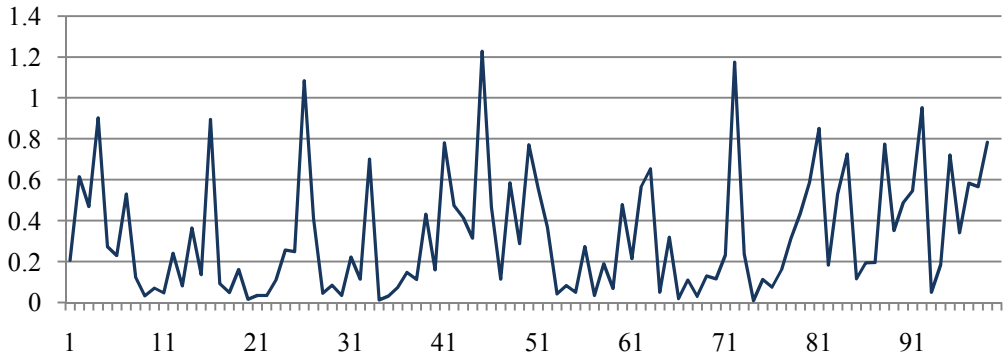

Figure 38. Relative error of the neural network predictor on the Duffing Map.

For the final comparison, the nearest neighbor methodology of prediction is applied to predict the next 100 data points of the Duffing Map. The nearest neighbor technique produced a result with a mean absolute error of 1.1519. Fig. 39 illustrates the relative error for the nearest neighbor technique on the prediction of the next 100 data points of the Duffing Map.

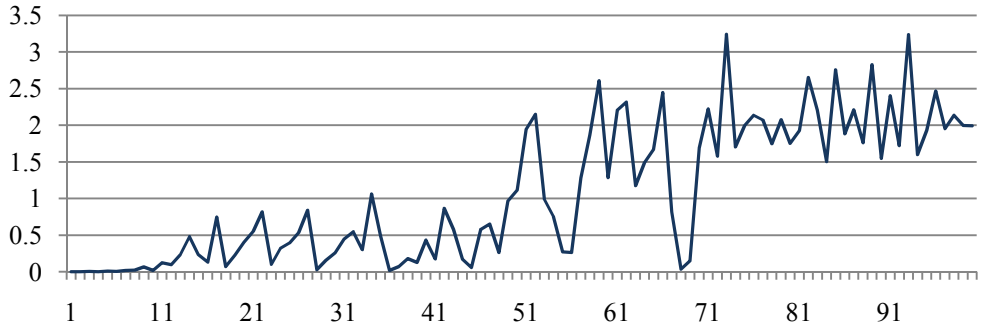

Figure 39. Relative error of the nearest neighbor prediction technique on the Duffing Map.

In the case of the Duffing Map, the neural network prediction methodology outperformed the two chaotic techniques. Disappointingly, the proposed technique was not as accurate as the neural network technique for prediction. However, it did outperform the nearest neighbor technique. Table 4 illustrates directly the performance of the three techniques on the Duffing Map. It is hypothesized that the cause of this poor performance was the sparseness of data in the center of the Duffing Map. As will be seen in the next sub-section, the new technique performs far better when the phase space does not contain this sparseness.

Table 3. Comparison of the mean absolute error of the prediction techniques on the Duffing Map.

Prediction Technique	**Mean Absolute Error**
Neural Network	0.3278
Nearest Neighbor	1.1519
Proposed Technique	0.9969

4.2.2 Prediction with the Henon Map

The Henon Map, as described in Eq. (27), is another well known difference equation with well understood dynamics. As with the Duffing map, the validation of the proposed prediction methodology uses the first 10,000 data points as the training set of the prediction algorithm. The prediction algorithm is then used to predict the next 100 points. The Lyapunov exponent for the 10,000 points of the Henon Map was calculated to be 0.427386 (given a = 1.25 and b = 0.3) and the optimal constant for the prediction algorithm given in Eq. (24) was generated by the modified evolutionary algorithm to be -4.4055. Table 4 illustrates the prediction accuracy for the first 10 points of the prediction.

$$\begin{cases} x_{t+1} = y_t + 1 - ax_t^2 \\ y_{t+1} = bx_t \end{cases} \tag{27}$$

As can be seen from Table 4, the prediction algorithm does much better with the Henon Map dataset than it did with the Duffing Map dataset. This is partly due to the lower Lyapunov exponent value of the first 10,000 points of the Henon Map and the lack of the sparse center region found in the Duffing Map. Fig. 40 shows the relative error for the 100 predictions of the Henon Map. Clearly, the prediction schema does fairly well with most of the 100 predictions. While there are some larger relative errors in the predictions, there is not the large error exhibited in the Duffing Map. The mean absolute error for the predictions of the Henon Map was calculated to be 0.512295.

Table 4. Prediction Accuracy for the First 10 Predictions for the Henon Map

Actual	Predicted	Absolute Error	Relative Error
1.148	1.177	0.029	0.025
-0.933	-1.030	0.097	0.104
0.125	-0.133	0.258	2.064
0.698	0.666	0.032	0.046
0.355	0.339	0.016	0.045
1.0328	1.039	0.006	0.006
-0.387	-0.411	0.024	0.062
1.101	1.07561	0.025	0.023
-0.812	-0.743	0.069	0.0850
0.408	0.550	0.142	0.348

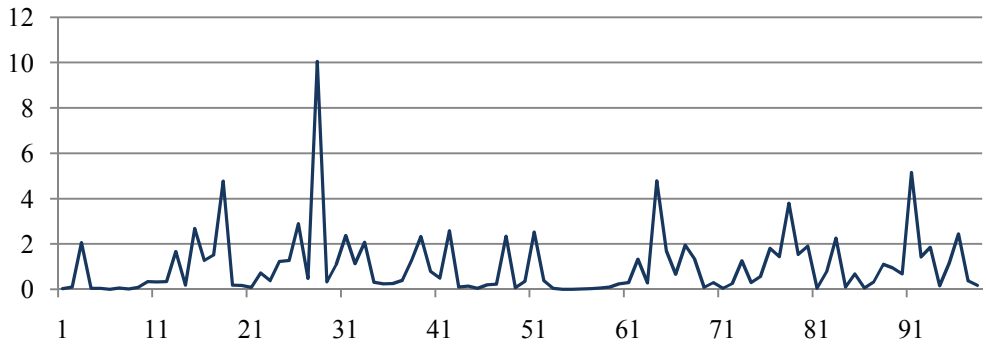

Figure 40. Relative error the proposed prediction technique on the Henon Map

Again, the results of the new prediction methodology should be compared with those of existing methods. Using the same initial conditions as those which created the results in Fig. 40, a neural network, with the topology as described for the Duffing Map example, was used to predict the next 100 points of the Henon Map. The neural network produced a mean absolute error of 0.8857. Fig. 41 shows the relative error of the neural network for the 100 prediction points.

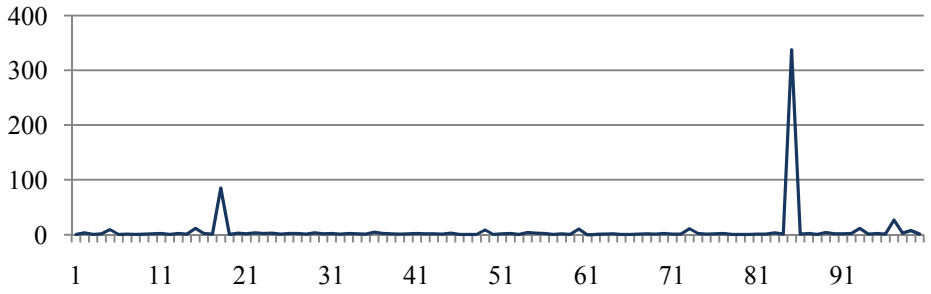

Figure 41. Relative error of the neural network on the Henon Map

To complete the comparison of the new prediction methodology to existing methodologies on the Henon Map, the nearest neighbor methodology, as described above in the Duffing Map example, was used to predict the next 100 points of the map. The nearest neighbor prediction methodology produced a mean absolute error of 0.6242. Fig. 42 plots the relative error of the nearest neighbor method prediction of the Henon Map.

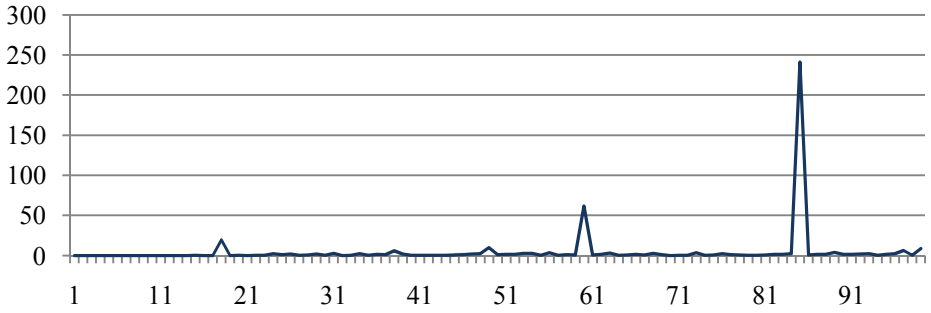

Figure 42. Relative error of the nearest neighbor prediction methodology on the Henon Map.

It is easily seen that the proposed prediction methodology outperforms the existing methodology techniques. It is also shown that the neural network technique

performs the worst of the three techniques. With the lack of sparseness that was experienced in the Duffing Map, it is clearly shown that chaotic prediction techniques are more adept than standard stochastic techniques for deterministically chaotic systems. This statement is especially true of those systems which do not contain sparse regions of phase space within the orbits of the system. Table 5 illustrates directly the comparison of the three prediction techniques.

Table 5. Comparison of errors for prediction of the Henon Map.

Prediction Technique	**Mean Absolute Error**
Neural Network	0.8857
Nearest Neighbor	0.6242
Proposed Technique	0.5123

4.2.3 Prediction on Wind Turbine Generator Speed.

For this section, data from a commercial wind turbine in service was captured at 10 second intervals. From the over 100 parameters of the captured SCADA data two parameters were used for this example, the generator speed and the wind speed. The combination of these two parameters was used in this prediction exercise. 4000 data points were used as the training set for the prediction algorithm and the generator speed was selected as the variable to predict. The original 4000 points of the generator speed are shown in Fig. 39, with the corresponding wind speed, and the phase portrait of those 4000 points is given in Fig. 44.

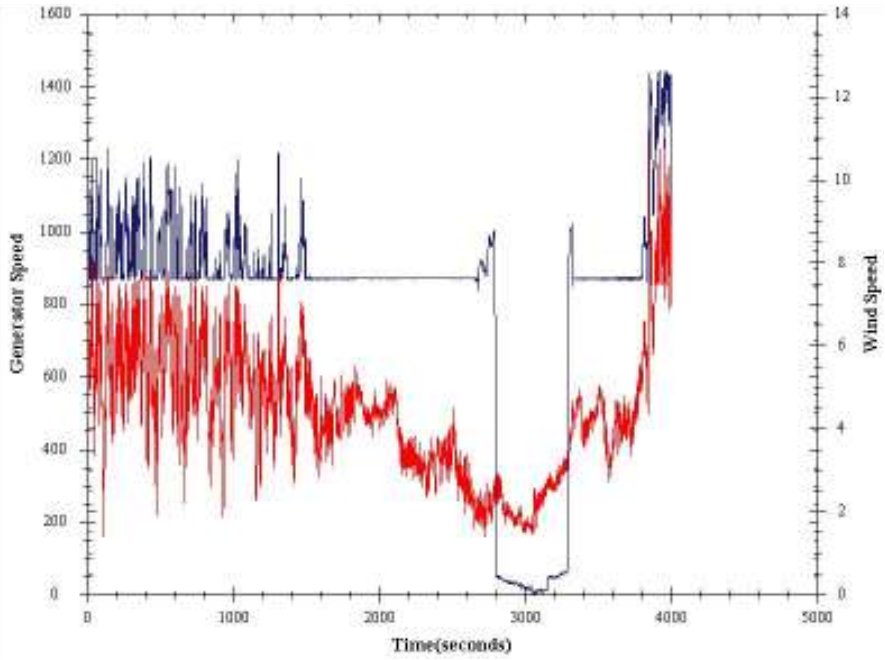

Figure 43. The first 4000 data points for the generator speed of a given wind turbine at 10 second intervals

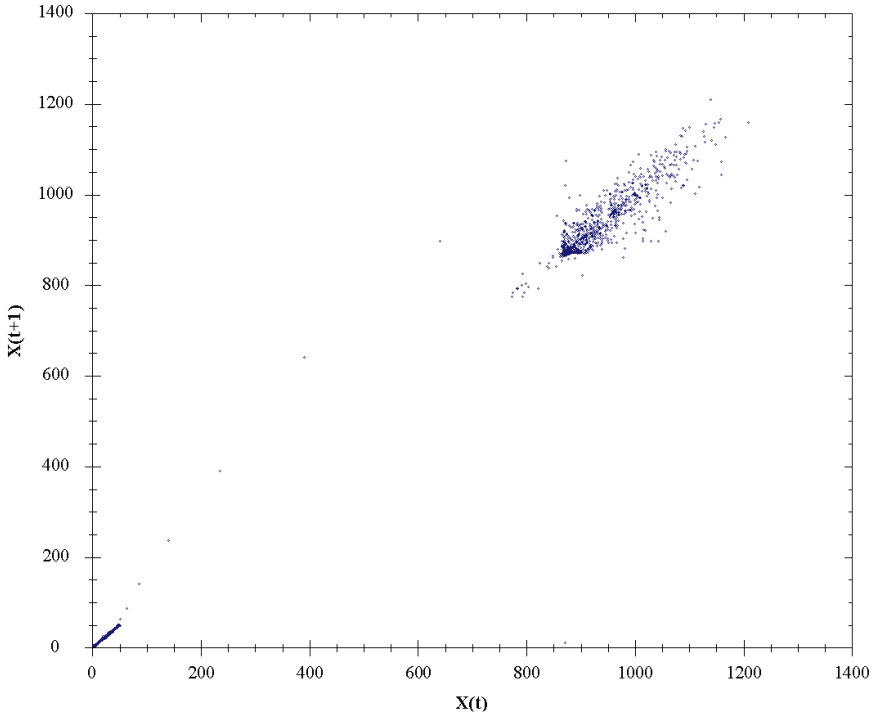

Figure 44. Phase Portrait of first 4000 points for generator speed of a given wind turbine.

The prediction algorithm is used to predict the next 100 points of the generator speed for the given turbine. The Lyapunov exponent for the first 4,000 points of the generator speed was calculated to be 0.17662 and the optimal constant for the prediction algorithm, given in Eq. (24), was generated by the modified evolutionary algorithm to be -3.6091. Table 4 illustrates the prediction accuracy for the first 10 points of the prediction.

Table 6. Prediction Accuracy for the First 10 Predictions for the Generator Speed

Actual	Predicted	Absolute Error	Relative Error
869.8	870.7	0.9	0.001
870	869.7	0.3	0.0003
870.5	870.0	0.5	0.0006
868.8	868.6	0.2	0.0002
868.1	871.9	3.8	0.004
871.7	870.8	0.9	0.001
870.1	870.2	0.1	0.0001
870.2	871.4	1.2	0.001
868.7	869.4	0.7	0.0008
870	868.84	1.16	0.001

As can be seen from Table 6, the prediction algorithm does very well with the prediction of generator speed. Fig. 45 illustrates the relative error for the 100 predictions of the generator speed. Clearly, the prediction schema does extremely well with the 100 predictions. The mean absolute error for the predictions of the generator speed was calculated to be 11.95782. The mean absolute error is influenced by the large values of the actual data. The mean relative error better illustrates the error rate for this data set.

The mean relative error for the 100 predictions of generator speed was calculated to be 0.013143 which is a strong validation of the proposed prediction methodology.

