TIME SERIES DATA MINING: IDENTIFYING TEMPORAL PATTERNS FOR CHARACTERIZATION AND PREDICTION OF TIME SERIES EVENTS

by

Richard J. Povinelli, B.A., B.S., M.S.

A Dissertation submitted to the Faculty of the Graduate School, Marquette University, in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

Milwaukee, Wisconsin

December, 1999

This work is dedicated to my wife, Christine,

our son, Christopher,

and his brother, who will arrive shortly.

Acknowledgment

I would like to thank Dr. Xin Feng for the encouragement, support, and direction he has provided during the past three years. His insightful suggestions, enthusiastic endorsement, and shrewd proverbs have made the completion of this research possible. They provide an example to emulate. I owe a debt of gratitude to my committee members, Drs. Naveen Bansal, Ronald Brown, George Corliss, and James Heinen, who each have helped me to expand the breadth of my research by providing me insights into their areas of expertise.

I am grateful to Marquette University for its financial support of this research, and the faculty of the Electrical and Computer Engineering Department for providing a rigorous and stimulating environment that exemplifies *cura personalis*.

I thank Mark Palmer for many interesting, insightful, and thought provoking conversations on my favorite topic, Time Series Data Mining, and on his, Fuzzy Optimal Control. I am indebted to him for the many hours he spent reviewing this manuscript.

I am deeply grateful to my wife, Christine, for her generous editing expertise, ongoing moral support, and acceptance of my long hours away from our family.

Abstract

A new framework for analyzing time series data called Time Series Data Mining (TSDM) is introduced. This framework adapts and innovates data mining concepts to analyzing time series data. In particular, it creates a set of methods that reveal hidden temporal patterns that are characteristic and predictive of time series events. Traditional time series analysis methods are limited by the requirement of stationarity of the time series and normality and independence of the residuals. Because they attempt to characterize and predict all time series observations, traditional time series analysis methods are unable to identify complex (nonperiodic, nonlinear, irregular, and chaotic) characteristics. TSDM methods overcome limitations of traditional time series analysis techniques. A brief historical review of related fields, including a discussion of the theoretical underpinnings for the TSDM framework, is made. The TSDM framework, concepts, and methods are explained in detail and applied to real-world time series from the engineering and financial domains.

Table of Contents

List of Tables

List of Figures

Glossary

X, Y	Time series
x_t, y_t	Time series observations at time index *t*
B	Backshift operator
Q	Phase space dimension, temporal pattern length
$\mathbb{R}, \mathbb{R}^Q$	The set of real numbers, real *Q*-space
τ	Embedding delay
$\mathbf{p}$	Temporal pattern
δ	Temporal pattern threshold, radius of temporal pattern cluster
d	Distance or metric defined on the phase space
P	Temporal pattern cluster
$\mathbf{x}_t$	Phase space point with time index *t*
$g(\cdot)$	Event characterization function
Λ	Index set of all of phase space points
M	Index set of phase space points within a temporal pattern cluster
$\tilde{M}$	Index set of phase space points outside a temporal pattern cluster
$c(M)$	Cluster cardinality
$c(\tilde{M})$	Non-cluster cardinality
μ_M	Cluster mean eventness
σ_M	Cluster standard deviation eventness
$\mu_{\tilde{M}}$	Non-cluster mean eventness
$\sigma_{\tilde{M}}$	Non-cluster standard deviation eventness
μ_X	Average eventness of all phase space points

$f(\cdot)$	Objective function
β	Percentage of the total phase space points
$b(\cdot)$	Repulsion force function, moderates δ
$\mathbf{X}$	Multi-dimensional time series
z_r	The test statistic for the runs test
α_r	Probability of a Type I error in rejecting the null runs test hypothesis
z_m	The test statistic for difference of two independent means test
α_m	Probability of a Type I error in rejecting the null difference of two independent means test hypothesis

Chapter 1 Introduction

The Time Series Data Mining (TSDM) framework, introduced by this dissertation, is a fundamental contribution to the fields of time series analysis and data mining. Methods based on the TSDM framework are able to successfully characterize and predict complex, nonperiodic, irregular, and chaotic time series. The TSDM methods overcome limitations (including stationarity and linearity requirements) of traditional time series analysis techniques by adapting data mining concepts for analyzing time series. This chapter reviews the definition of a time series, introduces the key TSDM concepts of events and hidden temporal patterns, and provides examples of problems the TSDM framework addresses.

A time series X is "a sequence of observed data, usually ordered in time" [1, p. 1].

$$X = \{x_t, t = 1, \ldots, N\}, \tag{1.1}$$

where t is a time index, and N is the number of observations. Time series analysis is fundamental to engineering, scientific, and business endeavors. Researchers study systems as they evolve through time, hoping to discern their underlying principles and develop models useful for predicting or controlling them. Time series analysis may be applied to the prediction of welding droplet releases and stock market price fluctuations [2, 3].

Traditional time series analysis methods such as the Box-Jenkins or Autoregressive Integrated Moving Average (ARIMA) method can be used to model such time series. However, the ARIMA method is limited by the requirement of stationarity of the time series and normality and independence of the residuals [1, 4, 5]. The statistical

characteristics of a stationary time series remain constant through time. Residuals are the errors between the observed time series and the model generated by the ARIMA method. The residuals must be uncorrelated and normally distributed.

For real-world time series such as welding droplet releases and stock market prices, the conditions of time series stationarity and residual normality and independence are not met. A severe drawback of the ARIMA approach is its inability to identify complex characteristics. This limitation occurs because of the goal of characterizing all time series observations, the necessity of time series stationarity, and the requirement of residual normality and independence.

Data Mining [6, 7] is the analysis of data with the goal of uncovering hidden patterns. Data Mining encompasses a set of methods that automate the scientific discovery process. Its uniqueness is found in the types of problems addressed – those with large data sets and complex, hidden relationships.

The new TSDM framework innovates data mining concepts for analyzing time series data. In particular, this dissertation describes a set of methods that reveal hidden patterns in time series data and overcome limitations of traditional time series analysis techniques. The TSDM framework focuses on predicting events, which are important occurrences. This allows the TSDM methods to predict nonstationary, nonperiodic, irregular time series, including chaotic deterministic time series. The TSDM methods are applicable to time series that appear stochastic, but occasionally (though not necessarily periodically) contain distinct, but possibly hidden, patterns that are characteristic of the desired events.

It is commonly assumed that the ARIMA time series models developed with past data will apply to future prediction. This is the stationarity assumption that models will not need to vary through time. ARIMA models also assume that the system generating the time series is linear, i.e., can be defined by linear differential or difference equations [8]. Unfortunately, the systems generating the time series are not necessarily linear or stationary.

In contrast, the TSDM framework and the methods built upon it can handle nonlinear and nonstationary time series. This framework is most useful for predicting events in a time series, which might include predicting when a droplet from a welder will release, when a stock price will drop, or when an induction motor adjustable speed drive system will fail. All these applications are well suited to this new framework and the methods built upon it.

The novel TSDM framework has its underpinnings in several fields. It builds upon concepts from data mining [6, 7], time series analysis [1, 4, 5], adaptive signal processing [9], wavelets [10-18], genetic algorithms [19-27], and chaos, nonlinear dynamics, and dynamical systems [28-35]. From data mining comes the focus on discovering hidden patterns. From time series analysis comes the theory for analyzing linear, stationary time series. In the end, the limitations of traditional time series analysis suggest the possibility of new methods. From adaptive signal processing comes the idea of adaptively modifying a filter to better transform a signal. This is closely related to wavelets. Building on concepts from both adaptive signal processing and wavelets, this dissertation develops the idea of a temporal pattern. From genetic algorithms comes a robust and easily applied optimization method [19]. From the study of chaos, nonlinear

dynamics, and dynamical systems comes the theoretical justification of the method, specifically Takens' Theorem [36] and Sauer's extension [37].

1.1 Data Mining Analogy

An analogy to gold mining helps clarify the problem and introduces two key data mining concepts. An analogy is the assumption that if two things are similar in one area, they will be similar in others. The use of the term data mining implies an analogy with gold mining. There are several parallels between the time series analysis problems discussed in this dissertation and this analogy.

As gold mining is the search for nuggets of gold, so data mining is the search for nuggets of information. In mining time series data, these nuggets are known as events. As gold is hidden in the ground or under water, nuggets of information are hidden in data. The first analogy is gained by comparing the definition of the gold nuggets with the definition of information nuggets. To the inexperienced miner, gold is gold, but to a veteran prospector, the size of the gold nuggets to be uncovered make a significant difference in how the gold mining is approached. Individual prospectors use primarily manual methods when looking for nuggets of gold that are ounces in weight [38]. Industrial mining companies may find it acceptable to look for gold at the molecular level [39]. Likewise, if a prospector is seeking silver or oil, the mining processes are different. This leads to the importance of clearly defining the nuggets of information that are desired, i.e., time series data mining requires a clear definition of the events to be mined. Without this clear definition of what is to be found, there is no way to know when either the gold nuggets or the information nuggets have been discovered.

The second analogy looks at how prospectors learn where to search for the gold nuggets. Prospectors look for specific geological formations such as quartz and ironstone, and structures such as banded iron formations [38]. They study where other prospectors have had success. They learn not to dig aimlessly, but to look for clues that a particular location might yield a gold strike. Similarly, it is necessary to define the formations that point to nuggets of information (events). In the context of time series analysis these, probably hidden, formations that identify an information strike are called temporal patterns – temporal because of the time nature of the problem and patterns because of their identifiable structure. Like gold prospectors, information prospectors understand that the clues need not be perfect, rather the clues need only to contribute to the overall effectiveness of the prediction.

The two analogies lead us to identify two key concepts and their associated requirements for data mining time series. The first concept is that of an event, which is an important occurrence. A clear definition of an event is required. The second concept is that of a temporal pattern, which is a potentially hidden structure in a time series. The temporal patterns are required to help predict events.

With the key TSDM concepts of events and temporal patterns defined, the next section presents the types of problems addressable by the TSDM framework.

1.2 Problem Statement

Figure 1.1 illustrates a TSDM problem, where the horizontal axis represents time, and the vertical axis observations. The diamonds show the time series observations. The squares indicate observations that are deemed important – events. Although the following

examples illustrate events as single observations, events are not restricted to be just single observations. The goal is to characterize and predict when important events will occur. The time series events in Figure 1.1 are nonperiodic, irregular, and contaminated with noise.

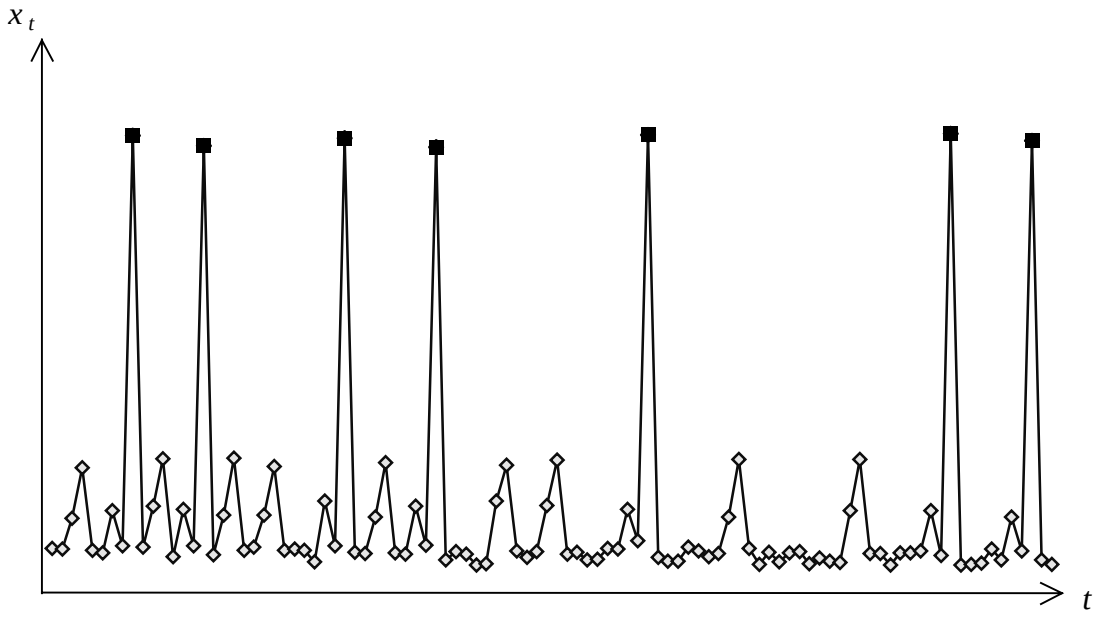

Figure 1.1 – Synthetic Seismic Time Series

To make the time series more concrete, consider it a measure of seismic activity, which is generated from a randomly occurring temporal pattern, synthetic earthquake, and a contaminating noise signal The goal is to characterize when peak seismic activity (earthquakes) occurs and then use the characterizations of the activity for prediction.

The next example of the type of problem the TSDM framework can solve is from the engineering domain. Figure 1.2 illustrates a welding time series generated by a sensor on a welding station. Welding joins two pieces of metal by forming a joint between them.

Predicting when a droplet of metal will release from a welder allows the quality of the metal joint to be monitored and controlled.

In Figure 1.2, the squares indicate the release of metal droplets. The diamonds are the stickout length of the droplet measured in pixels. The problem is to predict the releases using the stickout time series. Because of the irregular, chaotic, and noisy nature of the droplet release, prediction is impossible using traditional time series methods.

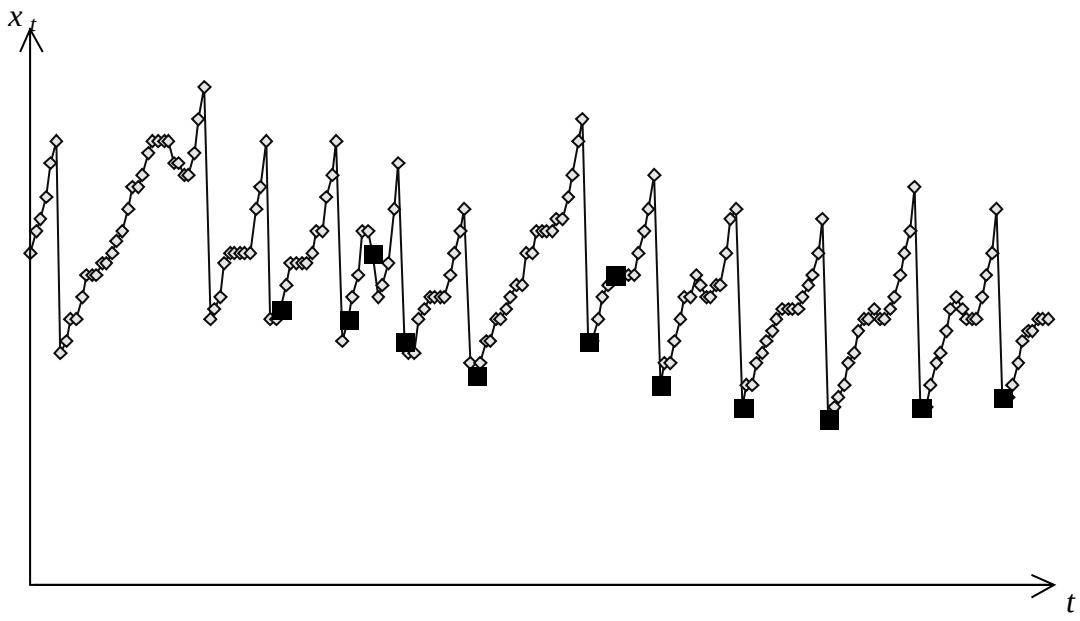

Figure 1.2 – Welding Time Series

Another example problem that is addressed by the TSDM framework is the prediction of stock prices. For this problem, the goal is to find a trading-edge, which is a small advantage that allows greater than expected gains to be realized. The goal is to find hidden temporal patterns that are on average predictive of a larger than normal increase in the price of a stock. Figure 1.3 shows a time series generated by the daily open price and volume of a stock. The bars show the volume of shares traded on a particular day. The

diamonds show the daily open price. The goal is to find hidden patterns in the daily open price and volume time series that provide the desired trading-edge.

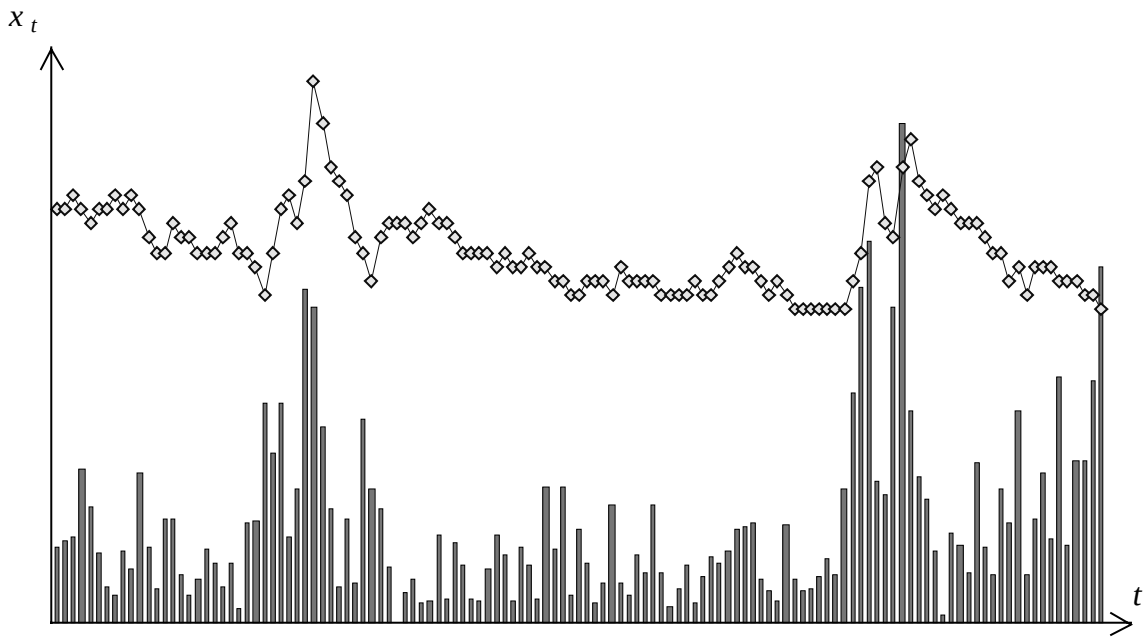

Figure 1.3 – Stock Daily Open Price and Volume Time Series

Now that examples of the types of problems addressable by the TSDM framework have been presented, the next section outlines the rest of the dissertation.

1.3 Dissertation Outline

The dissertation is divided into nine chapters. Chapter 2 reviews several of the constituent technologies underlying this research including time series analysis, data mining, and genetic algorithms. Additionally, Chapter 2 presents the theoretical background for the TSDM framework, reviewing Takens' Theorem.

Chapter 3 elaborates on the key TSDM concepts of events, temporal patterns, temporal pattern clusters, phase spaces and time-delay embeddings, augmented phase spaces, objective functions, and optimization.

Chapter 4 establishes the fundamental TSDM method for characterizing and predicting time series events. Chapter 5 clarifies the TSDM framework by analyzing a sequence of example time series. In Chapter 6, extensions of the TSDM method including data mining multiple time series and nonstationary temporal pattern time series are presented.

Chapters 7 and 8 discuss experimental results. Chapter 7 presents results from predicting droplet releases from a welder. In Chapter 8, the experimental results from analyzing stock market open price changes are presented. The last chapter summarizes the dissertation and discusses future work.