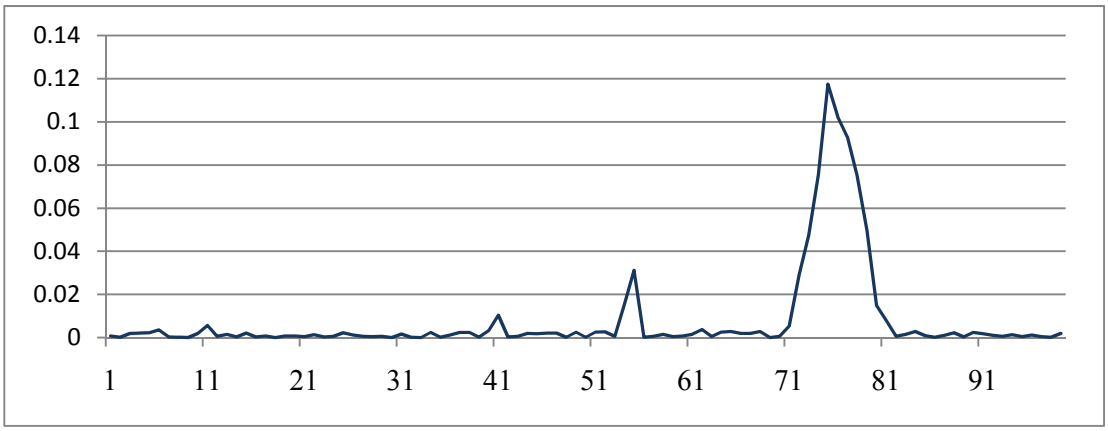

Figure 45. Relative error for generator speed predictions.

Again, a comparison was performed between the three prediction methodologies discussed above. The neural network prediction algorithm was configured as in the previous two examples and produced results with a mean absolute error of 25.6415 for the next 100 points in the wind dataset. For the same dataset and prediction points the nearest neighbor methodology performed better, producing results with a mean absolute error of 14.2319. The results of the neural network and nearest neighbor methodologies are given in Fig. 46 and 47 respectively. The direct comparison of the errors for the three prediction techniques is given in Table 7.

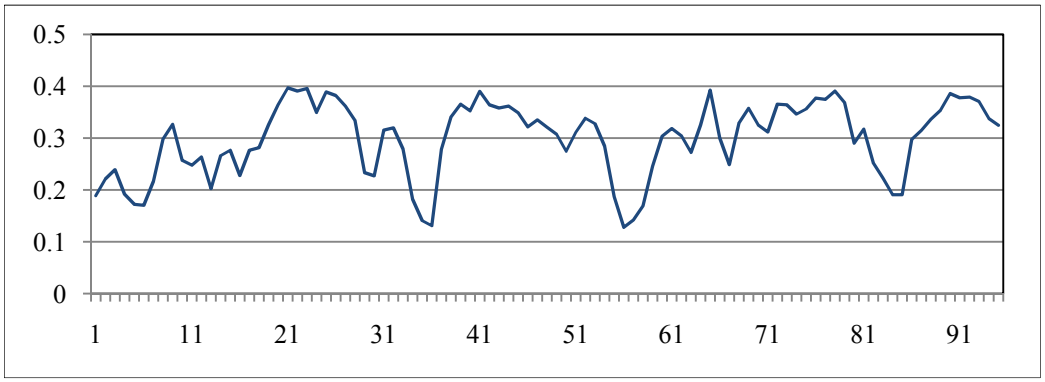

Figure 46. Relative error of the neural network for generator speed predictions.

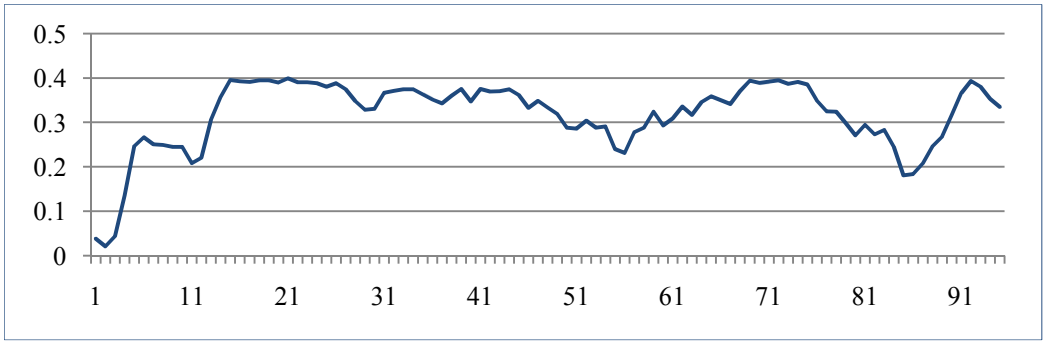

Figure 47. Relative error of the nearest neighbor methodology for generator speed predictions.

Table 7. Comparison of the prediction methodologies mean absolute error for the wind dataset.

Prediction Technique	Mean Absolute Error
Neural Network	25.6415
Nearest Neighbor	14.2319
Proposed Technique	11.9578

It is clear to see from Table 7 that the chaotic prediction methodologies again performed far better than the non-chaotic neural network on the chaotic dataset.

However, due to the nature of the data set, with its divergent regions of values, it is desirable to choose multiple regions of the dataset and compare the predictions of the three methodologies for each region. Table 8 describes the results of this comparison. The first column of Table 8 is the index from which the predictions began in each region.

Table 8. Mean absolute error of prediction of the wind generator dataset.

Starting Index	Proposed Technique	Nearest Neighbor	Neural Network
1001	9.879	9.983	11.388
2001	6.448	7.053	8.981
2901	5.310	6.753	93.513
3501	7.662	8.082	9.564

As can be seen from Table 8, the proposed prediction technique does very well with the wind turbine generator dataset. The technique was able to adequately predict data points in all regions. In the region which produced the most difficulty for the neural network, the region where the data tends to zero, the proposed technique was able to respond with accurate predictions. These results clearly indicate that the proposed technique is a viable prediction algorithm.

4.3 Discussion of Prediction Results

The results of the examples given above indicate that the proposed prediction methodology does well in practice and remains fairly accurate over the length of the predictions to the limit of the Lyapunov exponent. It is also clear that the algorithm does

much better for Lyapunov exponents which are smaller in value, as would be expected. Table 9 illustrates this fact through comparison of the three examples above.

Table 9. Prediction Methodology Accuracy

Dataset	**λ value**	**MAE**	**MRE**
Duffing Map	0.505	0.997	2.153
Henon Map	0.427	0.512	1.433
Generator Speed	0.177	11.958	0.0131

The results given in Table 9 hold great promise for the proposed methodology's accuracy. An additional benefit of the proposed prediction methodology is that it can be customized to the given dataset. Should a dataset produce prediction errors larger than desired it is possible to increase the frequency of the calculation of the constant *c* by the evolutionary algorithm. Additionally, the computation time for the predictions is moderate, relative to the dataset. Thus, the algorithm is appealing for control strategies as well as basic predictions.

CHAPTER 5. SYSTEM STATE CHANGE AND ANOMALY DETECTION IN CHAOTIC SYSTEMS

Many domains, such as industrial control systems, health care, and computer networking, are concerned with system state change detection. The early detection of system changes can assist in preventing catastrophic failure in critical systems such as the power grid. Additionally, state change detection can warn of such issues as cyber attacks on computing networks. The detection of these phenomena can be a challenging task. However, this task becomes more difficult when the system under question is represented by deterministically chaotic data.

The literature contains many references related to anomaly detection in linear, linearized, and stochastic systems. As examples, in the computer network/cyber domain research has been presented by Patcha *et al.* [54], Wang *et al* [55] and Fang *et al* [56]. Work was also reported in the medical domain such as the survey presented by Chandola *et al* [57] or the work of Chuah *et al* [58]. The state change detection domain is also well represented for the linear, linearized and stochastic realms. Azad *et al* [59] presented research related to state change detection in active and inactive systems and Radke *et al* [60] presented a survey of change detection techniques for the image domain. Unfortunately, the techniques of linear and stochastic state change, detection algorithms are insufficient for use on deterministically chaotic systems.

Rapid advances in technology, especially in safety critical domains, have resulted in a greater need for accurate description and anomaly detection of deterministically chaotic systems. Domains such as the Smart Grid [7], alternative energy control, advanced cryptographic communications and even the human heart are driven by

deterministic chaos and require accurate state characterization and early detection of forthcoming anomalies.

Limited research has been conducted in the domain of chaotic system state change detection. Tykierko [61] presented work which utilized changes in invariant metrics related to the chaotic system, such as the fractal dimension or the maximal Lyapunov exponent, to detect state changes. Chakraborty *et al* [62] proposed the use of symbolic dynamics filtering for anomaly detection. Ref. [62] was extended by Rao *et al* [63] in their review of the topic. However, these methodologies lack the ability to be readily visualized by the user and are computationally intense, therefore inappropriate for real time implementation.

This section presents a system state change detection algorithm specifically for systems exhibiting deterministic chaos. The presented material takes advantage of the reconstructed phase space of the chaotic system through the course grained Ergodicity Plot and a novel transition matrix. The novel transition matrix can be utilized to detect system changes as new data is streamed into the algorithm. As will be shown, the proposed algorithm can be used in real time systems due to its extremely small computational footprint and is highly effective at detecting small changes in a chaotic system as well as direct anomalies. Further, the proposed algorithm is tunable with user defined parameters to assist in reducing false positive rates.

5.1 Ergodic Transition Matrixes

It is possible to utilize the new Ergodicity Plot to better understand mathematically the regions of phase space which are visited by the orbits of the chaotic

system. To do so, a matrix of the transitions can be generated. Transition probability matrices are not novel in of themselves. The Markov Chain has been well studied and well represents the transition probabilities of a system [64]. However the matrix that will be generated here diverges from the standard Markov model through the use of the local Lyapunov exponent of the region represented by a member of the partition of the previous section to generate a transition measure rather than a transition probability.

The calculation of the local Lyapunov exponent is performed as given in Chapter 2, with only the data in the local region being considered. With the local Lyapunov exponent calculated for each region in the partitioned embedding phase space the Ergodic Transition Matrix can be generated as follows. Consider a equivariant partition *I* on the embedding phase space resulting in a $\frac{1}{\sqrt{n}} \times \frac{1}{\sqrt{n}}$ square matrix. For each cell, $m_{i,j}$ of the matrix, calculate the probability, ρ, that the orbit transitions from the *i*th region of the partitioned phase space to the *j*th region. Then the ergodic transition measure *etm*(●) for the cell $m_{i,j}$ is given in Eq. (28). The appendix of this thesis contains a pseudocode method of computer software for the creation of the ETM as described here. Table 9 illustrates a small portion the Ergodic Transition Matrix for the data used to generate the EP for the Cedar Rapids, IA temperature data discussed in Chapter 3 and illustrated in Fig. 34.

$$etm(m_{i,j}) = \rho\lambda \tag{28}$$

Table 10.Portion of the Ergodic Transition Matrix for the EP in Fig. 34.

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0.000779	0.000779	0	0	0	0

As can be seen in Table 10, The Ergodic Transition Matrix is often a sparse matrix. This is extremely beneficial though when seeking to perform real time analysis as will be shown in Chapter 6 with case studies. To facilitate the visualization of the entire Ergodic Transition Matrix the matrix is plotted as a surface plot in Fig. 48.

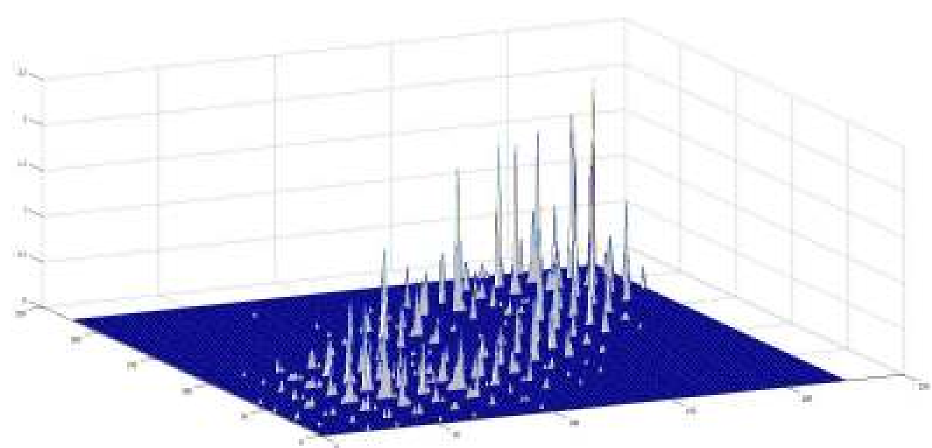

Figure 48. Surface Plot of the Ergodic Transition Matrix for the Data in Fig. 16.

The Ergodic Transition Matrix (ETM) is highly effective at representing the manners by which the chaotic system traverses the embedding phase space. More importantly though, the partitioning of the embedding phase space can be viewed as the generation of system states where each member of the partition represents a specific system state. Then, the ETM can be viewed as representational of the traversal of system states by the chaotic system. As such it can be used for system state change and anomaly detection as presented next.

5.2 Change Detection using the ETM

The ETM is a concise and manageable representation of the ergodicity of the chaotic system. As such, it contains useful information pertaining to the normal operating conditions of the chaotic system under consideration. It is possible to utilize the ETM to determine when the chaotic system is experiencing small state changes which could indicate the early occurrence of anomalous features (e.g. changes in the system states which indicate system degradation, possibly to failure). This section defines the use of the ETM for state change detection. Actual case study examples of this methodology are left for presentation in Chapter 6.

Consider a chaotic system which is currently operating without anomalies or outliers and is not experiencing changes in the system states. The attractor representing that system is representational of the system's normal operating conditions. As such, it is possible to form the EP and ETM from the system's attractor of normality. The ETM generated from the system's attractor, and EP, forms a baseline pattern of trajectory

traversal within the embedding phase space and may be used as a benchmark by which future traversals may be measured. Define this type of ETM as the ETM of Normality, ETM_{Norm}. It should be noted that, as with all machine learning techniques, the larger the training, or in this case normal, dataset available, within a reasonable limit, the more accurate the model will be. Thus, a sufficient sized sample of the normal operating states is desirous for the algorithm; however, this is subjective and dependent upon the system under consideration.

Given the ETM of Normality, ETM_{Norm}, for the system under consideration, it is possible to stream new data into the algorithm and compare the ETM of the new data to the ETM_{Norm}. To facilitate the comparison, a buffer is maintained, in a moving window method, of the new streaming data, as single data points do not facilitate the generation of an ETM. The size of this buffer is system dependent but a sufficient buffer size consists of 4-5 complete orbits of the chaotic system. The ETM of the streaming data is created each time a complete (mean) orbit has been added to the buffer. As the new orbit is added to the buffer, the previous first orbit of the buffer is removed to facilitate maintaining a constant buffer size. The absolute difference between the ETM for the buffer, ETM_{Buf}, and the, ETM_{Norm} represents the degree of system state change, Δs, as given in Eq. (29).

$$\Delta s = \left| ETM_{Norm} - ETM_{Buf} \right| \tag{29}$$

The degree of system state change, Δs, can be used to warn of systems which may be experiencing shifts in their operating states, as these phenomena will appear as differences between the buffer ETM from the ETM of Normality. It is possible to set a

threshold, ϵ, based on the degree of state change metric, Δs, such that a warning is given for state changes $\Delta s > \epsilon$. The degree of system state change metric is sensitive to even small changes in the system operation and is therefore highly effective for state change detection. The use of the alarm threshold affords the tuning of the state change detection algorithm to reduce false positive rates.

The buffer of the detection algorithm forms a moving window as new data is streamed into the system. Therefore, it is possible to continue to compare new data as it is encountered. Additionally, it is possible to evolve the ETM_{Norm} to include recently encountered data, such as may be desired for systems which may include a break in period. In such systems it would then be more desirable to consider the initial data as transient to the actual ETM_{Norm}. Obviously, evolution of the ETM_{Norm} is system dependent.

Due to the compact size of the ETM, as a real valued matrix, it is easy to see how this methodology could be used in a real time monitoring system with little computational overhead. The ETM_{Norm} is only a matrix of state change metrics, double precision, which is easily stored in resident memory for even the smallest of systems. The buffer ETM is easily generated with little overhead as well, making the proposed algorithm a very attractive near real-time detection system. The appendix of this thesis contains a pseudocode method for a computer program to perform anomaly and system state change detection as described here.

In conclusion, this section has presented a novel anomaly and state change detection technique. The validity of this technique on actual chaotic systems is given in Chapter 6. The technique has been shown to be computationally simple and maintains a

small system footprint allowing it to be utilized in a near real-time environment on smaller embedded devices.