Chapter 2 Historical Review

This chapter reviews the constituent fields underlying the Time Series Data Mining (TSDM) research. TSDM innovates concepts from time series analysis, chaos and nonlinear dynamics, data mining, and genetic algorithms. From time series analysis comes the theory for analyzing linear, stationary time series [1, 4, 5]. From dynamical systems comes the theoretical justification for the Time Series Data Mining (TSDM) methods, specifically Takens' Theorem [36] and Sauer' s extension [37]. From data mining comes the focus on discovering hidden relationships and patterns [6, 7, 40-44]. From genetic algorithms comes a robust and easily applied optimization method [19, 27].

2.1 ARIMA Time Series Analysis

The Box-Jenkins [4] or Autoregressive Integrated Moving Average (ARIMA) [1, 5] methodology involves finding solutions to the difference equation

$$\phi_p(B)\phi_P\left(B^L\right)x_t = \delta + \theta_q(B)\theta_Q\left(B^L\right)a_t \text{ [5, p. 570].} \quad (2.1)$$

- The nonseasonal autoregressive operator $\phi_p(B)$ of order p models low-order feedback responses.
- The seasonal autoregressive operator $\phi_P(B^L)$ of order P models feedback responses that occur periodically at seasonal intervals. For example, given a time series of monthly data, this operator would be used to model a regressive effect that occurs every January.
- The nonseasonal moving average operator $\theta_q(B)$ of order q models low-order weighted average responses.

- The seasonal moving average operator $\theta_Q(B^L)$ of order Q models seasonal weighted average responses.
- The terms x_t , a_t, and δ are the time series, a sequence of random shocks, and a constant, respectively.

The orders of the operator are selected ad hoc, and the parameters are calculated from the time series data using optimization methods such as maximum likelihood [4, pp. 208-209,274-281] and least squares [4, pp. 265-267]. The ARIMA method is limited by the requirement of stationarity and invertibility of the time series [5, p. 488], i.e., the system generating the time series must be time invariant and stable. Additionally, the residuals, the differences between the time series and the ARIMA model, must be independent and distributed normally [5, p. 183-193]. Although integrative (filtering) techniques can be useful for converting nonstationary time series into stationary ones, it is not always possible to meet all of the requirements.

This review of ARIMA time series modeling examines each of the terms given in (2.1), discusses the methods for identifying the orders of the various operators, and details the various statistical methods available to test the model's adequacy. Finally, this section discusses the integrative techniques that allow some nonstationary time series to be transformed into stationary ones.

The ARIMA model is best presented in terms of the following operators [4, p. 8, 5, p. 568]. The backshift operator B shifts the index of a time series observation backwards, e.g., $Bz_t = z_{t-1}$, and $B^k z_t = z_{t-k}$. The nonseasonal or first difference operator, $\nabla = 1 - B$, provides a compact way of describing the first difference. The seasonal

operator ∇_L is useful for taking the difference between two periodic or seasonal time series observations. It is defined as $\nabla_L = 1 - B^L$.

Having introduced the basic operator notation, the more complex operators presented in (2.1) can be discussed. The first operator from (2.1) is the nonseasonal autoregressive operator $\phi_p(B)$ [4, p. 9, 5, p. 570], also called the "Green's function" [1, p. 78]. This operator captures the systems dynamical response to a_t – the sequence of random shocks – and previous values of the time series [1, pp. 78-85]. The second operator is the nonseasonal moving average operator $\theta_q(B)$ [5, p. 570]. It is a weighted moving average of the random shocks a_t.

The third operator is the seasonal autoregressive operator $\phi_P(B^L)$. It is used to model seasonal regressive effects. For example, if the time series represents the monthly sales in a toy store, it is not hard to imagine a large increase in sales just before Christmas. This seasonal autoregressive operator is used to model these seasonal effects. The fourth operator is the seasonal moving average operator $\theta_Q(B^L)$. It also is useful in modeling seasonal effects, but instead of regressive effects, it provides a weighted average of the seasonal random shocks. The constant $\delta = \mu\phi_p(B)\phi_P(B)$, where μ is the mean of the modeled stationary time series [5, p. 571].

Bowerman [5, pp. 571] suggests three steps to determine the ARIMA model for a particular time series.

1. Should the constant δ should be included?
2. Which of the operators $\phi_p(B)$, $\phi_P(B^L)$, $\theta_q(B)$, and $\theta_Q(B^L)$ are needed?
3. What order should each selected operator have?

The δ should be included if

$$\frac{\mu(Z)\sqrt{c(Z)}}{\sigma_z} > 2, \tag{2.2}$$

where $\mu(Z)$ is the mean of the time series, $c(Z)$ is the number of time series observations, and σ_z is the standard deviation of the time series. Two statistical functions, the sample autocorrelation function (SAC) and sample partial autocorrelation function (SPAC), are used to determine the inclusion and order of the operators. The process for determining the inclusion and orders of the operators is somewhat involved and well explained in [5, pp. 572-574]. Its essence is to examine the shape of the SAC and SPAC. The procedure looks for these functions to "die down" or "cut off" after a certain number of lags. Determining whether the SAC or SPAC is dying down or cutting off requires expert judgment.

After the operators have been selected and their orders determined, the coefficients of the operators are estimated using a training time series. The coefficients are estimated using a least squares [4, pp. 265-267] or maximum likelihood method [4, pp. 208-209, 274-281].

Diagnostic checking of the overall ARIMA model is done by examining the residuals [5, p. 496]. The first diagnostic check is to calculate the Ljung-Box statistic. Typically, the model is rejected when the α corresponding to the Ljung-Box statistic is less than 0.05. For non-rejected models, the residual sample autocorrelation function (RSAC) and residual sample partial autocorrelation function (RSPAC) should have absolute *t* statistic values greater than two [5, p. 496]. For rejected models, the RSAC and

RSPAC can be used to suggest appropriate changes to enhance the adequacy of the models.

"Classic Box-Jenkins models describe stationary time series [5, p. 437]." However, several integrative or filtering methods transform nonstationary time series into stationary ones. The simplest nonstationary time series to make stationary is a linear trend, which is nonstationary because its mean varies through time. The nonseasonal operator ∇ or seasonal operator ∇_L is applied to remove the linear trend.

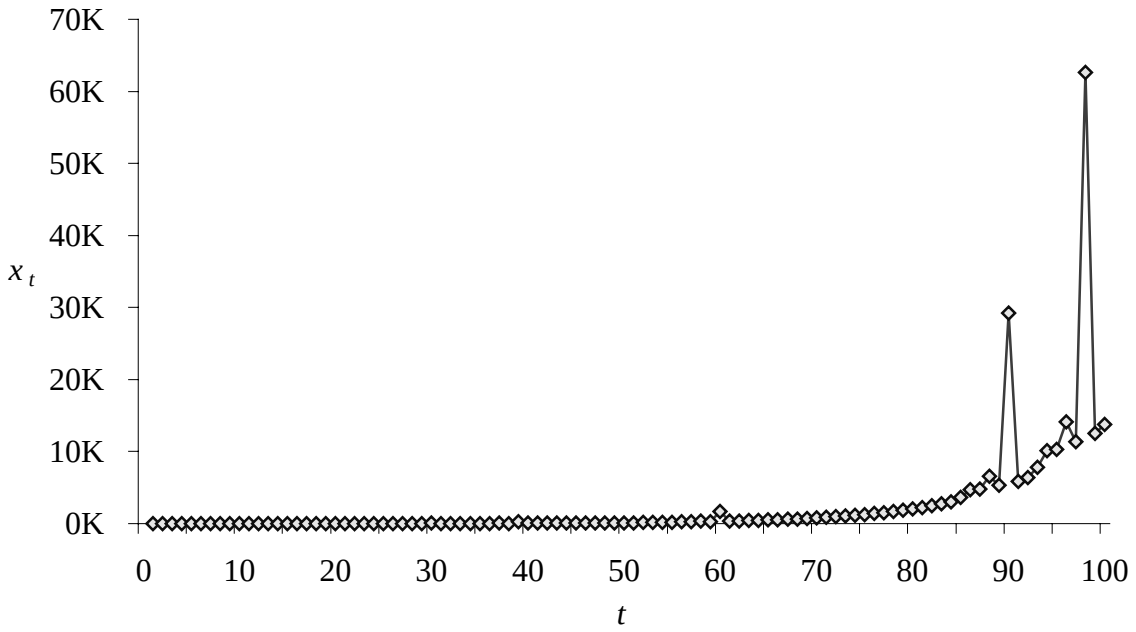

Figure 2.1 – Exponential Growth Time Series

A slightly more complex transformation is required for an exponential trend. One method takes the logarithm of the time series and applies the appropriate nonseasonal or seasonal operator to the resulting linear trend time series. Alternatively, the $\Delta^{\%}$ change transform may be used, where

$$\Delta^{\%} = \frac{1-B}{B}. \tag{2.3}$$

The transform is applied as follows:

$$z_t = \Delta^{\%} x_t = \frac{1-B}{B} x_t = \frac{x_t - x_{t-1}}{x_{t-1}}. \tag{2.4}$$

Figure 2.1 shows a time series with exponential growth. Figure 2.2 illustrates the transformed time series.

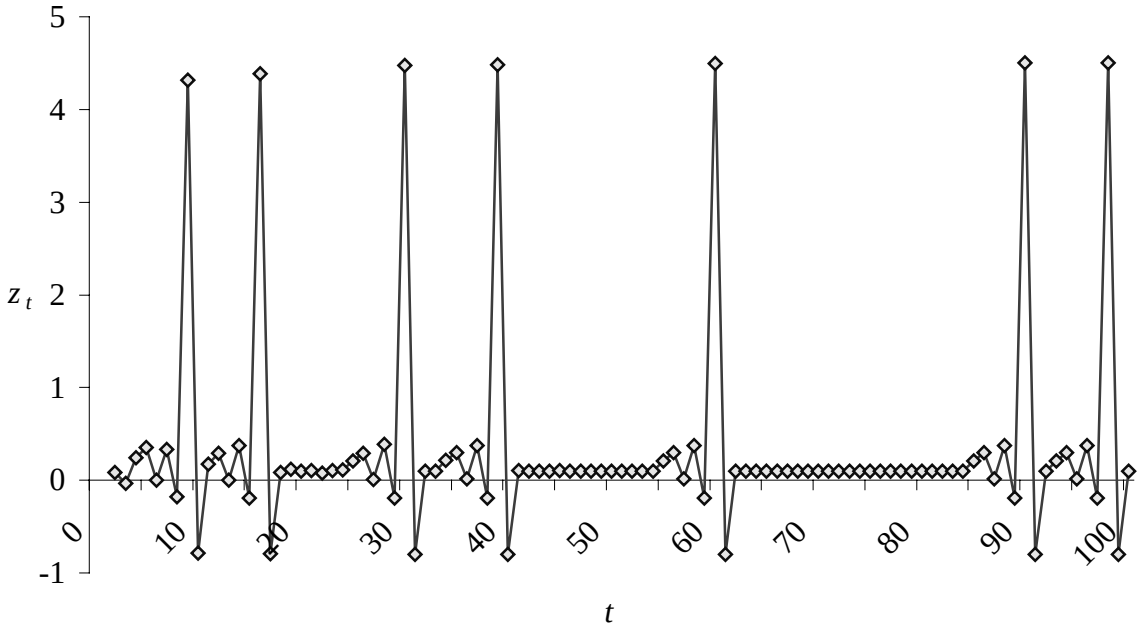

Figure 2.2 – Filtered Exponential Growth Time Series

For time series with nonstationary variances, there are two possible solutions. The first is to replace the time series with the square or some other appropriate root of the time series. Second, the time series may be replaced by its logarithm [5, pp. 266-270].

Given an adequate model, future time series values may be predicted using (2.1). An error confidence range may also be provided.

This section has reviewed the ARIMA or Box-Jenkins time series analysis method. The three references cited here [1, 4, 5] are excellent sources for further study of this topic. As discussed in this section, optimization methods are needed to find the

parameters for the ARIMA model. Similarly, optimization is a necessary component of the Time Series Data Mining (TSDM) framework. The next section presents the genetic algorithm optimization method used in TSDM.

2.2 Genetic Algorithms

A genetic algorithm is a stochastic optimization method based on the evolutionary process of natural selection. Although a genetic algorithm does not guarantee a global optimum, it is known to be effective in optimizing non-linear functions [19, pp. 106-120]. TSDM requires an optimization method to find optimizers for the objective functions. Genetic algorithm optimization is selected for this purpose because of its effectiveness and ease of adaptation to the objective functions posed by the TSDM framework.

This section briefly discusses the key concepts and operators used by a binary genetic algorithm [19, pp. 59-88, 22, pp. 25-48, 23, pp. 33-44, 24, pp. 42-65]. The genetic algorithm process also is discussed. The four major operators are selection, crossover, mutation, and reinsertion. The fifth operator, inversion, is used infrequently. The concepts of genetic algorithms are fitness or objective function, chromosome, fitness of a chromosome, population, and generation.

The fitness function is the function to be optimized, such as

$$f(x) = -x^2 + 10x + 10000. \tag{2.5}$$

A chromosome is a finite sequence of 0's and 1's that encode the independent variables appearing in the fitness function. For equation (2.5), the chromosomes represent values of *x*. Given an eight-bit chromosome and a two's complement encoding, the values of *x* for several chromosomes are given in Table 2.1.

Chromosome	x	$f(x)$, fitness
10000000	-128	-7664
00000000	0	10000
01111111	127	-4859
11111100	-4	9944

Table 2.1 – Chromosome Fitness Values

The fitness is the value assigned to a chromosome by the fitness function. The population is the set of all chromosomes in a particular generation, e.g., the four chromosomes in Table 2.1 form a population. A generation is an iteration of applying the genetic algorithm operators.

The most common genetic algorithm process is defined as follows. Alternative genetic algorithm processes may reorder the operators.

Initialization

while stopping criteria are not met

Selection

Crossover

Mutation

Reinsertion

The initialization step creates, usually randomly, a set of chromosomes, as in Table 2.1. There are many possible stopping criteria, e.g., halting after a fixed number of generations (iterations) or when fitness values of all chromosomes are equivalent.

The selection process chooses chromosomes from the population based on fitness. One selection process is based on a roulette wheel. The roulette wheel selection process

gives each chromosome a portion of the roulette wheel based on the chromosome's fitness. The roulette wheel is spun, and the winning chromosome is placed in the mating or crossover population. Usually the individuals are selected with replacement, meaning any chromosome can win on any spin of the roulette wheel.

The second type of selection is based on a tournament. In the tournament, *n* chromosomes – usually two – are selected at random, normally without replacement. They compete based on fitness, and the winner is placed in the mating or crossover population. This process is repeated until there are no individuals left. The whole tournament process is run *n* times, where *n* is the number of chromosomes in each round of the tournament. The output of the selection process is a mating population, which is usually the same size as the original population.

Given the initial population from Table 2.1, a tournament without replacement is demonstrated in Table 2.2. The crossover population is formed from the winners.

Tournament	Round	Competitor 1	Competitor 2	Winner
1	1	10000000 (-7664)	01111111 (-4859)	01111111
1	2	00000000 (10000)	11111100 (9944)	00000000
2	1	01111111 (-4859)	11111100 (9944)	11111100
2	2	00000000 (10000)	10000000 (-7664)	00000000

Table 2.2 – Tournament Selection Example

Crossover is the process that mixes the chromosomes in a manner similar to sexual reproduction. Two chromosomes are selected from the mating population without replacement. The crossover operator combines the encoded binary format of the parent chromosomes to create offspring chromosomes. A random crossover locus is chosen, and

the parent chromosomes are split at the locus. The tails of the chromosomes are swapped, yielding new chromosomes that share the genetic material from their parents. Figure 2.3 shows the crossover process.

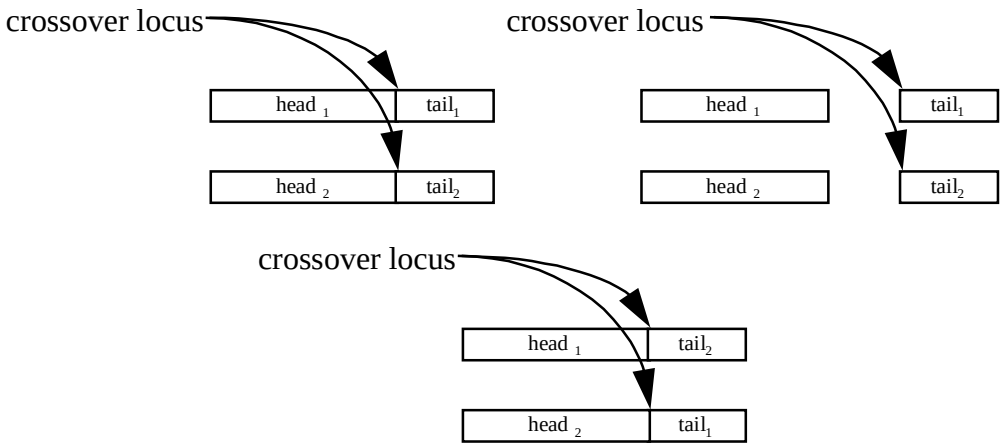

Figure 2.3 – Chromosome Crossover

A variation on the crossover process includes using a fixed rather than random locus and/or using a crossover probability that the selected pair will not be mated.

Continuing the example, the crossover process is illustrated in Table 2.3, where ↑ is the crossover locus.

Mating Pair	Parent 1	Parent 2	Offspring 1	Offspring 1
1	11111↑100	00000↑000	00000100	11111000
2	000↑00000	011↑11111	00011111	01100000

Table 2.3 – Crossover Process Example

The mutation operator randomly changes the bits of the chromosomes. The mutation probability is usually set in the range of 0.1 to 0.01%. For the running example, the mutation process is shown in Table 2.4, where only one bit is mutated.

Pre-mutation	Post-mutation
00000100	00000100
111**1**1000 111	**0**1000
00011111	00011111
01100000	01100000

Table 2.4 – Crossover Process Example

The reinsertion or elitism operator selects the top *n* chromosomes to bypass the selection, crossover, and mutation operations. By applying elitism, the top individuals pass directly from one generation to the next unmodified. This operator is used to ensure that the most fit individuals are not lost due to the stochastic nature of the selection and crossover processes.

For the example, no reinsertion is used. The next generation with fitness values is presented in Table 2.5. A comparison of Table 2.1 and Table 2.5 show that better solutions have evolved through the genetic algorithm process.

Chromosome	*x*	*f*(*x*), fitness
00000100	4	10024
11101000	-24	9184
00011111	31	9349
01100000	96	1744

Table 2.5 – Resulting Genetic Algorithm Population

In summary, a genetic algorithm is a stochastic, global optimization method based on the evolutionary theory of survival of the fittest. The genetic algorithm applies four operators (selection, crossover, mutation, and reinsertion) to search for objective function

optimizers. The use of an optimization method will form a key component of the TSDM framework, specifically in finding the hidden temporal patterns introduced in Chapter 1. The next section presents the theoretical justification for searching for these hidden temporal patterns.

2.3 Theoretical Underpinnings of Time Series Data Mining

This section shows how Takens' Theorem provides the theoretical justification for the TSDM framework. Takens proved, with certain limitations, that the state space of an unknown system can be reconstructed [36, 37].

Theorem (Takens) [36]: Let the state space M of a system be Q dimensional, $\varphi : M \to M$ be a map that describes the dynamics of the system, and $y : M \to \mathbb{R}$ be a twice continuously differentiable function, which represents the observation of a single state variable. The map $\Phi_{(\varphi,y)} : M \to \mathbb{R}^{2Q+1}$, defined by

$$\Phi_{(\varphi,y)}(x) = \left(y(x), y(\varphi(x)), \ldots, y\left(\varphi^{2Q}(x)\right)\right), \tag{2.6}$$

is an embedding. An embedding is a homeomorphic mapping from one topological space to another [45, pp. 679-680], where a homeomorphic map is continuous, bijective (one-to-one and onto), and its inverse is continuous [45, pp. 1280].