CHAPTER 6. CASE STUDIES

This section illustrates the effectiveness of the techniques proposed in this thesis through case studies in various domains. The case studies presented here represent real world systems and actual collected data. However, due to the sensitivity of some of the data presented, any information reflecting specific individuals has been cleansed and represented by discrete naming conventions which protect individual identities. The studies utilize one or more techniques from the material presented in the previous sections of this thesis.

6.1 Classification of MMO Users through Ergodicity Plots

This section describes research conducted in the domain of motif detection and association in Massive Multiplayer Online (MMO) environments. This research included classifying user types through network motif pattern recognition techniques, comparing these motifs to chaotic attractors reconstructed for each user, and determining if any such comparisons could be used as classification techniques in their own right. This case study illustrates the ability of chaotic attractors and EPs to be used as classifiers for instances of chaotic systems.

The dataset used for the research conducted was gleaned from publicly available Internet Relay Chat (IRC) logs for the online game Eve Online®. The dataset represented individual messages posted to the IRC logs by members of the gaming community. There were 274 channels from which the data was taken. Combined, there were 1,075,490 messages logged. Each message consisted of a timestamp indicating when the message

posted to the IRC channel, the username (avatar name) of the user posting the message, and the message content.

The data obtained from the IRC logs represented a time span beginning on November 2006 and ending on May 2010. These logs were parsed into a SQL server database to facilitate easy retrieval of the data. In addition to the simple parsing of the data, a database of id numbers for each user (avatar) which consisted of simple integer values was generated. This was done in order to preserve privacy of the individual users during the course of the research. Additionally, a database of relations was generated to indicate the relationships between users in the IRC channels. These relations were generated based on temporal proximity to a given message. Therefore, distinct users who had posted messages to the IRC channel in a given time prior and following the posting of a message are considered to have a relationship, albeit implicit.

A network diagram was generated for each user in the database using an open source network software tool. A visual inspection of these diagrams revealed patterns which could be used for classification. Due to these patterns being detected the author undertook a pattern analysis of the network diagrams and discovered a number of distinct classes of patterns. For the sake of the present study, the number of classes was limited to be three; however this is a user selected parameter and can be changed. These classes were related to the user's role within the game. The three classifications that emerged were the leader role (one who leads missions within the MMO environment), workers (one who follows instructions and works at a low level to perform tasks during MMO missions) and finally spies (one who communicates with both their own team and the opposing team in the MMO environment). These classifications were confirmed through

polling users of the IRC logs in this particular MMO environment. Fig. 47 illustrates the network diagram for a user classified as a leader, while Fig. 48 shows a follower and Fig. 49 a spy.

As can be clearly seen in Fig. 49-51 there are distinct relationship link motifs for each class of user. Leaders link to a select few other users in the IRC channels[4] whereas workers link to a large number of other users. It is clear from Fig. 51 that users classified as spies link to two or more distinct user groups. Assumptions concerning different IRC channels and chat rooms were taken into account in developing these diagrams and do not influence the classifications presented herein.

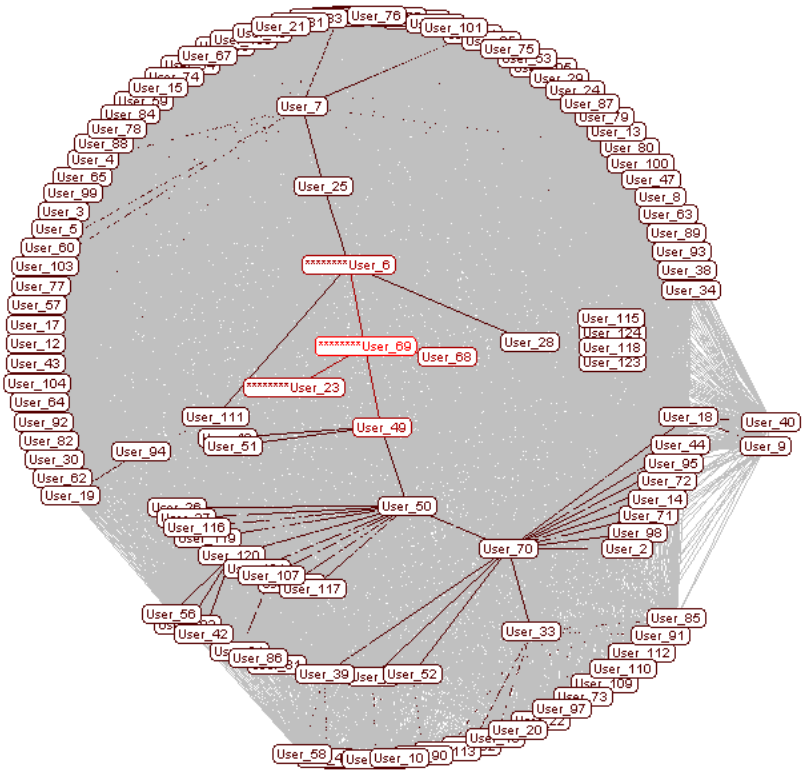

Figure 49. Network Diagram for a user classified as a leader.

[4] For the game known as Eve Online it has been shown that leaders are most often utilizing voice communications such as Skype rather than IRC chat.

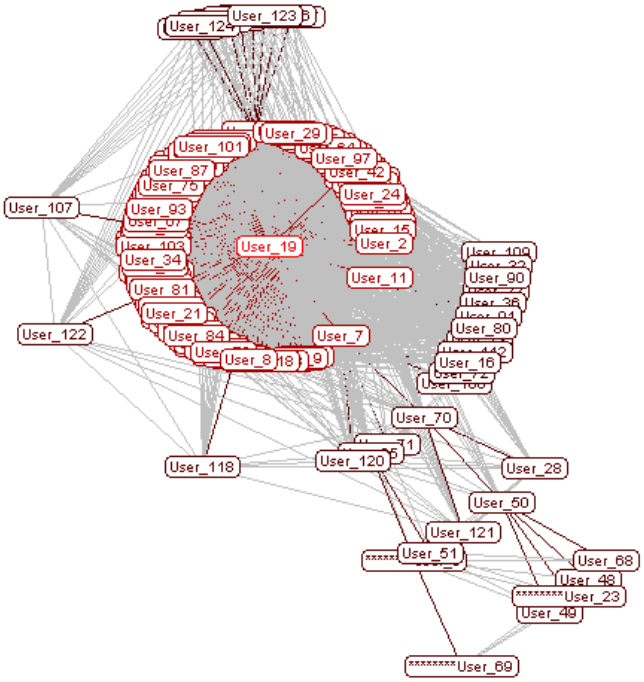

Figure 50. Network Diagram for a user classified as a worker.

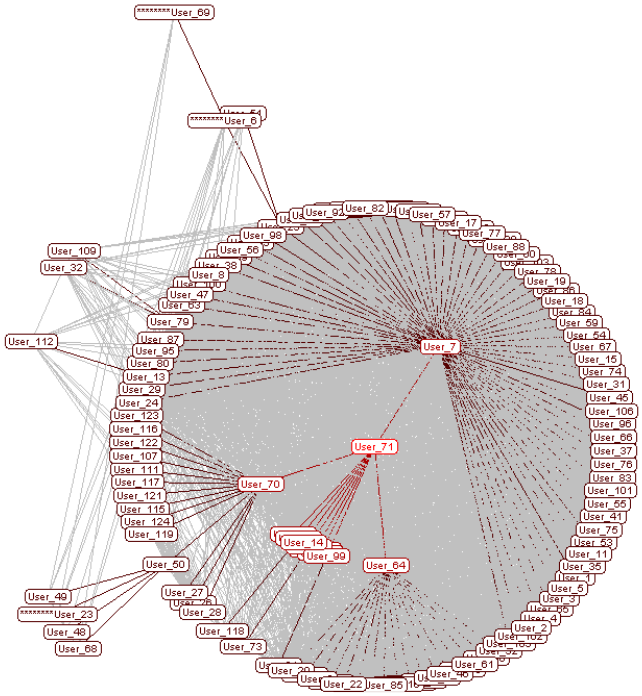

Figure 51. Network Diagram for a user classified as a spy.

From the network diagram classification it was possible to determine specific network motifs. These motifs were discovered during the pattern recognition phase of this research. The motifs alone are sufficient for correctly classifying the network diagrams. It was discovered that this classification was accurate to above 80% for the data tested. The IRC logs indicated that some users only communicated once or twice during the period considered and as such these may have been misclassified.

In addition to performing a pattern discovery analysis of the network diagrams, research was performed to understand if the data representing user chats in IRC channels was nonlinear (deterministically chaotic). To determine if the user data in the IRC channels comes from a deterministically chaotic distribution a dataset was created of the user data which utilized the timestamp and the length of the message. These values form a vector for each communication event. The timestamp is transformed to represent the hour in which the message was sent but the temporal order is maintained during the transformation. The text length is used to represent the size of the message. The message content is not used for this research which presents a large benefit for the analysis of large datasets.

Transforming the IRC data into embedding phase space utilizes the embedding theorems discussed in Chapter 2 of this thesis. This results in visualization (in 2 or 3 dimensions in these cases) of the chaotic system. The ergodicity of the orbits in the trajectory offer a unique classification point that will be discussed shortly. Fig. 52-54 illustrates the chaotic attractors for the users whose network diagrams are given in Fig. 49-51.

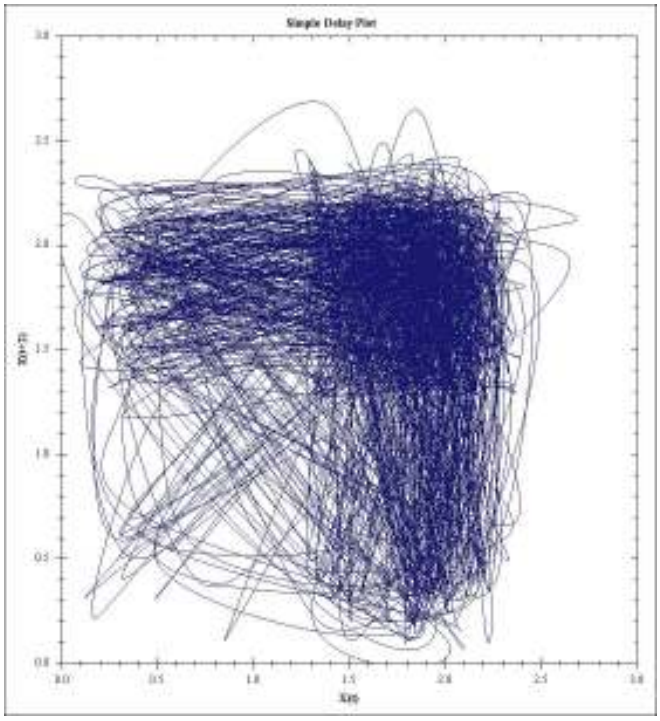

Figure 52. Chaotic Attractor for user in Fig. 47.

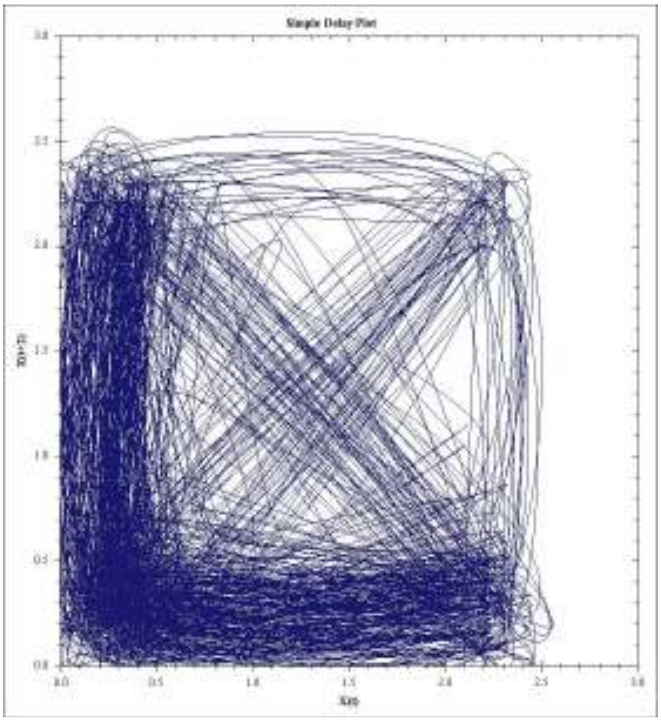

Figure 53. Chaotic Attractor for user in Fig. 48.

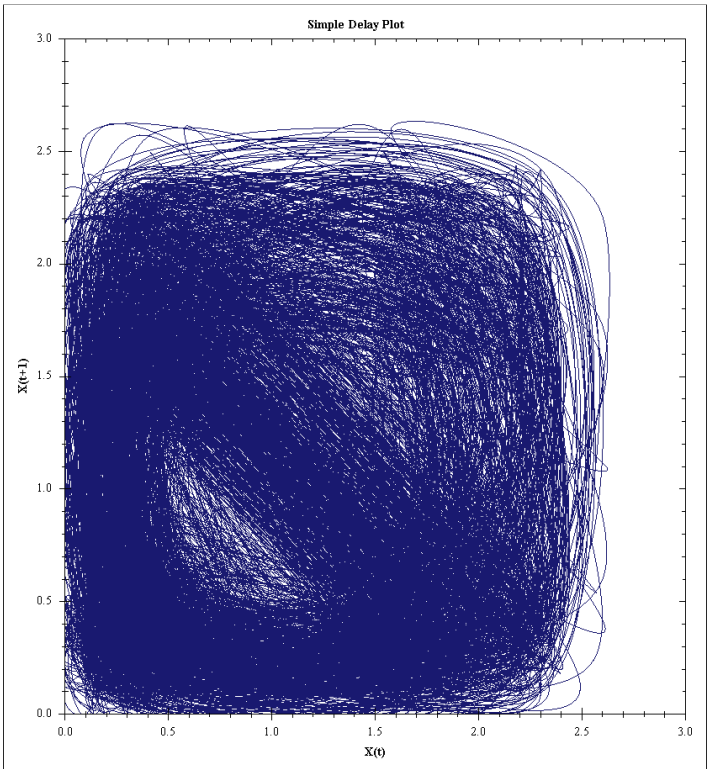

Figure 54. Chaotic Attractor for user in Fig. 49.

As can be seen in Fig. 52-54, the attractors for each specific type of user are distinct. This uniqueness between classes was discovered to exist in common for the users when classified by their network diagrams. Since the network diagrams and the chaotic attractors represent two distinct types of information from the dataset it is possible to conclude that the classifications are distinct enough as well.

The plots in Fig. 52-54 are difficult to read and extract understandable information from. However, these plots can be used as another form of classification. It was discovered that leaders communicated with short directed messages while workers had a tendency to communicate with longer, more disjoint, messages and for a longer

time. The users classified as spies tended to closely mimic the workers with the addition of some of the leader characteristics. This discovery is seen graphically in Fig. 53-55.

Since the data from the IRC channels if from a chaotic distribution it is ergodic. To better visualize the embedding phase spaces, Ergodicity Plots were constructed by following the methods described in section 3 of this thesis. The EP is a condensed version of the chaotic attractor and is suitable for rapid classification of users based upon communications data. Fig. 55-57 illustrates the Ergodicity Plots for the attractors given in Fig. 52-54. The EPs of Fig. 55-57 have been colored red to highlight their differences from the chaotic attractors.