If the embedding is performed correctly, Takens' Theorem guarantees that the reconstructed dynamics are topologically identical to the true dynamics of the system. Therefore, the dynamical invariants also are identical [46]. Hence, given a time series X, a state space topologically equivalent to the original state space can be reconstructed by a process called time-delay embedding [28, 37].

The difficulty in the time-delay embedding process is in estimating Q, the original state space dimension. Fortunately, as shown in [2, 3, 28, 46], useful information can be extracted from the reconstructed state space even if its dimension is less than $2Q+1$.

This dissertation uses Takens' Theorem to provide the strong theoretical justification for reconstructing state spaces using time-delay embedding. The dynamics of the reconstructed state spaces can contain the same topological information as the original state space. Therefore, characterizations and predictions based on the reconstructed state space can be as valid as those that could be performed on the original state space. This is true even for chaotic dynamics, which are discussed in the next section.

2.4 Chaotic Time Series

The most interesting time series presented in this dissertation may be classified as chaotic. (See Chapters 7 and 8.) This section provides a definition and discussion of chaotic time series.

"Chaos comprises a class of signals intermediate between regular sinusoidal or quasiperiodic motions and unpredictable, truly stochastic behavior [28, p. 11]." A working definition of a chaotic time series is one generated by a nonlinear, deterministic process highly sensitive to initial conditions that has a broadband frequency spectrum [28].

The language for describing chaotic time series comes from dynamical systems theory, which studies the trajectories described by flows (differential equations) and maps (difference equations), and nonlinear dynamics, an interdisciplinary field that applies

dynamical systems theory in numerous scientific fields [30]. The key concept for describing chaotic time series is a chaotic attractor.

Let *M* be a manifold (a smooth geometric space such as a line, smooth surface or solid [30, p. 10]), $f : M \to M$ be a map, and

$$S = \{x_0 : x_0 \in S,\ f^n(x_0) \in S, \forall n\} \subset M \tag{2.7}$$

be an invariant set. A positively invariant set is one where $n \geq 0$. A closed invariant set $A \subset M$ is an attracting set, if there exists a neighborhood *U* of *A* such that *U* is a positively invariant set, and $f^n(x) \to A\ \forall x \in U.$ A dense orbit is a trajectory that passes arbitrarily close to every point in the set [30]. An attractor is defined as an attracting set that contains a dense orbit. Figure 2.4 illustrates the concept of an attractor with the arrows representing state trajectories.

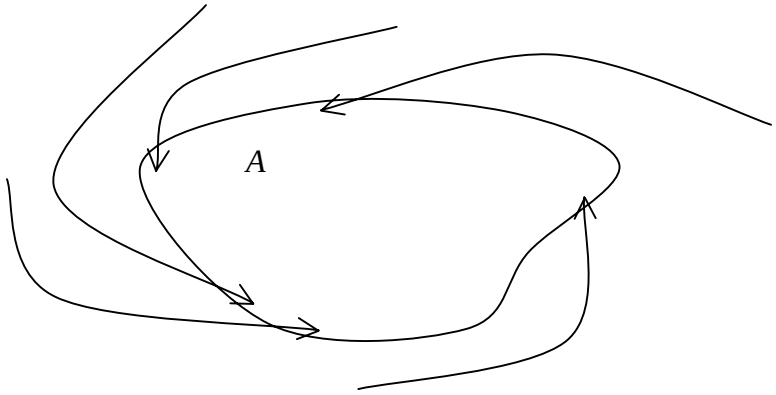

Figure 2.4 - Attractor

Thus, a chaotic time series is defined as one generated by observing a state variable's trajectory on a map with a chaotic attractor. Since a chaotic time series is deterministic, it is predictable. However, since it is highly dependent on initial conditions, the prediction horizon is very short. The TSDM framework provides methods that use Takens' Theorem to exploit the short-term predictability of chaotic time series. The next

section presents data mining, which leads to the idea of searching in the short time horizon where chaotic time series are predictable.

2.5 Data Mining

Weiss and Indurkhya define data mining as "the search for valuable information in large volumes of data. Predictive data mining is a search for very strong patterns in big data that can generalize to accurate future decisions [7]." Similarly, Cabena, *et al.*, define it as "the process of extracting previously unknown, valid, and actionable information from large databases and then using the information to make crucial business decisions [43]."

Data mining evolved from several fields, including machine learning, statistics, and database design [7]. It uses techniques such as clustering, association rules, visualization, decision trees, nonlinear regression, and probabilistic graphical dependency models to identify novel, hidden, and useful structures in large databases [6, 7].

Others who have applied data mining concepts to finding patterns in time series include Berndt and Clifford [47], Keogh [48-50], and Rosenstein and Cohen [51]. Berndt and Clifford use a dynamic time warping technique taken from speech recognition. Their approach uses a dynamic programming method for aligning the time series and a predefined set of templates.

Rosenstein and Cohen [51] also use a predefined set of templates to match a time series generated from robot sensors. Instead of using the dynamic programming methods as in [47], they employ the time-delay embedding process to match their predefined templates.

Similarly, Keogh represents the templates using piecewise linear segmentations. "Local features such as peaks, troughs, and plateaus are defined using a prior distribution on expected deformations from a basic template [48]." Keogh's approach uses a probabilistic method for matching the known templates to the time series data.

The TSDM framework, initially introduced by Povinelli and Feng in [3], differs fundamentally from these approaches. The approach advanced in [47-51] requires *a priori* knowledge of the types of structures or temporal patterns to be discovered and represents these temporal patterns as a set of templates. Their [47-51] use of predefined templates completely prevents the achievement of the basic data mining goal of discovering useful, novel, and hidden temporal patterns.

The next chapter introduces the key TSDM concepts, which allow the TSDM methods to overcome the limitations of traditional time series methods and the more recent approaches of Berndt and Clifford [47], Keogh [48-50], and Rosenstein and Cohen [51].

Chapter 3 Some Concepts in Time Series Data Mining

Chapter 1 presented two of the important concepts in Time Series Data Mining (TSDM), i.e., events and temporal patterns. In this chapter, these concepts are explained in further detail. Other fundamental TSDM concepts such as event characterization function, temporal pattern cluster, time-delay embedding, phase space, augmented phase space, objective function, and optimization are defined and explained. The chapter also provides examples of each concept.

3.1 Events

In a time series, an event is an important occurrence. The definition of an event is dependent on the TSDM goal. In a seismic time series, an earthquake is defined as an event. Other examples of events include sharp rises or falls of a stock price or the release of a droplet of metal from a welder.

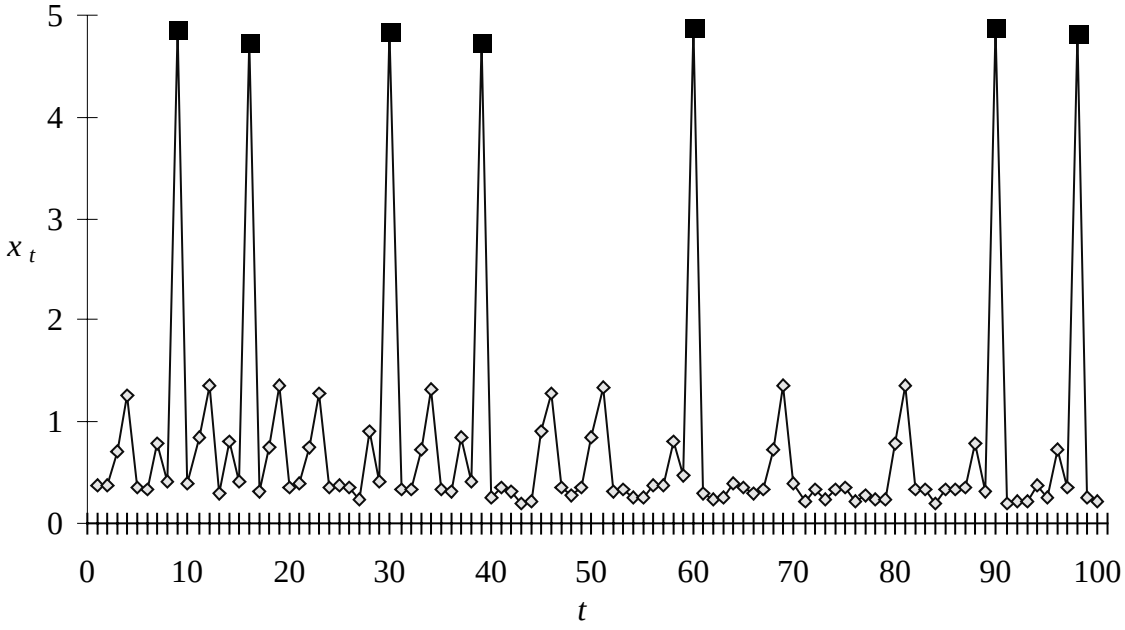

Figure 3.1 – Synthetic Seismic Time Series with Events

3.1.1 Event Example – Synthetic Earthquakes

Figure 3.1 shows a synthetic example time series, which is useful for explaining events. Let

$$X = \{x_t, t = 1, \ldots, N\} \tag{3.1}$$

be a synthetic time series representing seismic data, where $N = 100$. The diamonds show the values of observations at particular time indices. The squares indicate observations that are deemed important – events.