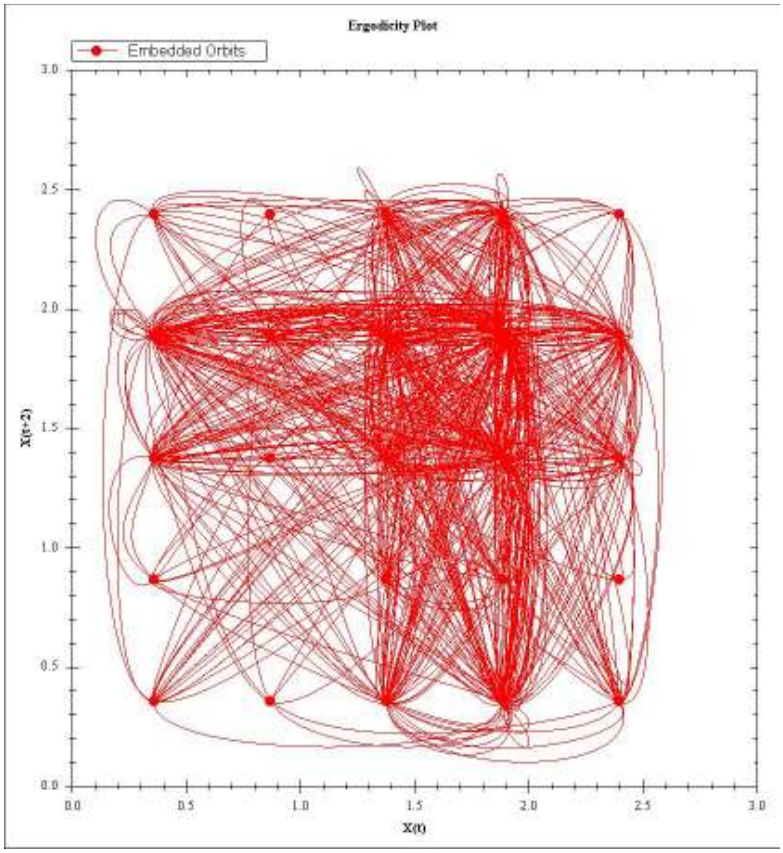

Figure 55. EP for the attractor shown in Fig. 50.

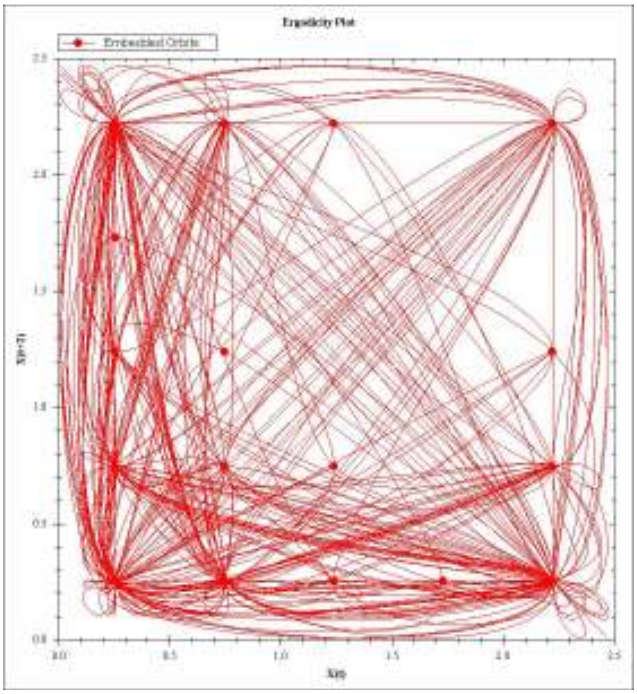

Figure 56. EP for the attractor shown in Fig. 51.

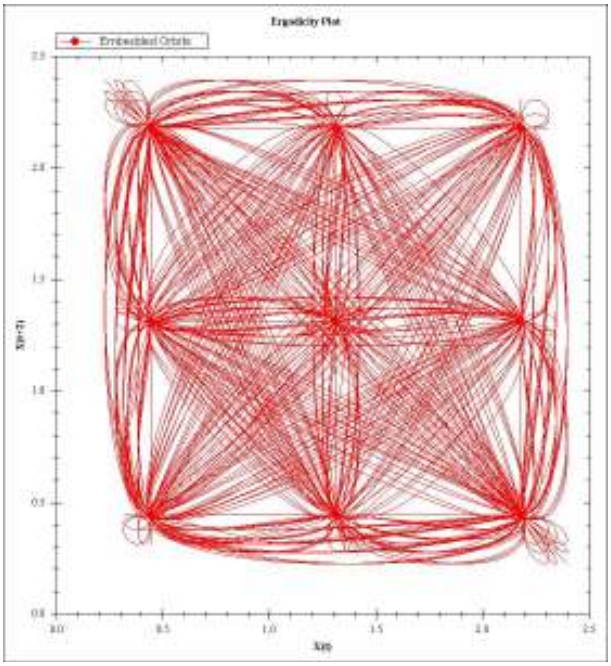

Figure 57. EP for the attractor shown in Fig. 52.

The research discussed in this section has shown that using simply the time stamp and message size of a log of messages for a specific user it is possible to classify the user as a leader, a worker or a spy (as well as other classes defined by the user) in the MMO environment. The computation time of this classification is greatly reduced over semantic mining of the tests of communications thus offering an added benefit to the algorithm. Further, the work presented in this section has highlighted an application of the novel Ergodicity Plot (EP). The EP is easily utilized in a classification, or pattern recognition, scenario in a far more efficient manner than the simple reconstruction of the attractor of the nonlinear system. The use of the EPs for classifying the users in the MMO environment presented an accuracy of 81%. This compares very well with the computationally more intense method of developing network diagrams for each user.

6.2 Chaotic Attractors for Cyber Security

Cyber security is critical for uninterrupted functioning of our government, private and public enterprises. Realizing the cyber security threat, the Cyber Command has been created by our government to protect our infrastructure. Protection of such national infrastructure depends upon real time detection of cyber threats and subsequent remedial action. Extensive resources are being leveraged by government and private enterprises for cyber security but the complexity of the problem requires new ideas to be developed and implemented to overcome it. Many companies face cyber attacks daily because of the nature of their business. This section presents the theory and results of a unique research project to develop an innovative attractor model for cyber security.

Today's advanced, targeted malware typically goes undetected by commercial anti-virus software. The Advanced Persistent Threat (APT), in particular, is a classification of adversary which uses social engineering and sophisticated malicious code to gain a persistent presence inside an organization and, over time, to exfiltrate sensitive and proprietary company information [65]. Since these attacks change rapidly and are highly targeted, knowledge of past attacks is not sufficient to prevent future attacks. In particular, the traditional reactive approach of creating and using security patches does not root out the next threat. Rather, a robust and dynamic model of normal behavior (for a computer, user, network, etc.) is needed in order to identify malicious or abnormal patterns of behavior. Most approaches to modeling system behavior do not take into account the non-linearity (i.e., time dependence) of the data. The proposed methodology, however, will leverage deterministic chaos to learn the behavioral norm for systems in an adaptive fashion. The proposed evolutionary attractor model adapts as new information comes in. This case study project uses the theory of deterministic chaos and chaotic attractors to develop an adaptive model of normal behavior for individual user email. The intent is to use such models to detect behaviors that lie outside the region of normal behavior, thus identifying sophisticated attacks such as the APT attack in real time.

6.2.1 Normal Cyber Activity

Users operate in the cyber domain as part of their daily activity. Various tasks are accomplished in this domain including communications, planning, productivity, transactions, and entertainment. By nature, users perform these actions within some relative pattern. These patterns are governed by many factors including sleeping habits,

family life, and work schedules. These patterns are inherently nonlinear (time-dependent) due to many intervening events. This section discusses the exploitation of these patterns for determining a strange attractor which represents a user's "*normal"* cyber activity and how that normality may be used to detect cyber attacks.

Consider a user in a corporate environment. On a given day that user checks email, surfs the Internet, interacts with software (causing file I/O operations to take place), sends instant messages, checks social networking sites, banks online, and so forth. Those actions can all be considered linked to cyber activity directly or indirectly. Monitoring these activities in some form of a log file generates a time series data set which can be analyzed for nonlinearity. It is hypothesized that while exact activity at the same time every day is an unlikely event, certain forms of cyber activity will be conducted in relative proximity to a pattern of some given time interval. This pattern forms a strange attractor.

Translation of the monitoring log into values that can be utilized by a nonlinear analysis is not a trivial task. Email activity is logged containing parameters for the sender's email address, the recipient's email address, the size of the email in bytes, the number of attachments, the time it was sent, the GMT time offset, and the content of the subject line. The majority of this data is non-numeric and therefore must undergo some transformation, or discretization, to be utilized in the reconstruction of a nonlinear phase portrait for that user.

Email addresses can be transformed simply through the assignment of a unique identifying integer or through more complicated means such as clustering by domain. The size of emails and number of attachments already represent a real-valued

measurement. The GMT time offset is also an integer value and can remain as such. The subject line transformation poses an interesting problem as it is not easily transformed into a real value. To transform the subject line into a real value, a cluster analysis can be performed on all of the subject lines in a given training dataset. Each subject line will therefore belong to a specific cluster. The number of that cluster can be used to represent the subject line for the email under consideration. Time should also be truncated to an integer value for the hour of the day to avoid setting the granularity of the model too fine. Other forms of cyber activity may be transformed similar to email activity. Once the transformation of the cyber activity logs into real-valued space has been performed, the data may be analyzed to determine if the activity represents a deterministically chaotic system. This determination was performed following the methods given in Chapter 2 of this thesis.

Knowing that the system is deterministically chaotic indicates a strange attractor can be recreated in embedding phase space to properly represent the dynamics of the system. The attractor reconstruction is performed as explained in Chapter 2 of this thesis. If the data used to reconstruct the attractor is known to come from normal cyber activity and that there were no cyber attacks recorded in the data, then the strange attractor can be used as the boundaries of normality for the training data. This normality can be associated with a single user if the training data represents the cyber activity of that user only. Each user will have a unique normality attractor. This is a fact that should be exploited to monitor for cyber attacks.

A cyber attack can be detected as an outlier to the attractor of normal behavior. A point would be considered an outlier if it is outside the boundaries of the attractor of

normality for the user under consideration as discussed in Chapter 3 of this thesis. However, not all points that fall outside the attractor will be from a cyber attack. Often times a user's work details changes, this will have an impact on normality for that user. Hence, points outside the bounds of the attractor are flagged as anomalous and monitored. Should further activity, similar to the anomalous behavior, be detected the attractor can be evolved to include that activity. Other heuristics may also be used. Initially, this would be a manual process to ensure proper characterization of the anomalous behavior. Should no further activity correlate to the anomalous detected activity, then that activity is most likely something that should be considered for deeper analysis by a technician.

The monitoring of cyber activity for outliers to the bounds of the attractor could be adjusted such that the number of false positives being reported is reduced. Conversely, the system could be adjusted to report every anomaly without evolution of the attractor. Evolution and reporting are fully customizable by the end user to assure satisfactory results.

6.2.2 Case Study Results

For this project, 28 users were chosen as test subjects. The dataset collected for building the attractors contained metadata on email into and out of a government contractor company for each individual user. A training set of two months' worth of email was collected for each user and was used to create chaotic attractors for each user. The email address parameter and the email size parameter were large enough that they could have overwhelmed the remaining parameters of the email data. Therefore, all parameters are normalized such that they reside in a similar (albeit not the same) interval

to avoid this problem. Additionally, the reconstruction of phase space is performed on the norm of the vector of all the parameters for each email as opposed to individual parameters alone. This allows for a better understanding of the data in general and a cleaner attractor representation for detection. With the norm calculated it is possible to determine if the system is truly nonlinear by calculating the maximum Lyapunov exponent of the data. Fig. 58 illustrates the raw data that has been normalized for the emails of a specific user. The Lyapunov exponent of this data was calculated to be 3.4328 which clearly indicates a chaotic system.

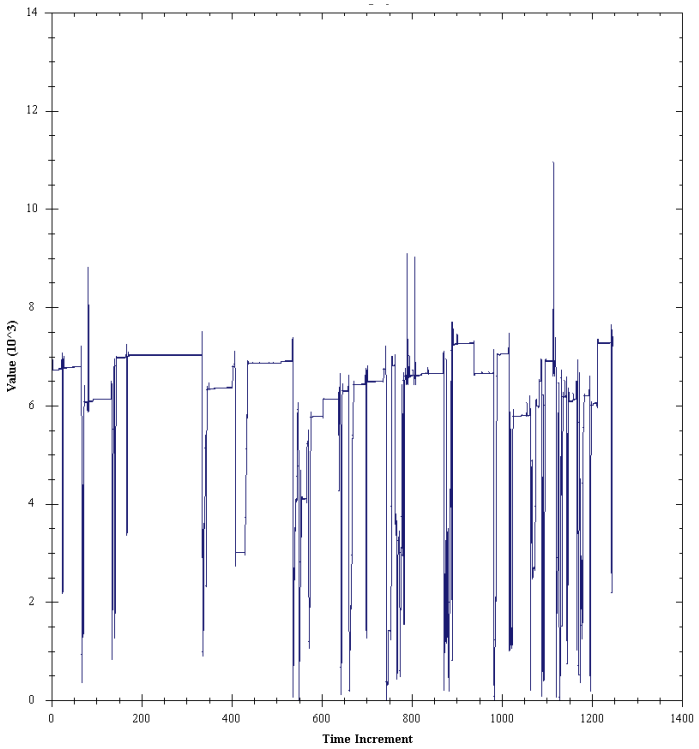

Figure 58. Raw normalized, timestamp ordered email data for a single user.

Fig. 59 displays the embedded phase space for the data in Fig. 58. It is easy to see the ergodicity of the cyber behavior of this user from the repetitious revisiting of specific regions within the phase space. New emails were added to the attractor generated in Fig.

59. As this new email data was evaluated, points which were outside the bounds of the attractor were automatically flagged as anomalous and therefore highlighted to be monitored. Fig. 60 depicts this activity and the flagged data points. Since the emails identified here were responded to and were from the same (new) sender email address, the emails are likely legitimate. It may be desirable to include this as a rule for when to include outliers and evolve the attractor. Fig. 61 indicates another email attractor for a separate user. In this figure the outliers are identified and labeled.

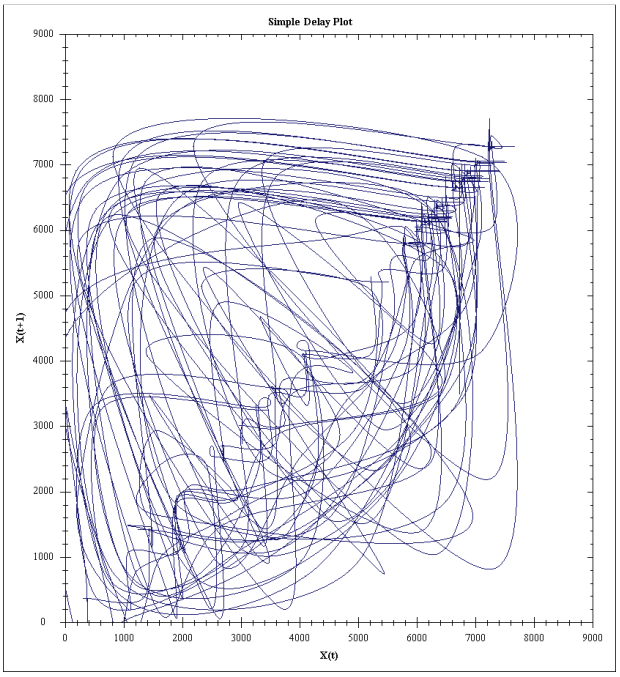

Figure 59. Embedded Phase Space for the data in Fig. 56.

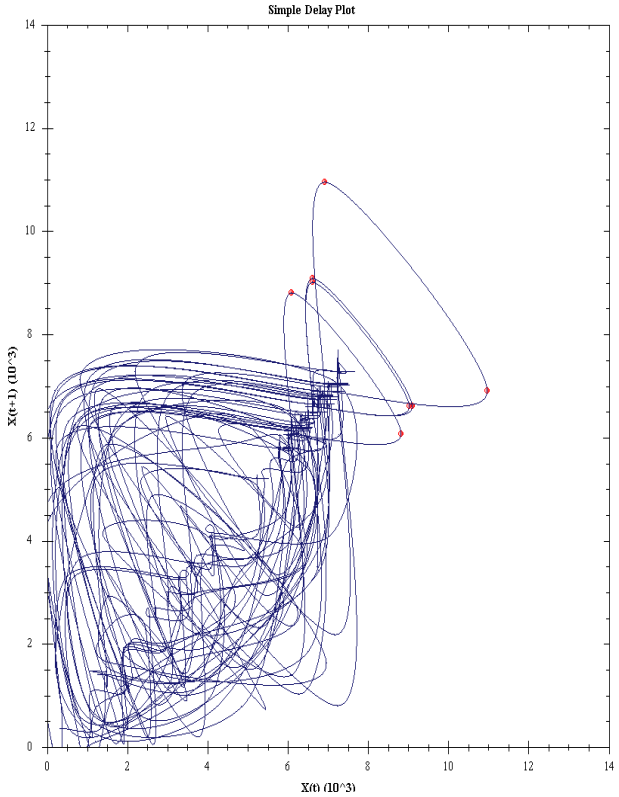

Figure 60. Email Attractor with detected anomalies.

This section has presented the results of a cyber security project. An explanation of the use of strange attractors to model normal cyber behavior was given. It was successfully shown in this case study that cyber activity is deterministically chaotic. Further, the novel methodologies of section 3 of this thesis were supported through the detection of email anomalies in the cyber data set.

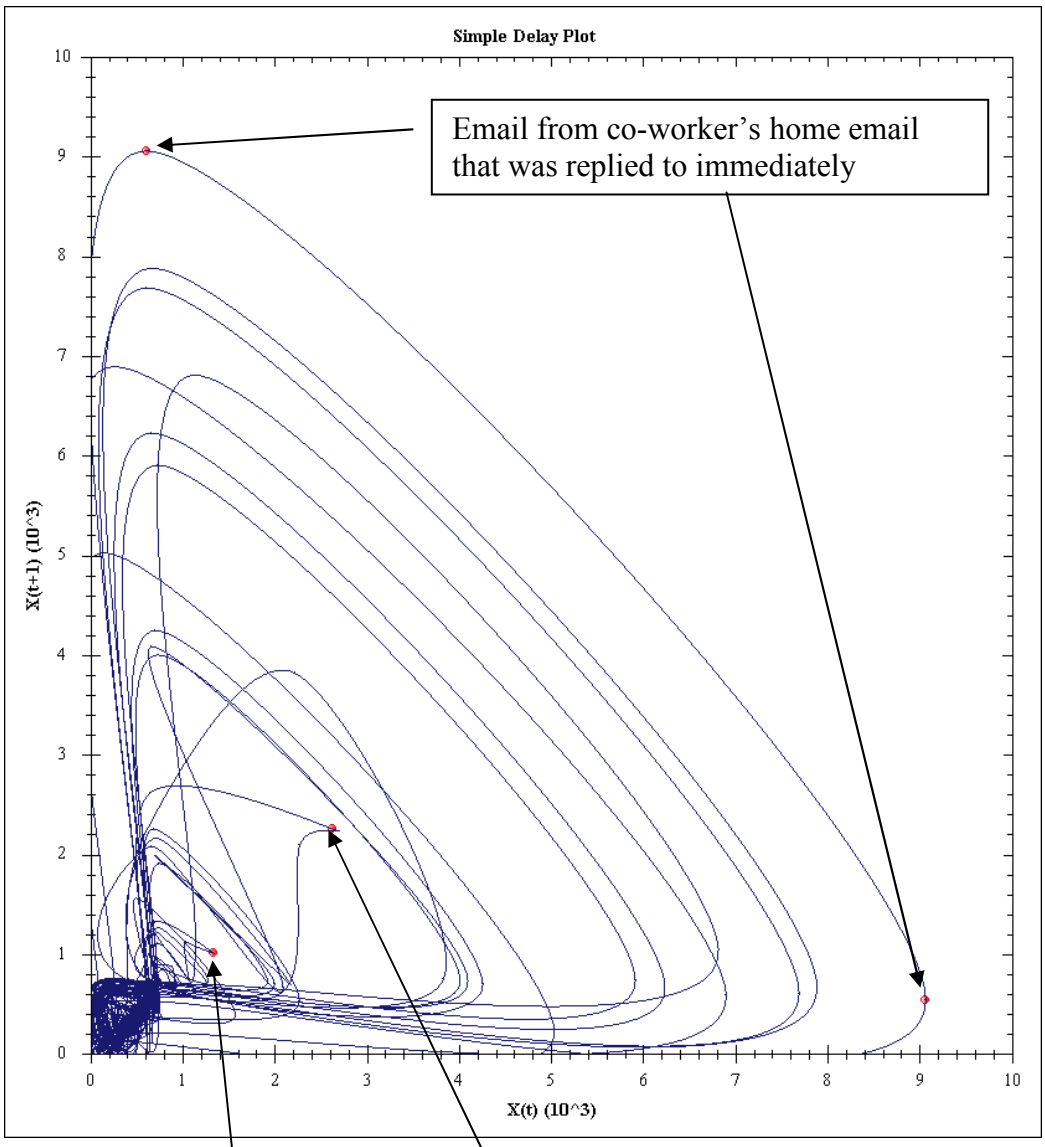

Figure 61. Email attractor with outliers described.

6.3 ETMs for Anomaly and State Change Detection Case Study

This section presents case studies supporting the proposed system state change detection algorithm described in Chapter 5.2. The case studies presented use data from both mathematical difference equations and real world systems. The difference equations considered in this section are the Duffing Map and the Henon Map. The real world systems observed here consist of a cyber security dataset and a dataset captured from existing wind turbines in service.

6.3.1 EMTs for the Duffing and Henon Maps

The Duffing map, given in Eq. (12), is a well known chaotic system. In order to represent the effectiveness of the proposed detection methodology a set of 4000 iterations is used as the initial training set forming the ETM_{Norm}. The first 4000 iterations of the Duffing map were illustrated in Fig. 23 for the x axis. The ETM_{Norm} for this initial set was generated, using a 15 by 15 square partition, and is plotted as a contour map in Fig. 62.

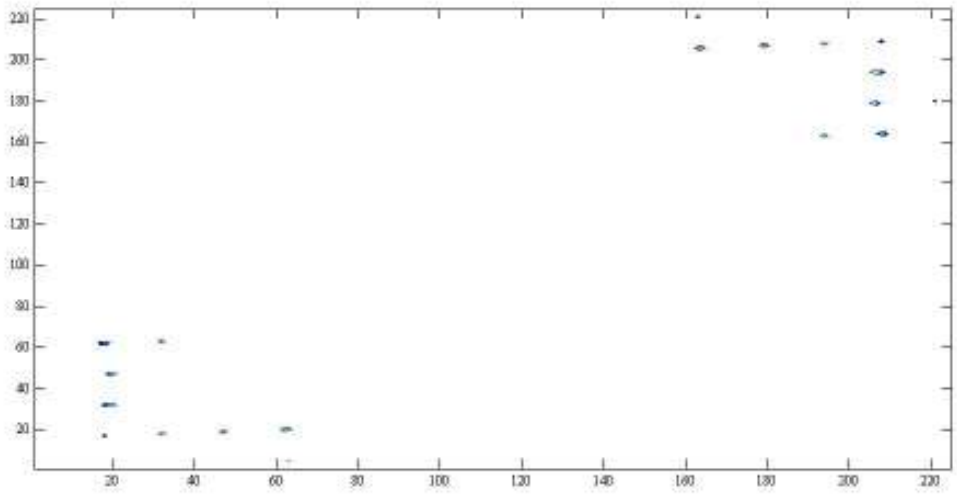

Figure 62. Contour plot of the ETM of Normality for the first 4000 iterations of the Duffing Map.

As would be expected for the Duffing Map, the majority of the transitions take place in the periphery of the attractor. For the purposes of simple change detection illustration, the second 4000 points of the Duffing Map were generated and ran as streaming data to the change detection algorithm. The mean orbit length of the Duffing Map's EP for a phase space generated with delay $\tau = 1$ and embedding dimension $d = 2$ was calculated as 219 iterations. Therefore, a buffer window of size 876 iterations was used to compare the second 4000 iterations to the first. Table 11 illustrates the mean differences, δ_{mean}, minimum differences, δ_{min}, (greater than zero) and the maximum differences, δ_{max}, between the ETM_{Norm} and the ETM_{Buff} for the streaming of the second 4000 iterations.

Table 11. Differences between the ETM of Normality and the buffer ETM for the 1st and 2nd 4000 iterations of the Duffing Map

Buffer Number	δ_{min}	δ_{max}	δ_{mean}
1	1.05E-04	0.0186	4.58E-06
2	3.98E-05	0.0211	8.53E-06
3	3.98E-05	0.0452	1.59E-05
4	1.05E-04	0.0532	1.80E-05
5	1.05E-04	0.0727	2.12E-05
6	3.98E-05	0.0612	2.05E-05
7	3.98E-05	0.0383	1.33E-05
8	1.05E-04	0.0163	8.29E-06
9	1.05E-04	0.0320	1.13E-05
10	1.08E-04	0.0457	1.37E-05
11	3.46E-05	0.0293	1.00E-05
12	4.26E-05	0.0174	7.80E-06
13	3.98E-05	0.0271	1.08E-05
14	1.05E-04	0.0467	1.72E-05
15	1.05E-04	0.0467	1.72E-05

To better illustrate the absolute differences between the ETM_{Norm} (first 4000 iterations) of the Duffing Map and the ETM for the second 4000 iterations Fig. 63 plots the absolute difference between the cells of the compared ETMs as a line graph. The plot in Fig. 63, considers an ETM for the entire second 4000 points as opposed to the buffer methodology whose results were illustrated in Table 11. As can be seen in Fig. 63, the absolute differences between the two ETMs are relatively small. This illustrates the ability of the proposed detection methodology to detect even small state changes in the system.

To complete the case study of the Duffing Map the third 4000 iterations of the map were streamed against the initial 4000 iterations in a manner similar to the second 4000 iterations. The buffer size remained constant at 876 iterations. Table 12 illustrates the differences between the $ETM_{Norm}$and the ETM_{Buff} for the streaming of the third 4000 iterations in a manner identical to that of Table 10.

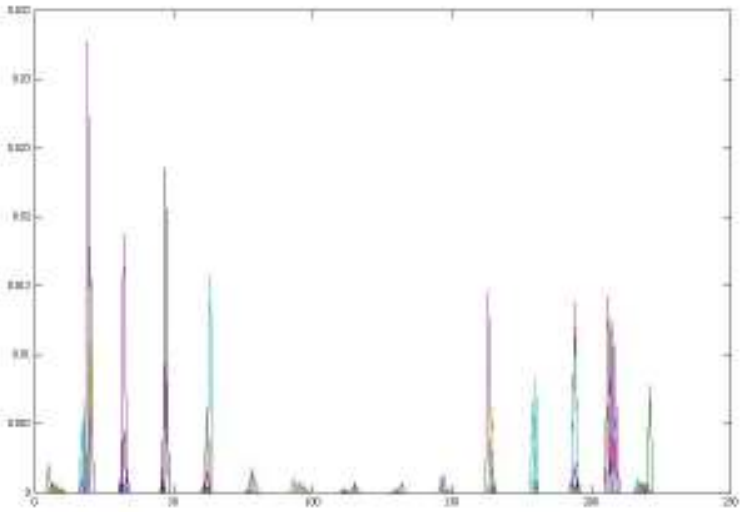

Figure 63. Absolute differences between the ETM of Normality and the ETM of the second 4000 iterations of the Duffing Map.

Table 12. Differences between the ETM of Normality and buffer ETM for the 1st and 3rd 4000 iterations of the Duffing Map

Buffer Number	δ_{min}	δ_{max}	δ_{mean}
1	3.46E-05	7.35E-02	2.00E-05
2	3.98E-05	6.69E-02	1.99E-05
3	3.98E-05	5.12E-02	1.76E-05
4	3.98E-05	3.16E-02	1.16E-05
5	1.05E-04	1.94E-02	7.67E-06
6	3.46E-05	2.25E-02	8.67E-06
7	3.46E-05	2.09E-02	8.27E-06
8	3.46E-05	2.00E-02	8.22E-06
9	3.46E-05	1.29E-02	7.82E-06
10	4.26E-05	1.26E-02	7.09E-06
11	1.42E-04	3.41E-02	1.18E-05
12	3.46E-05	5.15E-02	1.79E-05
13	1.08E-04	5.45E-02	1.91E-05
14	1.05E-04	5.33E-02	1.77E-05
15	6.22E-05	3.43E-02	1.22E-05

The example of the Duffing map has illustrated that the proposed state change detection methodology is highly effective at detecting even small changes to the system. It was shown that the Duffing map does experience slight system shifts as it moves through time and that these shifts can be detected as state changes to the system. However, the Duffing Map case study does not directly indicate the ability of the proposed algorithm to detect direct outliers of the system. To illustrate this ability the Henon map, given in Eq. (14), is used.

The generation of the ETM_{Norm} for the Henon map (with $a = 1.4$ and $b = 0.3$) is performed using the first 4000 iterations of the difference equation for the embedding of

the *x* axis data with a square 20 by 20 partition. Fig. 64 displays the EP for the Henon map and Fig. 65 illustrates the contour plot for the ETM_{Norm} for this system.

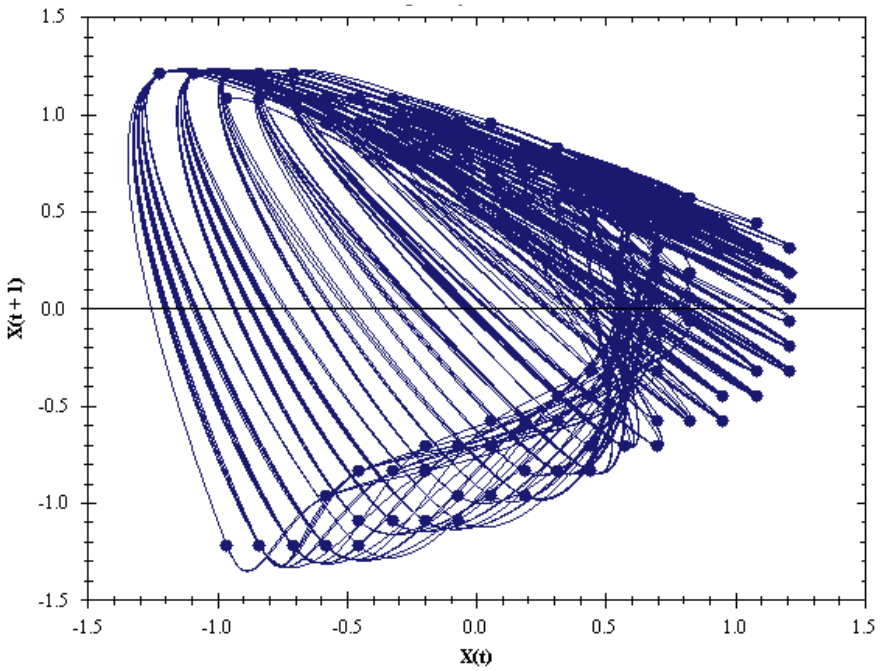

Figure 64. EP of the first 4000 iterations of the Henon Map

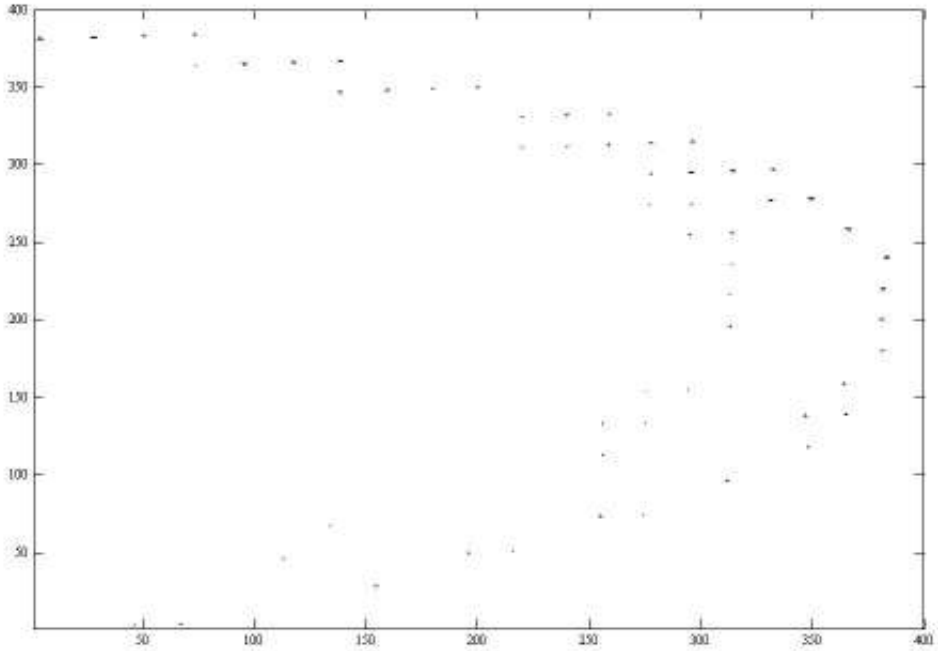

Figure 65. Contour plot of the ETM of Normality for the first 4000 iterations of the Henon Map

To illustrate the ability of the proposed methodology to detect direct anomalies in a chaotic system, a single point is modified in the first buffer of the Henon map. The mean orbit length of the Henon map with an imposed 20 by 20 square partition is 76; therefore, a sufficient buffer size is 304 iterations. Fig. 66 displays the contour plot of the first buffer of the Henon map with a single point modified by 10% to form an anomaly. Fig. 67 displays the contour plot of the absolute differences between the ETM_{Norm} and the ETM of the first buffer for the Henon Map.