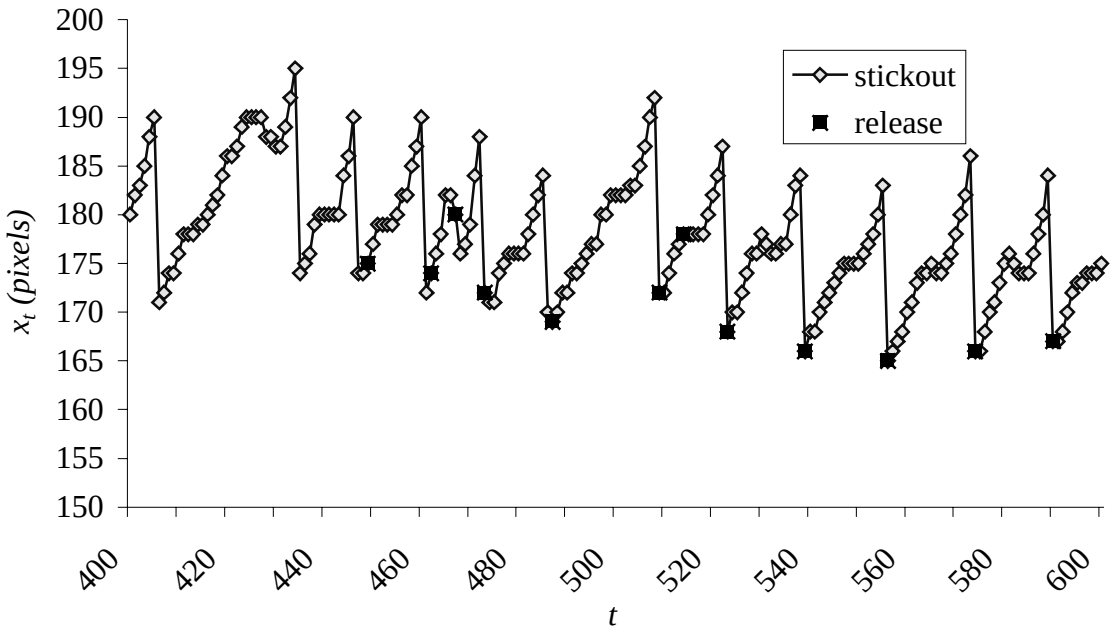

Figure 3.2 – Welding Time Series

3.1.2 Event Example – Metal Droplet Release

Figure 3.2 shows a welding time series. Let

$$X = \{x_t, t = 400, \ldots, 600\} \tag{3.2}$$

be a time series of metal droplet stickout lengths. The diamonds in Figure 3.2 are the stickout lengths measured in pixels. Let

$$Y = \{y_t, t = 400, \ldots, 600\} \tag{3.3}$$

be a binary (1 for an event, 0 for a nonevent) time series of droplet releases. In Figure 3.2, the squares indicate when $y_t = 1$, i.e., when a droplet of metal has released.

3.1.3 Event Example – Spikes in Stock Open Price

Let $X = \{x_t, t = 1,\ldots,126\}$ be the daily open price of a stock for a six-month period as illustrated by Figure 3.3. For this time series, the goal is to find a trading-edge, which is a small advantage that allows greater than expected gains to be realized. The stock will be bought at the open of the first day and sold at the open of the second day. The goal is to pick buy-and-sell-days that will, on average, have greater than expected price increases. Thus, the events, highlighted as squares in Figure 3.3, are those days when the price increases more than 5%.

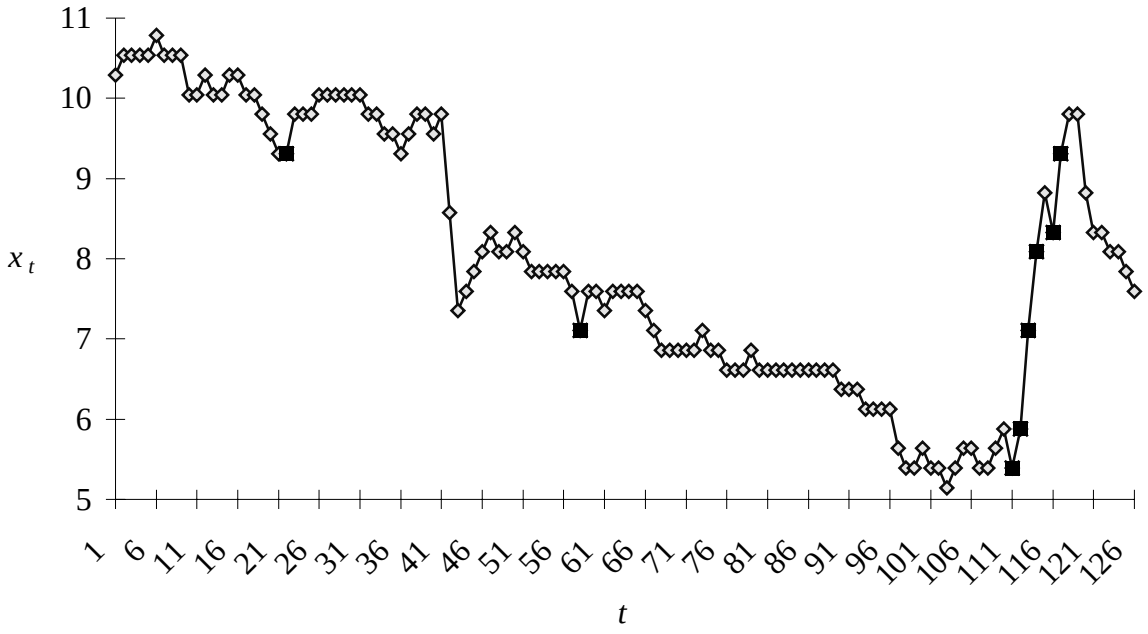

Figure 3.3 – Stock Daily Open Price Time Series

3.2 Temporal Pattern and Temporal Pattern Cluster

The next important concept within the TSDM framework is the temporal pattern. A temporal pattern is a hidden structure in a time series that is characteristic and predictive of events. The temporal pattern **p** is a real vector of length *Q*. The temporal pattern will be represented as a point in a *Q* dimensional real metric space, i.e., $\mathbf{p} \in \mathbb{R}^Q$.

The vector sense of **p** is illustrated in Figure 3.4, which shows the synthetic seismic time series without any contaminating noise. The hidden temporal pattern **p** that is characteristic of the events is highlighted with gray squares. Since the contaminating noise has been removed, the temporal pattern perfectly matches the sequence of time series observations before an event.

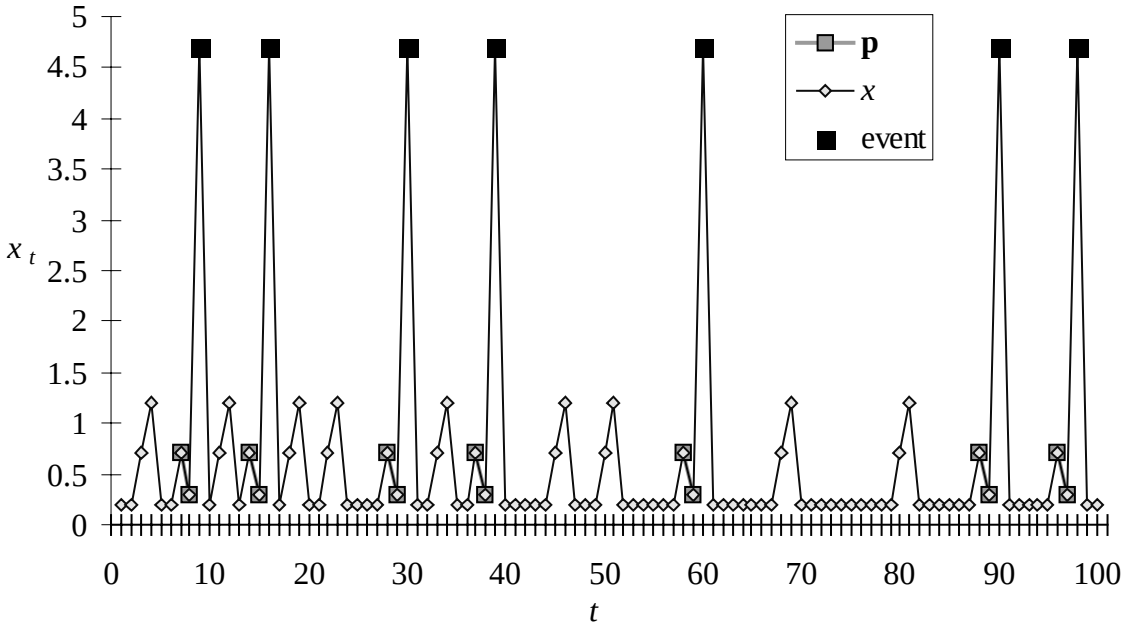

Figure 3.4 – Synthetic Seismic Time Series without Contaminating Noise with Temporal Pattern and Events

Figure 3.5 shows the synthetic seismic time series with contaminating noise. Because of the noise, the temporal pattern does not perfectly match the time series

observations that precede events. To overcome this limitation, a temporal pattern cluster is defined as the set of all points within δ of the temporal pattern.

$$P = \left\{ a \in \mathbb{R}^Q : d(\mathbf{p}, a) \leq \delta \right\}, \tag{3.4}$$

where *d* is the distance or metric defined on the space. This defines a hypersphere of dimension *Q*, radius δ, and center **p**.