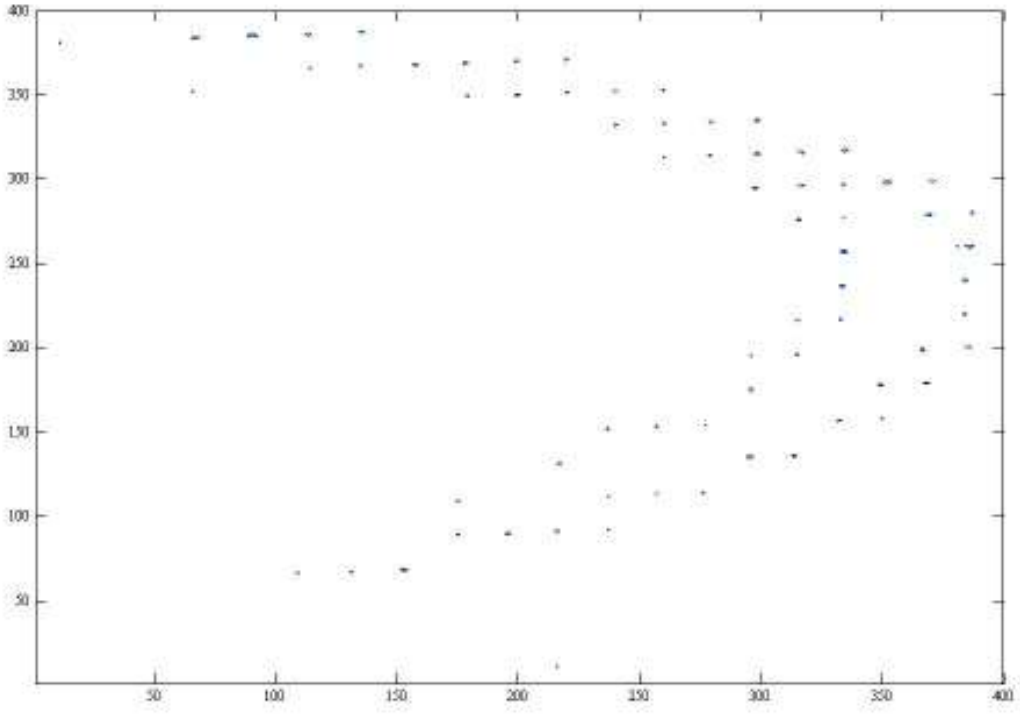

Figure 66. Contour plot of the ETM of the first buffer of the Henon Map with a single point modified manually by 10% to be an anomaly.

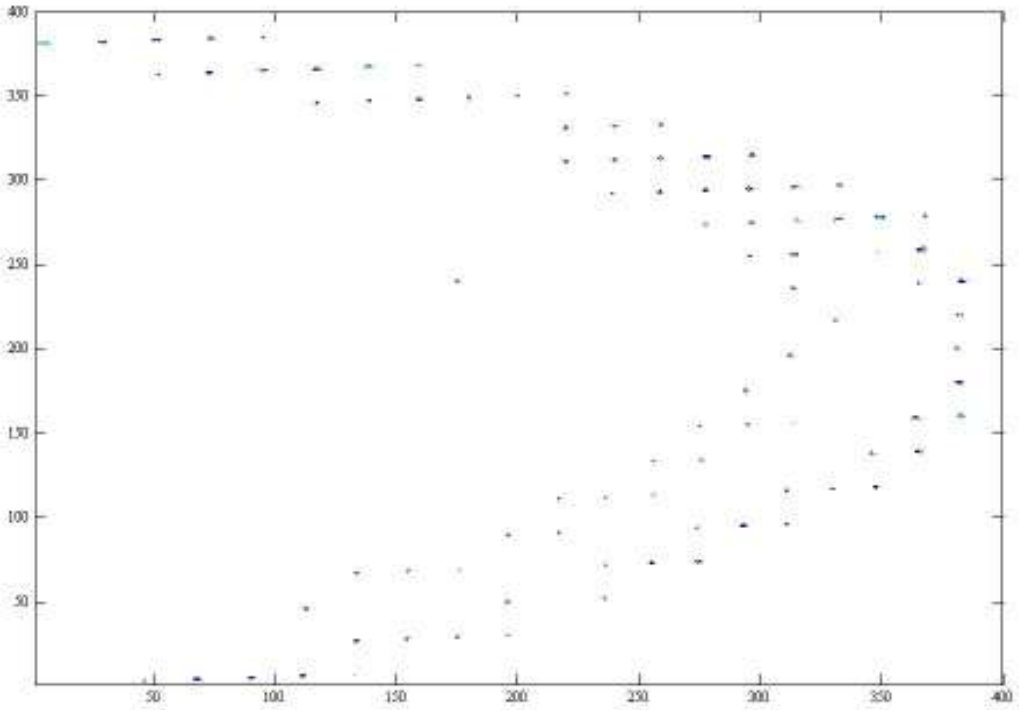

Figure 67. Contour plot of the absolute differences of the ETM of Normality and the ETM of the first buffer (with a single point modified) for the Henon Map.

As can be seen in Fig. 67, the proposed methodology correctly displays a shift in the system for a single anomalous point. This detection is highly effective at detecting even small anomalies. The data point which was changed to develop the ETMs for Fig. 16-17 was originally -1.0667 and was changed by 10%; however the proposed methodology detected this anomaly as a direct outlier in the system showing a maximum $\Delta s = 0.0106$. Given that the mean difference between the ETM_{Norm} and the first buffer of the Henon map is ~0.00205, the maximum difference would be detected even though a threshold may have been set to filter out the small changes of the system.

This sub-section has illustrated the use of the proposed methodology on two well known; the difference equations of the Henon map and the Duffing map. The use of the Duffing map illustrated the ability of the system to detect even small shifts in the chaotic

system which could indicate state changes forthcoming. The Henon map was used to illustrate the ability of the proposed methodology to detect direct anomalies. Further, this sub-section has illustrated the effectiveness of the novel Ergodic Transition Matrix to detect system state changes and anomalies. It was show, using two deterministically chaotic systems that the ETM detection algorithm is highly accurate and is capable of detecting extremely small system state changes as well as direct anomalies in mathematical difference equations.

6.3.2 ETMs for Cyber Security

This sub-section presents the results of a case study in which the proposed chaotic state change and anomaly detection algorithms were applied to a cyber security dataset. The dataset used for this study was originally produced by the Massachusetts Institute of Technology's (MIT) Lincoln Laboratory for the Defense Advanced Research Projects Agency (DARPA) [66]. The dataset consists of network data captured from a simulated United States military network. The simulated network was representational of a true military cyber network while maintaining obvious identification constraints pertinent to the national security level of the network. The captured data was organized into 5, week long, increments. The first, week long, increment of data represented a clean network with no intrusions or attacks. The second week of data represented the network in normal operation with 43 intrusions/attacks tagged with their attack name and time. The remaining 3, week long, increments represented normal network operations without tagging the intrusions/attacks.

For the purposes of this case study, only week 1 and week 2 data was used. This constraint was placed to illustrate clearly the ability of the proposed ETMs to capture anomalies present in the data. The ETM of Normality, ETM_{Norm} was generated using the data from week 1. The continuous buffer ETM, ETM_{Buff} was created using through streaming the data from week 2. The ETM_{Buff} was generated for in segments of 4 mean orbits of the data, where each orbit consisted of 228 records captured from the network.

The DARPA Intrusion Detection dataset used here was the 1999 version of the available datasets (3 dataset were available: 1998, 1999 and 2000). This version consisted of 6 files for each day of the week being considered – a Transmission Control Protocal (tcp) dump of data being received from outside the network, a tcp dump of data generated inside of the network, Solaris BSM audit data, NT audit data, dumps of selected directories and a file system listing record. Of these 6 files, only two were considered for this case study: the tcp dumps from both internal and external sources. These two files contained the same variables – a record identification number, the time of the record, the source address of the record, the destination address of the record, the protocol which was used, and a comment field. There exist a total of 14,406,511 records for week 1 and 13,178,081 records for week 2 in the files being considered.

The proposed ETM state change and anomaly detection methodology functions best when the system dynamics are represented by a single variable. In order to utilize the ETM methodology on the DARPA dataset a transformation was required to render the data useful. This transformation consisted of discretizing the source, destination and protocol fields. The identification, time and comment fields were removed from the dataset while the time-based ordering of the data was preserved, as is required for chaotic

systems. The number of distinct values for the three variables considered is given in Table 13, where only week 1 and week 2 were considered.

Table 13. Number of distinct values for the variables considered in the DARPA case study.

Week	Variable	Number of Distinct Values
1	Source	1631
2	Source	1610
1	Destination	1638
2	Destination	1616
1	Protocol	47
2	Protocol	47

To facilitate the most complete representation of the system, each record was appended with an integer value which represented the string concatenation of the three variables considered. Thus a record which contained a discretized source of 1245, a discretized destination of 1047 and a discretized protocol of 25 was appended with the integer 1245104725. It is this integer value that is used to create the embedding phase space for the intrusion detection dataset, which is also used to form the ETMs which will be considered here.

Initially, it was hypothesized that a single ETM of Normality would be generated based on the training data from week 1. Following this methodology, the validation data of week 2 would be streamed into the ETM system in buffers whose sizes are four mean orbits. The ETM of the buffer is compared to the ETM of Normality for detection of system shifts or anomalies. The ETM partition size that showed optimal results was a 15 x 15 square partition (this was based on an embedding phase space reconstruction with a time delay of 8 and an embedding dimension of 3).

Disappointingly, it was discovered that this process was not effective at detecting intrusions/attacks. It was determined that this was due to the overwhelming size, and variance, of the training data. Table 14 illustrates this fact through the use of a confusion matrix of the single ETM of Normality ran on the cyber dataset. The confusion matrix indicates the number of actual intrusion/attacks that were detected in the system (upper left cell), the number of records which were falsely determined to be intrusions/attacks (upper right cell), the number of intrusions/attacks which were not detected (lower left cell) and the number of records which were not intrusions/attacks and were not detected as an intrusion/attack (lower right cell).

Table 14. Confusion matrix for the proposed ETM detection system on the DARPA dataset with a single ETM of Normality.

	True Anomaly	**False Anomaly**
True Anomaly	24	345
False Anomaly	19	13177693

As can be seen in Table 14, the single ETM of Normality misclassified 345 records sets as intrusions/attacks which were not of such a class. Also, the system also ignored 19 intrusions/attacks which were present in the system. The false positive rate of the single ETM of Normality for the system is unacceptable for use as a true cyber security detection system. Further, the number of intrusions/attacks which were missed by the system is also too high to be acceptable.

To facilitate a more robust detection system, it was determined that an ensemble of ETMs of Normality should be created from the week 1 training data. To perform this task, it was determined to split the training data into a number of segments, each of which consisted of a relatively equal number of mean orbits. After much trial and error it was

found that 18 segments worked most effectively for the cyber security dataset. Therefore, an ensemble of 18 ETMs of Normality was generated. Fig 68 illustrates two of these ETMs of Normality in the form of a surface plot of the ETM.

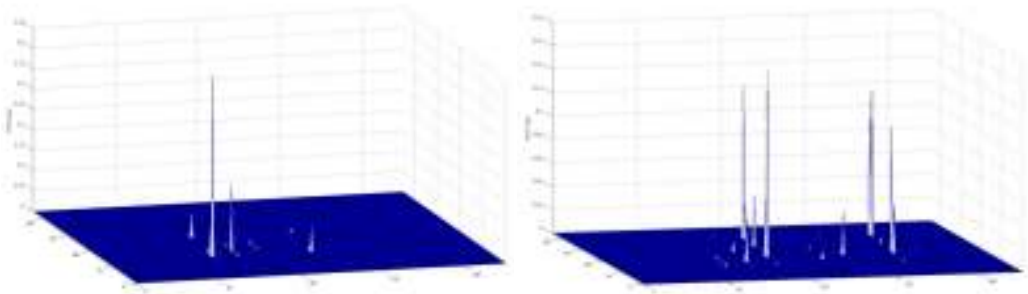

Figure 68. Two examples from the ensemble of ETMs of Normality for the DARPA dataset.

The use of the ensemble of ETMs of Normality inherently increased the computation cost of comparison to the buffer ETM. Each buffer ETM was required to be compared to the full ensemble of ETMs of Normality. This required a determination of which ETM in the ensemble of ETMs of Normality best matched the buffer ETM, and calculation of the differences between that ETM of Normality and the buffer ETM. If the difference between the two ETMs was above the given threshold then the buffered record set was flagged as containing an intrusion/attack. While there was an increase in computation, it was discovered that the system still operated in near real-time conditions due to the sparse matrix format of the ETMs. Table 15 illustrates, in the form of a confusion matrix, the gain in accuracy that was achieved through the use of the ensemble of ETMs of Normality.

Table 15. Confusion matrix for the DARPA dataset using an ensemble of ETMs of Normality.

	True Anomaly	**False Anomaly**
True Anomaly	42	8
False Anomaly	1	13178030

As can be seen in Table 15, the use of an ensemble of ETMs of Normality greatly increased the accuracy of the detection system. With a single missed intrusion/attack and only 8 record sets misclassified as intrusions/attacks, the detection system is highly useful as a cyber security intrusion/attack detection system. Thus, this it has been successfully shown that the ETM detection system proposed in this thesis is effective at detection of anomalies and can be made to perform very well even with extreme dataset sizes.

6.3.3 ETMs for the Detection of Mechanical System Change

This sub-section highlights a case study of system change detection, using the proposed ETM change detection system, in a mechanical system. The system under consideration for this study is a wind turbine in existing use on a wind farm in Iowa. Data from the wind turbine Supervisory Control and Data Acquisition (SCADA) system was collected by the Intelligent Systems Laboratory of the University of Iowa [35]. The captured data was continuous for only short times due to the collection system deployed at the time of the data capture. Thus, the case study of this sub-section uses this SCADA data to build a model of continuous wind turbine data which is then slowly modified to indicate a shift in the mechanical system.

The SCADA data used in this study consisted of 105 different variables, along with time, date, and the turbine number, for each record. The data used here was collected every 10 minutes from a single wind turbine. For the purposes of this study only

two variables were used from the 105 available variables – wind speed and torque. Other variables were initially considered, such as bearing temperatures and drive train acceleration; however it was discovered that each of these variables varied proportionately with the wind speed. Thus, it was decided that wind speed and torque would suffice for the study.

The SCADA data was used to form a model of torque versus wind speed. To facilitate an accurate model, many different samples of data were extracted from the SCADA data (Fig. 69 illustrates an instance, of the torque measurement over time, from the samples extracted). This extracted data was then used to generate an equation in which torque is a function of wind speed. The equation generation was performed using a genetic program which evolved a population, of randomly defined equations. The genetic program's fitness function was the calculation of the MAE of the evolved equation from the actual data.