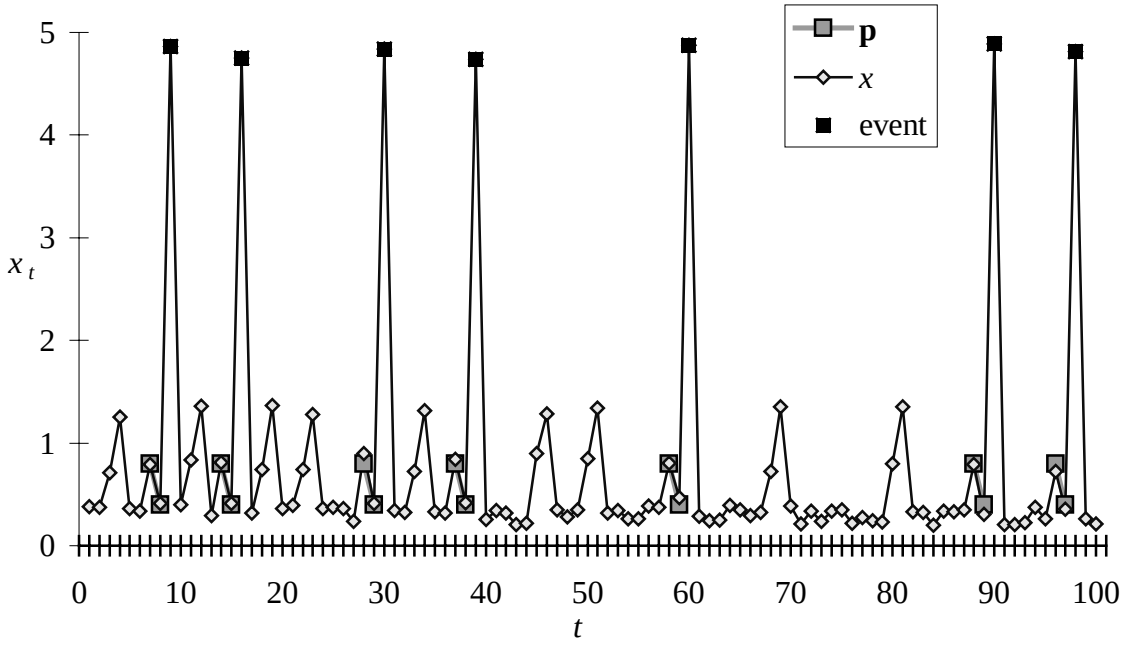

Figure 3.5 – Synthetic Seismic Time Series with Temporal Pattern and Events

The observations $\left\{ x_{t-(Q-1)\tau}, \ldots, x_{t-2\tau}, x_{t-\tau}, x_t \right\}$ form a sequence that can be compared to a temporal pattern, where x_t represents the current observation, and $x_{t-(Q-1)\tau}, \ldots, x_{t-2\tau}, x_{t-\tau}$ past observations. Let $\tau > 0$ be a positive integer. If *t* represents the present time index, then $t - \tau$ is a time index in the past, and $t + \tau$ is a time index in the future. Using this notation, time is partitioned into three categories: past, present, and future. Temporal patterns and events are placed into different time categories. Temporal patterns occur in the past and complete in the present. Events occur in the future.

The next section presents the concept of a phase space, which allows sequences of time series to be easily compared to temporal patterns.

3.3 Phase Space and Time-Delay Embedding

A reconstructed phase space [28, 35, 52], called simply phase space here, is a *Q*-dimensional metric space into which a time series is embedded. As discussed in Chapter 2, Takens showed that if *Q* is large enough, the phase space is homeomorphic to the state space that generated the time series [36]. The time-delayed embedding of a time series maps a set of *Q* time series observations taken from *X* onto $\mathbf{x}_t$, where $\mathbf{x}_t$ is a vector or point in the phase space. Specifically, $\mathbf{x}_t = \left(x_{t-(Q-1)\tau}, \ldots, x_{t-2\tau}, x_{t-\tau}, x_t\right)^T$.

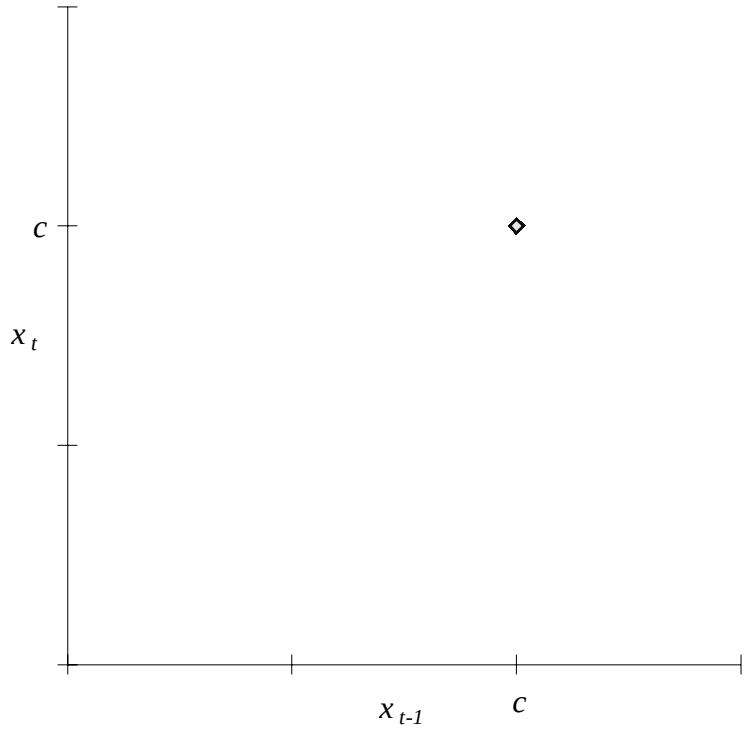

Figure 3.6 – Constant Value Phase Space

For example, given a constant value time series $X = \{x_t = c, t = 1, \ldots, N\}$, where *c* is a constant, the phase space has a single point as illustrated by Figure 3.6. Figure 3.7

shows a two-dimensional phase space that results from the time-delayed embedding of the synthetic seismic time series presented in Figure 3.1. The temporal pattern and temporal pattern cluster also are illustrated. For this time-delayed embedding, $\tau = 1$. Every pair of adjacent observations in the original time series forms a single point in this phase space.

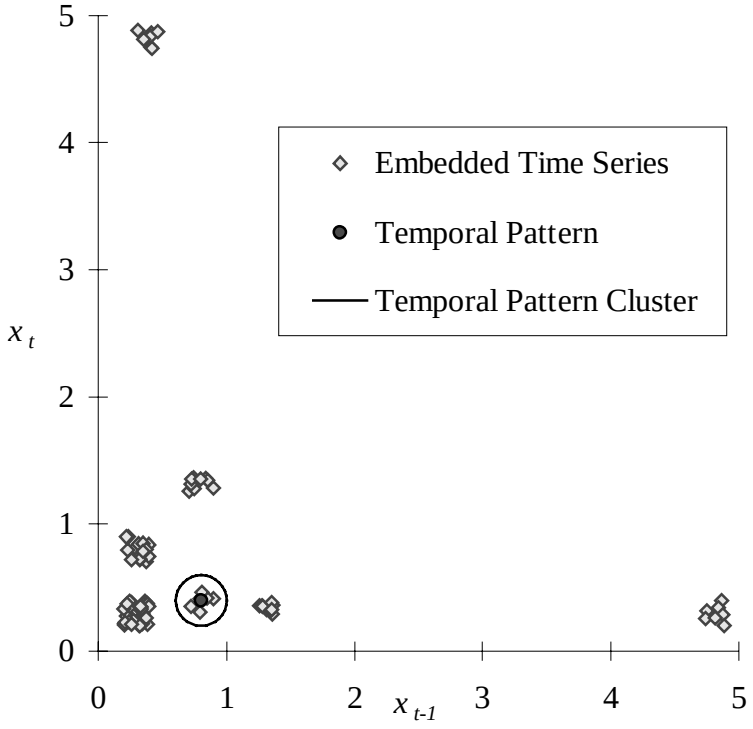

Figure 3.7 – Synthetic Seismic Phase Space

Figure 3.8 shows the two-dimensional phase space of the welding time series presented by Figure 3.2, and Figure 3.9 shows the two-dimensional phase space of the stock time series presented by Figure 3.3. Note that $\tau = 1$ for both embeddings.

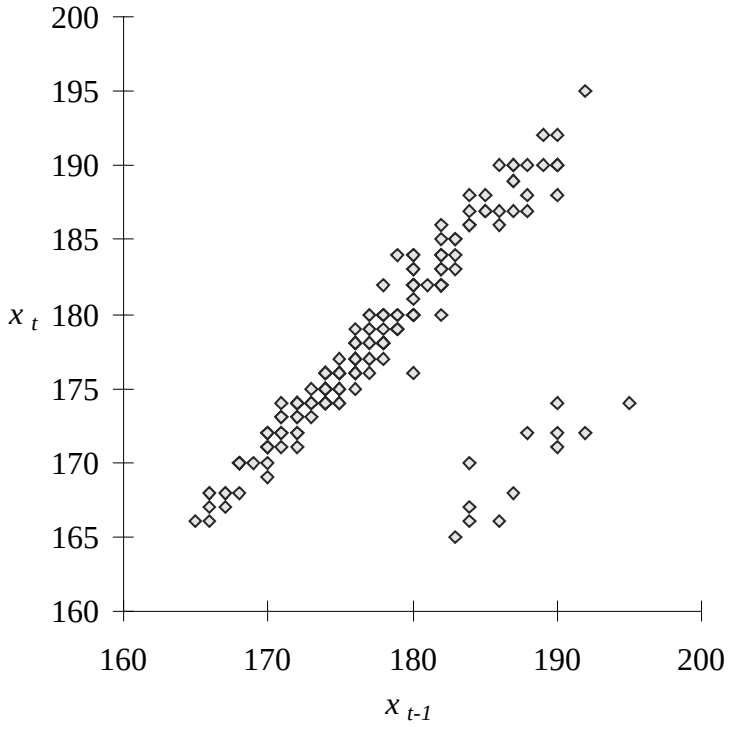

Figure 3.8 – Welding Phase Space

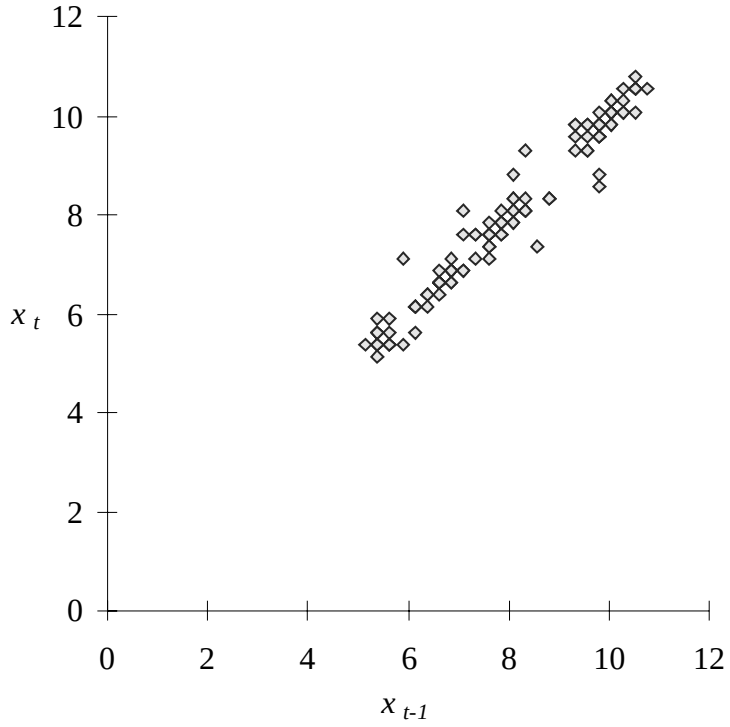

Figure 3.9 – Stock Daily Open Price Phase Space

To determine how well a temporal pattern or a phase space point characterizes an event requires the concept of an event characterization function as introduced in the next section.