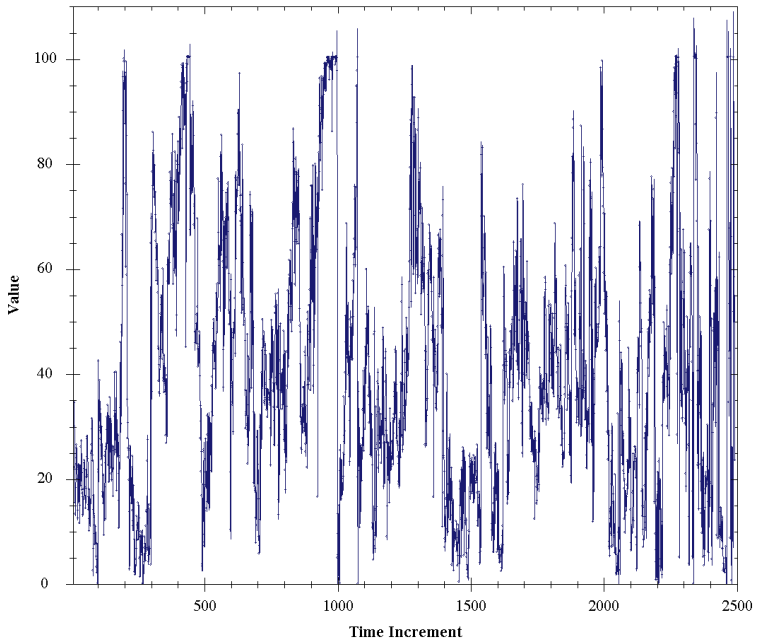

Figure 69. Torque (vertical axis) given in relation to time for a single wind turbine for 17.36 days.

The software used to generate model equation for torque, using the genetic program algorithm, is a freely available solution created at Cornell University entitled Eureqa [67]. This software generate the equation used for this case study as a function of wind, as given in Eq. (30) where ω is the wind speed at a given time. This model produced a polynomial whose MAE to the actual data was 0.0178.

$$torque = 2.1467 - 2.3249\omega + 96.6\omega^2 - 19.59\omega^3 + 2.11\omega^4 - 0.114\omega^5 + 0.0024\omega^5 \quad (30)$$

To generate the data for the ETM system shift and anomaly detection algorithm to use, wind speed must be given for each time increment. Rather than develop a model for wind speed, which could introduce further differences between the model and the real data, it was decided to use data collected from an anemometer. The author purchased and installed a commercial anemometer at a height of 30 feet above the ground in an open area with no buildings or significant contours within 500 feet of the site. Data was captured for a period of three months from this instrument and was used to generate the model data for this case study.

An ETM of Normality was generated using two months worth of the model data. For this time frame no anomalies or shifts were allowed to enter the torque model, thus forming the normal operating conditions of the modeled turbine. The embedding phase space for this system was generated with a time delay of 2 and an embedding dimension of 3. The ETM's for this system were created based upon a 10 x 10 square partitioning of the embedding phase space. The ETM of Normality for this time period is presented in the surface plot of Fig. 70.

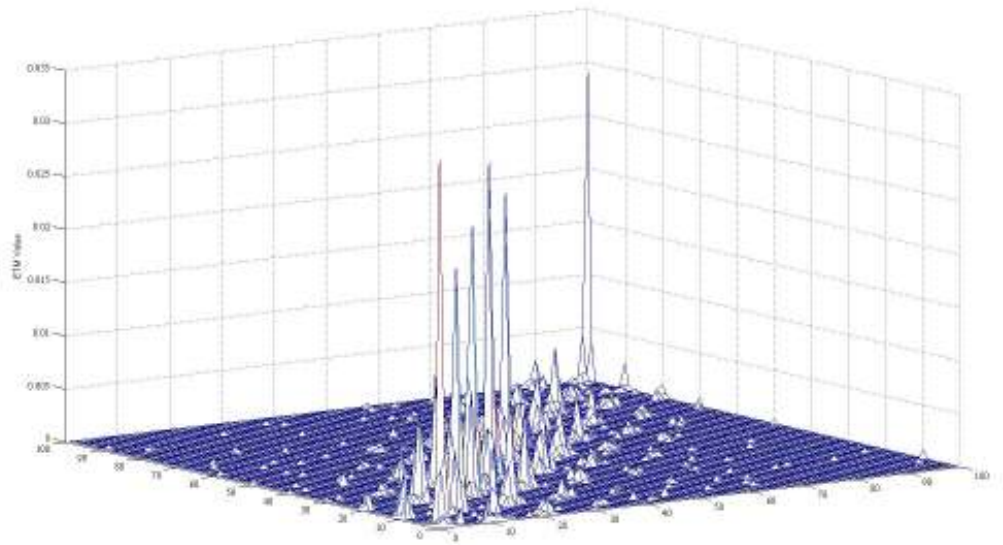

Figure 70. ETM of Normality for the wind turbine torque model.

To simulate the wind turbine experiencing small system changes in the torque parameter, the model equation was modified to include a very small increase in the torque value. This was accomplished through the addition of a value proportionate to the torque value as given in Eq. (31). This modified data was then streamed into the ETM detection system as a buffer, in a manner similar to previous case studies in this thesis. Fig. 71 illustrates an example ETM for this buffer.

$$torque = torque + (tourque * 0.001) \tag{31}$$

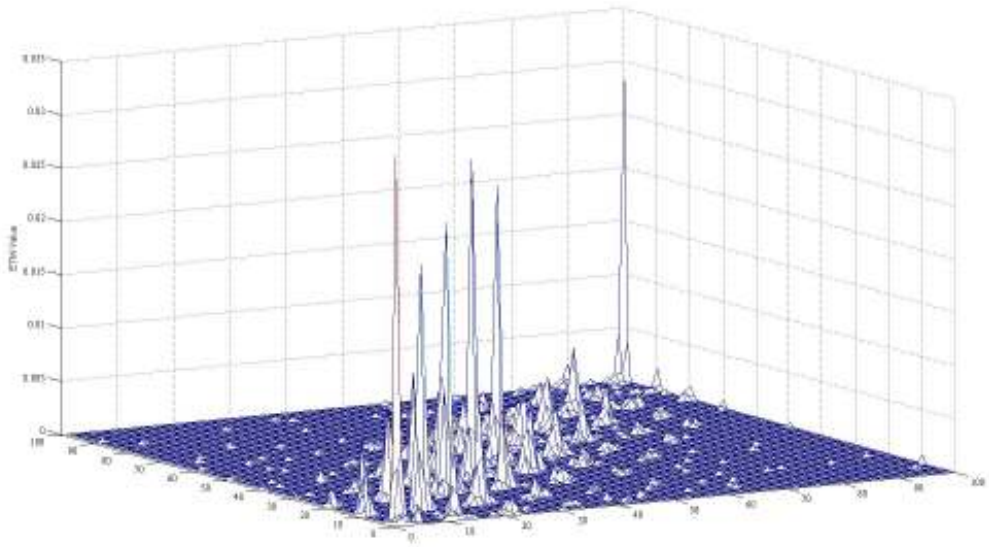

Figure 71. Buffer ETM for the wind turbine torque model.

It is very difficult to detect the differences present between the ETM of Normality given in Fig. 70 and the Buffer ETM given in Fig. 71. To facilitate ease of observation, a surface plot of the absolute differences between these ETMs is given in Fig. 72. As can be seen from Fig. 72, the ETM system change and anomaly detection methodology was successful at capturing the extremely small system shift introduced in this model. In fact, it was discovered that the system detected the small perturbations in the first buffer ETM that contained the perturbations.

Fig. 72 definitively illustrates that the ETM system shift and anomaly detection methodology effectively captures minute system changes. The scale of the absolute difference (vertical axis) in Fig. 72 is in the level of 10^{-4}, hence even very small system shifts are able to be captured using the ETM methodology.

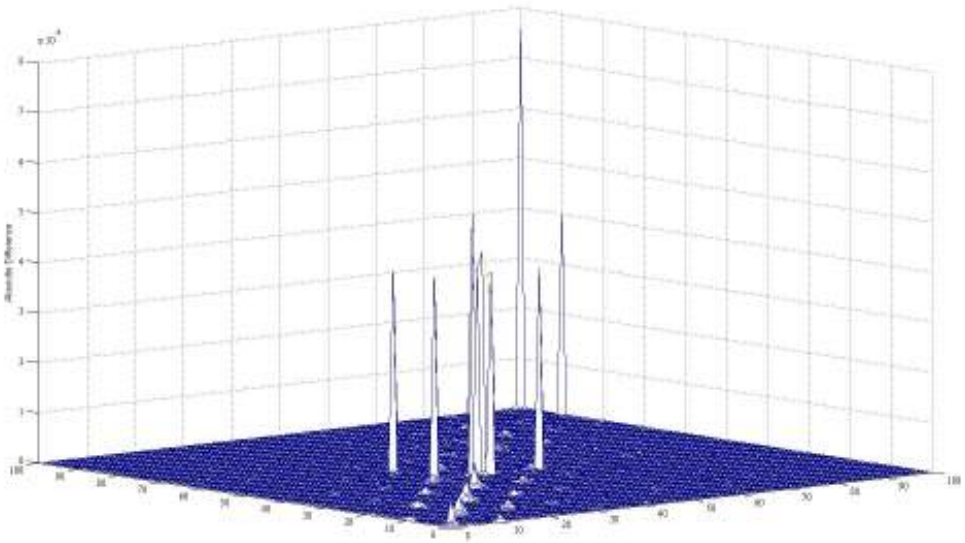

Figure 72. Absolute difference between the ETM of Normality (Fig. 70) and the buffer ETM (Fig. 71) for the wind turbine torque model.

This section has effectively illustrated that the ETM system shift and anomaly detection methodology is adept at detecting direct anomalies (e.g. the Duffing Map) and system shifts (e.g. the Henon Map) in mathematical difference equations. Further, this section has show that this methodology is effective at detecting direct anomalies (e.g. DARPA Intrusion Detection dataset) and system shifts (e.g. the wind turbine torque model) in real world systems as well. The results of the case studies in this section indicate that there is just cause to pursue further research into the application of the detection methodology presented in Section 5.2

CHAPTER 7. CONCLUSIONS

This thesis has presented research related to the detection of normality and non-normality in deterministically chaotic systems. The research presented illustrated a number of novel techniques for accomplishing this detection. This concluding section highlights these techniques and the results of the use of these techniques in the various case studies which were presented.

In Section 3, a new methodology for visualizing the states of a chaotic system was introduced. The traversal of system states was represented by this visualization known herein as the Ergodicity Plot (EP). The proposed EP represented system dynamics in a much clearer fashion than has been given in standard Recursion Plots and can be used to detect system normality. Further, the use of the EP allowed for detection of frequent system state change motifs which are then used to predict the next system states.

A new prediction technique for chaotic systems was presented in Section 4 of this thesis. This technique took into account the sensitive dependence on initial conditions, which is a hallmark of chaotic systems, to produce more accurate results in prediction. The technique was validated on three dataset, the standard Henon and Duffing maps as well as a dataset captured from a working industrial wind turbine. The results of the prediction technique on these datasets indicated that it is a viable technique for determining future values of a chaotic system,

Section 5 presented a novel anomaly and state change detection technique. This technique utilized the inherent ergodicity of chaotic systems to determine normality of the system and to detect small system shifts as well as direct anomalies. It was shown that the presented technique offers a unique ability to detect such changes and does so in a

computationally small manner which allows for real time deployment of the technique. The technique was validated in Section 6 using the standard Henon map, a mechanical system and a cyber security dataset. The results of these validation case studies clearly indicated the viability of the technique in real world scenarios.

7.1 Future Research

The research described above has illustrated that there is merit in pursuing further work related to the domain of normality and anomaly/state change detection in chaotic systems. The encouraging results of the presented novel algorithms and methodologies, as seen in the successful case studies, should be extended to facilitate better understanding of deterministically chaotic systems and to assist in real time monitoring of such systems. This section describes future work extending the research of the previous sections.

The novel Ergodic Transition Matrix of Chapter 5 accurately detects outliers, anomalies and system state changes in deterministically chaotic systems. Future work concerning the ETMs includes applying the ETM detection methodology to embedded devices to facilitate real-time anomaly and system shift detection in mission critical electronic devices. Devices such as critical communications systems and air navigation/collision avoidance systems may benefit from the application of this detection methodology.

Finally, future research in this domain includes the use of Ergodic Transition Matrices on human physiological data. Some initial work performed in this area has indicated the ability to accurately monitor and detect anomalies in the human heart.

Future research should explore the possibilities of the detection of such ailments as epileptic seizures prior to their attack, detection of heart arrhythmias prior to a heart attack or stroke, and possible detection of heightened stress in soldiers as they enter combat arenas.

APPENDIX

C# Method for the Calculation of Lyapunov Exponent

This section presents a sample C# method for the calculation of the Lyapunov exponent discussed in Chapter 2 of this thesis. The method assumes that the time series dataset has been stored in a C# data table object and that the embedding dimension of the reconstructed phase space has been stored as an integer in a variable called _embeddingDim. Finally, the method presented here assumes that the time delay parameter of the reconstructed phase space has been stored as an integer in a variable called _timeDelay. Fig A1-A2 presents the C# code for this method.

```
private double CalculateLyapunov()
{
      double sum = 0.0;
      int cnt = 0;
      int embedDim = _embeddingDim;
      int _tau = _timeDelay;

      for (int tt = 0; tt < _currentData.Rows.Count - (int)(embedDim * _tau);
            t++)
      {
            //create a vector for the initial point
           Dictionary<int, double> initPt = new Dictionary<int, double>();
           for (int m = 0; m < embedDim; m++)
           {
             initPt.Add(m,
                  Convert.ToDouble(_currentData.Rows[tt + (int)(m * _tau)][0]));
           }
```

Figure A1. C# Method for calculating the Lyapunov exponent of a time series.

```
//find the closest point to the initial point
Dictionary<int, double> closePoints = new Dictionary<int, double>();
for (int uu = 0; uu < _currentData.Rows.Count - (int)(embedDim * _tau); uu++)
{
   if (uu == tt)
      continue;
   Dictionary<int, double> pt = new Dictionary<int, double>();
   for (int n = 0; n < embedDim; n++)
   {
      pt.Add(n,Convert.ToDouble(_currentData.Rows[uu + (int)(n * _tau)][0]));
   }
   //calculate the Euclidean distance between these points
   double tempDist = 0.0;
   foreach (int d in initPt.Keys)
   {
       tempDist += Math.Pow(initPt[d] - pt[d], 2);
   }
   tempDist = Math.Sqrt(tempDist);
   closePoints.Add(uu, tempDist);
}
int closestPt = -1;
double distance = double.MaxValue;
foreach (int w in closePoints.Keys)
   if (closePoints[w] < distance)
   {
       distance = closePoints[w];
       closestPt = w;
   }
//Evolve the attractor by _tau and recalclate the distance of each point
Dictionary<int, double> initPt2 = new Dictionary<int, double>();
for (int m = 0; m < embedDim; m++)
{
   initPt2.Add(m, Convert.ToDouble(_currentData.Rows[tt + (int)_tau +
               (int)(m * _tau)][0]));
}

Dictionary<int, double> closePt2 = new Dictionary<int, double>();
for (int m = 0; m < embedDim; m++)
{
    closePt2.Add(m, Convert.ToDouble(_currentData.Rows[closestPt +
               (int)_tau + (int)(m * _tau)][0]));
}
double tDist = 0.0;
foreach (int d in initPt2.Keys)
{
    tDist += Math.Pow(initPt2[d] - closePt2[d], 2);
}
tDist = Math.Sqrt(tDist);
//calculate the sum
double val = Math.Abs(tDist / distance);
if (val != 0.0)
{
   if (val < double.MaxValue)
       sum += Math.Log(val, 2);
   if (sum < double.MinValue || sum > double.MaxValue)
        break;
}
cnt++;
tt += (int)_tau;
}
if (sum == 0.0)
  return sum;
sum = sum * (1.0 / (double)(cnt * _tau));
  return sum;
}
```

Figure A2. Method for calculating the Lyapunov exponent of a time series continued.

Pseudocode Method for Creating an Ergodicity Plot

This section presents pseudocode for a computer software method for creating an ergodicity plot discussed in Chapter 3 of this thesis. The code supplied here assumes that the time series has been transformed into embedding phase space using the time delay and embedding dimension parameters discussed in Chapter 2 of this thesis. The code also assumes a user supplied parameter, gridSize, of the size of the partition to be imposed on the phase space. Figure A3 presents this pseudocode.

```
CreateErgodicityPlot(int gridSize)
{
   //determine the maximum and minimum values of the phase space
   //methods are not supplied as they are intuitive
   double max = MaximumPhaseSpaceValue();
   double min = MinimimPhaseSpaceValue();
   double binSize = (max-min)/gridSize;
   List bins = new List();
   for tt=0; tt<gridSize; tt++
        for ii=0; ii<gridSize; ii++;
        {
          Bin.LowY = min + (tt * binSize);
          Bin.HighY = min + (tt * binSize) + binSize;
          Bin.LowX = min + (ii*binSize);
          Bin.HighX = min + (ii*binSize) + binSize;
          bins.Add(Bin);
        }
   //assign each point to the centroid of the bin that contains that point
   for tt=0; tt<embeddingPhaseSpace.Size; tt++
   {
        double x = embeddingPhaseSpace[tt].X;
        double y = embeddingPhaseSpace[tt].Y;
        foreach Bin in bins
        {
          if(x>=Bin.LowX and x<=Bin.HighX)
             if(y>=Bin.LowY and y<=Bin.HighY)
                Bin.Points.Add(x,y);
        }
   }
   //plot the centroids following the time series order, place the bin in an orbit
   List orbits = new List(); //list of bins in order of traversal
   foreach point in embeddingPhaseSpace
        foreach Bin in bins
           if(Bin.Points.Contains(point))
            {
                Plot(Bin.Center);
                Orbits.Add(Bin);
             }
```

Figure A3. Pseudocode method for creating an ergodicity plot.

Pseudocode Method for Creation of an ETM

This section presents a pseudocode method for the creation of an Ergodicity Transition Matrix discussed in Chapter 5 of this thesis. The presented method assumes that an Ergodicity Plot, whose pseudocode presentation was given in previous appendix, has previously been created and uses the Orbits parameter, and the Bin structures, of that method. The code also assumes a user supplied parameter, gridSize, of the size of the partition to be imposed on the phase space. Figure A4 presents this pseudocode.

```
CreateErgodicityTransitionMatix(gridSize, Orbits)
{
   size = gridSize * gridSize;
   double[,] transitions = new double[size,size];

   //initialize the array
   for ii=0; ii<size; ii++
   {
        for tt=0; tt<size; tt++;
           transitions[ii,tt] = 0;
   }

   //using the bins of the Ergodicity Plot, create the ETM
   for (int tt = 1; tt < orbits.Count; tt++)
   {
       Bin toCell = orbits[tt];
       Bin fromCell = orbits[tt - 1];
       row = (gridSize * fromCell.Center) + fromCell.Center;
       col = (gridSize * toCell.Center) + toCell.Center;
       transitions[row, col] += 1;
   }
   for (int r = 0; r < gridsize * gridsize; r++)
   {
        for (int c = 0; c < gridsize * gridsize; c++)
             transitions[r, c] = (transitions[r, c]/orbits.Count);
   }
```

Figure A4. Pseudocode method for the creation of an ETM.

Pseudocode Method for Using ETMs of Normality and Streaming Data to Detect Anomalies

This section presents a pseudocode method for a computer software program for the detection of anomalies or system state changes in a chaotic system using the methodology described in Chapter 5 of this thesis. The presented method assumes that the time series data for the training set has been transformed into embedding phase space as described in Chapter 2 of this thesis. The method also assumes that the user is supplying the streaming data as a parameter in segments of 3-4 mean orbit lengths. Finally, the method requires a user supplied parameter which determines the threshold of differences between the training data ETM and the streamed data ETM. Methods of previous appendices are also utilized. Fig. A5 illustrates the pseudocode method.

```
DetectShiftsAndAnomalies(trainingData, streamedData, gridSize, deviation)
{

   //create the etm of the training data
   Orbits = CreateErgodicityPlot(trainingData, gridSize);
   normETM = CreateErgodicityTransitionMatrix(trainingData, gridSize, Orbits);

   //create the etm of the streaming data
   SOrbits = CreateErgodicityPlot(streamedData, gridSize);
   buffETM = CreateErgodicityTransitionMatrix(streamedData, gridSize, SOrbits);

   //compare the matrices
   for ii=0; ii<gridSize*gridSize; ii++
        for tt=0; tt<gridSize*gridSize; tt++
           if(AbosluteValue(normETM[ii,tt]-buffETM[ii,tt])>deviation)
                return true;

   //no anomalies or shifts are detected if execution arrives at this point
   return false;
}
```

Figure A5. Pseudocode method for the detection of anomalies or system shifts in streamed data into the ETM methodology.

REFERENCES

[1] E. Lorenz, *The Essence of Chaos*. Seatle, WA: The University of Washington Press, 1993.

[2] Z. Zhang, K. Lam, W. Yan, H. Gao, and Li.Y, "Time Series Prediction using Lyapunov Exponents in Embedding Phase Space," *Computers and Electrical Engineering*, vol. 30, no. 1, pp. 1-15, 2004.

[3] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, 2nd ed. Cambridge, UK: Cambridge University Press, 2004.

[4] A. Isidori, *Nonlinear Control Systems*, 3rd ed. Berlin: Springer, 1995.

[5] P. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Boston, MA: Pearson Education, 2006.

[6] T Lefebvre, H. Bruyninckx, and J. De Schutter, "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Transactions on Automatic Control*, vol. 47, no. 8, pp. 1406-1409, August 2002.

[7] C. Gellings, *The Smart Grid: Enabling Energy Efficency and Demand Response*. Lilburn, GA: The Fairmont Press, 2009.

[8] S. Chaudhury, A. Smith, B. Anderson, S. Ghose, and P. Jessen, "Quantum Signitures of Chaos in a Kicked Top," *Nature*, vol. 461, pp. 768-771, October 2009.

[9] E. Lorenz, "Deterministic Nonperiodic Flow," *Journal of Atmospheric Science*, vol. 20, pp. 130-141, 1963.

[10] T. Li and J. Yorke, "Period Three Implies Chaos," *The American Mathematical Monthly*, vol. 82, no. 10, pp. 985-992, 1975.

[11] A. Fraser, "Information and Entropy in Strange Attractors," *IEEE Transactions on Information Theory*, vol. 35, no. 2, pp. 245-262, March 1989.

[12] S. Strogatz, *Nonlinear Dynamics and Chaos*. New York: Westview Press, 1994.

[13] P. Grassberger and I. Procaccia, "Measuring the Strangeness of Strange Attractors," *Physica* , vol. 9D, pp. 189-208, 1983.

[14] E. Glasner and B. Weiss, "Sensitive Dependence on Initial Conditions," *Nonlinearity*, vol. 6, no. 6, pp. 1067-1075, 1993.

[15] H Furstenberg, "Poincare Recurrence and Number Theory," *Bulletin of the American Mathematical society*, vol. 5, no. 3, pp. 211-234, 1981.

[16] W. Briggs and V. Henson, *The DFT: An Owner's Manual for the Discrete Fourier Transform*. Philadelphia PA: SIAM, 1987.

[17] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia: SIAM, 1992.

[18] J. Gao, Y. Cao, W. Tung, and J. Hu, *Multiscale Analysis of complex Time Series*. Hoboken, NJ: Wiley & Sons, 2007.

[19] J. Sprott, *Chaos and Time-Series Analysis*. Oxford, UK: Oxford University Press, 2003.

[20] A. Wolf, J. Swift, H. Swinney, and J. Vastano, "Determining Lyapunov Exponents from a Time Series," *Physica D: Nonlinear Phenomena*, vol. 16, no. 3, pp. 285-317, 1985.

[21] M. Rosenstein, J. Collins, and C. De Luca, "A Practical method for Calculating Largest Lyapunov Exponents from Small Data Sets," *Physica D: Nonlinear Phenomena*, vol. 65, no. 1-2, pp. 117-134, 1993.

[22] J. Holzfuss and U. Parlitz, "Lyapunov Exponents from Time Series," in *Lecture Notes in Mathematics Vol. 1486*. Berlin: Springer, 1991, ch. 4, pp. 263-270.

[23] J. Sprott, "A Simple Chaotic Delay Differential Equation," *Physics Letters A*, vol. 366, no. 4-5, pp. 397-402, 2007.

[24] T. Schreiber and A. Schmitz, "Improved Surrogate Data for Nonlinearity Tests," *Physical Review Letters*, vol. 77, no. 4, pp. 635-638, 1996.

[25] J. Theiler, S. Eubank, A. Longtin, B. Galdrikian, and J. Farmer, "Testing for Nonlinearity in Time Series: The Method of Surrogate data," *Physica D*, vol. 58, no. 1-4, pp. 77-94, 1992.

[26] R. Steuer, J Kurths, C. Daub, J. Weise, and J. Selbig, "The Mutual Information: Detecting and Evaluating Dependencies Between Variables," *Bioinformatics*, vol. 18, no. 90002, pp. S231-S240, 2002.

[27] F. Takens, "Detecting Strange Attractors in Turbulence," *Lecture Notes in Mathematics*, vol. 898, pp. 366-381, 1981.

[28] M. Kennel, R. Brown, and H. Abaranel, "Determining embedding Dimension for Phase-Space Reconstruction using a Geometric Construction," *Physical Review A*, vol. 45, no. 6, pp. 3403-3411, 1992.

[29] M. Hong-guang and H. Chong-zhao, "Selection of Embedding Dimenstion and Delay Time in Phase Space Reconstruction," *Journal of Xi'an Jiaotong University*, vol. 38, no. 4, pp. 335-338, 2004.

[30] J. Eckmann and D. Ruelle, "Ergodic Theory of Chaos and Strange Attractors," *Review of Modern Physics*, vol. 57, no. 3, pp. 617-656, 1985.

[31] N. Packard, J. Crutchfield, J. Farmer, and R. Shaw, "Geometry from a Time Series," *Physical Review Letters*, vol. 45, no. 9, pp. 712-716, 1980.

[32] J. Milnor, "On the Concept of Attractor," *Communications in Mathematical Physics*, vol. 99, no. 2, pp. 177-195, 1985.

[33] P. Cvitanovic, G. Gunaratne, and I. Procaccia, "Topological and Metric Properties of Henon-type Strange Attractors," *Physical Review A*, vol. 38, no. 3, pp. 1503-1520, 1988.

[34] D. Cross and R. Gilmore, "Representation Theory for Strange Attractors," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 80, no. 5, pp. 056207-1 - 056207-6, 2009.

[35] A. Kusiak, Z. Zhang, and M. Li, "Optimization of Wind Turbine Performance with Data-Driven Models," *IEEE Trans. on Sustainable Energy*, vol. 1, no. 2, pp. 66-76, 2010.

[36] N. Packard, J. Crutchfield, J. Farmer, and R. Shaw, "Geometry from a Time Series," *Physical Review Leters*, vol. 45, no. 9, pp. 712-716, 1980.

[37] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys*, vol. 41, no. 3, 2009.

[38] I. Aydin, M. Karakose, and E. Akin, "Chaotic-based Hybrid Negative Selection Algorithm and its Application in Fault and Anomaly Detection," *Expert Systems with Applications*, vol. 37, no. 7, 5285-5294 2010.

[39] T. Geisel and S. Thomae, "Anomalues Diffusion in Intermittent Chaotic Systems," *Physical Review Letters*, vol. 52, no. 22, pp. 1936-1939, 1984.

[40] W. Xiong, H. Hu, Y. Yang, and Q. Wang, "Anomaly Detection of Network Traffic based on the Largest Lyapunov Exponent," in *2nd Intl Conf on Advanced Computer Control*, Shenyang, 2010, pp. 521-585.

[41] L. Sheu, H. Chen, J. Chen, and L. Tam, "Chaotic Dynamics of the Fractionally Damped Duffing Equation," *Chaos, Solitons & Fractals*, vol. 32, no. 4, pp. 1459-1468, 2007.

[42] J. Chen and W. Chen, "Chaotic Dynamics of the Fractionally Damped van der Pol Equation," *Chaos, Solitons & Fractals*, vol. 35, no. 1, pp. 188-198, 2008.

[43] P. Zhao, L. Xing, and J. Yu, "Chaotic Time Series Prediction: From One to Another," *Physics Letters A*, vol. 373, no. 25, pp. 2174-2177, 2009.

[44] J. Eckmann, S. Kamphorst, and D. Ruelle, "Recurrence Plots of Dynamical Systems," *Europhys. Lett.*, vol. 4, no. 9, pp. 973-977, 1987.

[45] B. Eckhardt and D. Yao, "Local Lyapunov Exponents in Chaotic Ssytems," *Physica D*, vol. 65, no. 1-2, pp. 100-108, 1993.

[46] D. Lind and B. Marcus, *An Introduction to Symbolic Dynamics and Coding*. Cambridge: Cambridge University Press, 1996.

[47] S. Gupta and A. Ray, "Pattern Identification using Lattice Spin Systems: A Thermodynamic Formalism," *Applied Physics Letters*, vol. 91, pp. 194105-1-194105-3, 2007.

[48] J Crutchfield and D Feldman, "Regularitires Unseen, Randomness Observed: Level of Entropy Convergence," *Chaos*, vol. 13, no. 1, pp. 25-60, 2003.

[49] J. Farmer and J. Sidorowich, "Predicting Chaotic Time Series," *Physical Review Letters*, vol. 39, no. 8, pp. 845-848, 1987.

[50] M. Casdagli, "Nonlinear Prediction of Chaotic Time Series," *Physica D: Nonlinear Phenomena*, vol. 35, no. 3, pp. 335-356, 1989.

[51] D. Karunasingha and S. Liong, "Enhancement of Chaotic Time Series Prediction with Real-time Noise Reduction," in *Proc. of the Intl. Conf. on Small Hydropowere*, Sri Lanka, October 2007.

[52] C. Damle and A. Yalcin, "Flood Prediction Using Time Series Data Mining," *Jrnl. of Hydrology*, vol. 333, no. 2-4, pp. 305-316, 2007.

[53] A Eiben and J Smith, *Introduction to Evolutionary Computation*. Berlin: Springer, 2003.

[54] A. Patcha and J. Park, "An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends," *Computer Networks*, vol. 51, no. 12, pp. 3448-3470, 2007.

[55] K. Wang and S. Stolfo, "Anomalous Payload-based Network Intrusion Detection," *Lecture Nots in Computer Science: Recent Advances in Intrusion Detection*, vol. 3224/2004, pp. 203-222, 2004.

[56] L. Fang and N. Peng, "LAD: Localization Anomaly Detection for Wireless Sensor Networks," *Journal of Parallel and Distributed Computing*, vol. 66, no. 7, pp. 874-886, 2006.

[57] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1-58, 2009.

[58] M. Chuah and F. Fu, "ECG Anomaly Detection via Time Series Analysis," *Lecture Notes in Computer Science: Frontiers of High Performance Computing and Networking*, vol. 4743, pp. 123-135, 2007.

[59] A. Azad, S. Alouf, E. Altman, V. Borkar, and G. Paschos, "Optimal Sampling for State Change Detection with Application to the Control of Sleep Mode," in *Proc. of the 48th IEEE Conf on Decision and Control*, Shanghai, December 2009, pp. 1645-1650.

[60] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image Change Detection Algorithms: A Systematic Survey," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 294-307, 2005.

[61] M. Tykierko, "Using Invariants to Change Detection in Dynamical Systems with Chaos," *Physica D*, vol. 237, no. 1, pp. 6-13, 2008.

[62] S. Chakraborty, S. Sarkar, and A. Ray, "Symbolic Identification and Anomaly Detection in Complex Dynamical Systems," in *2008 American Control Conference*, Seattle, WA, 2008, pp. 2792-2797.

[63] C. Rao, A. Ray, S. Sarkar, and M. Yasar, "Review and Comparative Evaluation of Symbolic Dynamic Filtering for Detection of Anomaly Patterns," *Signal, Image and Video Processing*, vol. 3, no. 2, pp. 101-114, 2009.

[64] K. Chung, *Markov Chains with Stationary Transition Probabilites*, 2nd ed. New York: Springer, 1967.

[65] A. Smith and N. Toppel, "Case Study: Using Security Awareness to Combat the Advanced Persistent Threat," in *Proc. of the 13th Colloquium for Information Systems Security Education*, Seattle, WA, 2009, pp. 64-70.

[66] Lincoln Laboratory MIT's. DARPA Intrusion Detection Dataset. A public web site http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1999data.html.

[67] M. Schmidt and H. Lipson, "Distilling Free-Form Natural Laws from Experimental Data," *Science*, vol. 324, no. 5923, pp. 81-85, 2009.