3.4 Event Characterization Function

To link a temporal pattern (past and present) with an event (future) the "gold" or event characterization function $g(t)$ is introduced. The event characterization function represents the value of future "eventness" for the current time index. It is, to use an analogy, a measure of how much gold is at the end of the rainbow (temporal pattern). The event characterization function is defined *a priori* and is created to address the specific TSDM goal. The event characterization function is defined such that its value at t correlates highly with the occurrence of an event at some specified time in the future, i.e., the event characterization function is causal when applying the TSDM method to prediction problems. Non-causal event characterization functions are useful when applying the TSDM method to system identification problems.

For the time series illustrated in Figure 3.1, the goal is to predict occurrences of synthetic earthquakes. One possible event characterization function to address this goal is $g(t) = x_{t+1}$, which captures the goal of characterizing synthetic earthquakes one-step in the future.

Alternatively, predicting an event three time-steps ahead requires the event characterization function $g(t) = x_{t+3}$. A more complex event characterization function that would predict an event occurring one, two, or three time-steps ahead is

$$g(t) = \max\{x_{t+1}, x_{t+2}, x_{t+3}\}. \tag{3.5}$$

In Figure 3.2, the TSDM goal is to predict the droplet releases using the stickout time series. Specifically, the objective is to generate one time-step predictions of when metal droplets will release from a welder. In the previous event characterization functions $g(t)$ was defined in terms of x_t – the same time series that contains the temporal patterns. However, in this example, the temporal patterns are discovered in a different time series from the one containing the events. Thus, the event characterization function is $g(t) = y_{t+1}$, where Y is defined by (3.3).

In Figure 3.3, the goal is to decide if the stock should be purchased today and sold tomorrow. The event characterization function that achieves this goal is

$$g(t) = \frac{x_{t+1} - x_t}{x_t}, \tag{3.6}$$

which assigns the percentage change in the stock price for the next day to the current time index.

3.5 Augmented Phase Space

The concept of an augmented phase space follows from the definitions of the event characterization function and the phase space. The augmented phase space is a Q+1 dimensional space formed by extending the phase space with $g(\cdot)$ as the extra dimension. Every augmented phase space point is a vector $< \mathbf{x}_t, g(t) > \in \mathbb{R}^{Q+1}$.

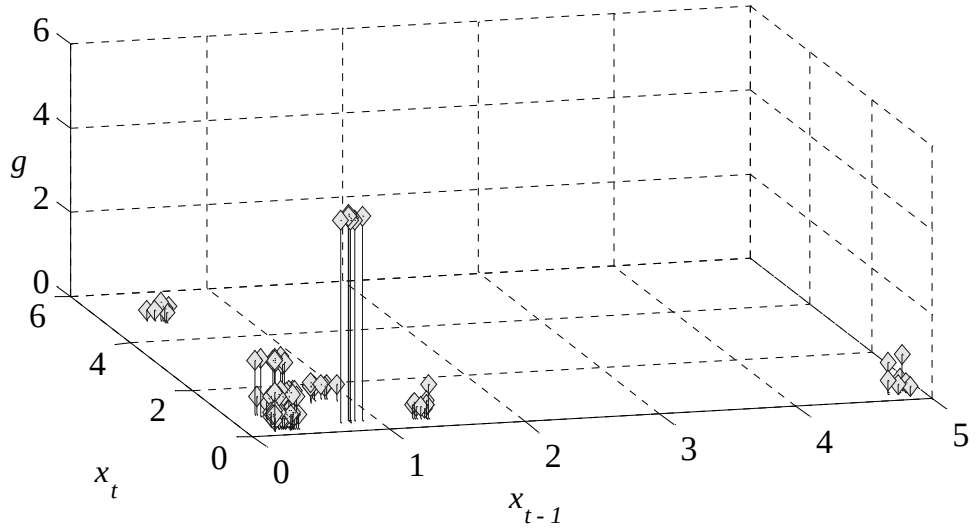

Figure 3.10 – Synthetic Seismic Augmented Phase Space

Figure 3.10, a stem-and-leaf plot, shows the augmented phase space for the synthetic seismic time series. The height of the leaf represents the significance of $g(\cdot)$ for that time index. From this plot, the required temporal pattern and temporal pattern cluster are easily identified.

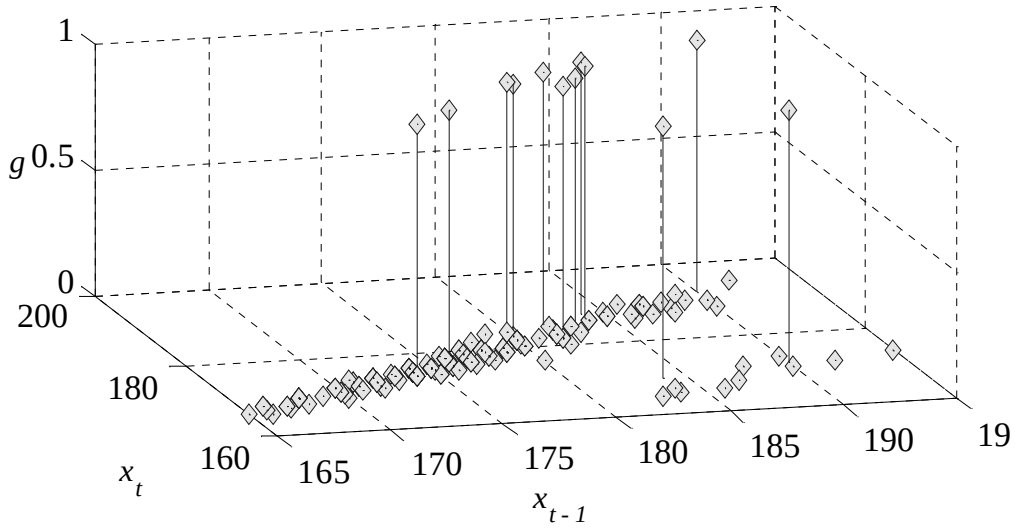

Figure 3.11 – Welding Augmented Phase Space

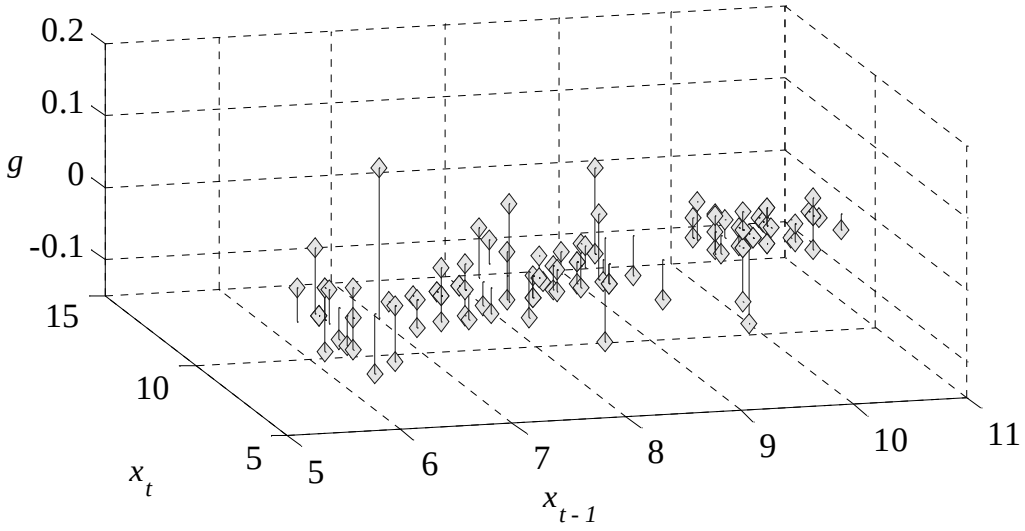

Figure 3.12 – Stock Daily Open Price Augmented Phase Space

Figure 3.11 and 3.12 show the augmented phase spaces for the welding time series and the Stock Daily Open Price, respectively. In both of these plots the desired temporal patterns and temporal pattern clusters are hidden. Appropriate filtering and higher order augmented phase spaces are required to allow the hidden temporal patterns in these time series to be identified. These techniques are discussed in Chapter 6.

Identifying the optimal temporal pattern cluster in the augmented phase space requires the formulation of an objective function, which is discussed in the next section.

3.6 Objective Function

The next concept is the TSDM objective function, which represents the efficacy of a temporal pattern cluster to characterize events. The objective function *f* maps a temporal pattern cluster *P* onto the real line, which provides an ordering to temporal pattern clusters according to their ability to characterize events. The objective function is constructed in such a manner that its optimizer P^* meets the TSDM goal.

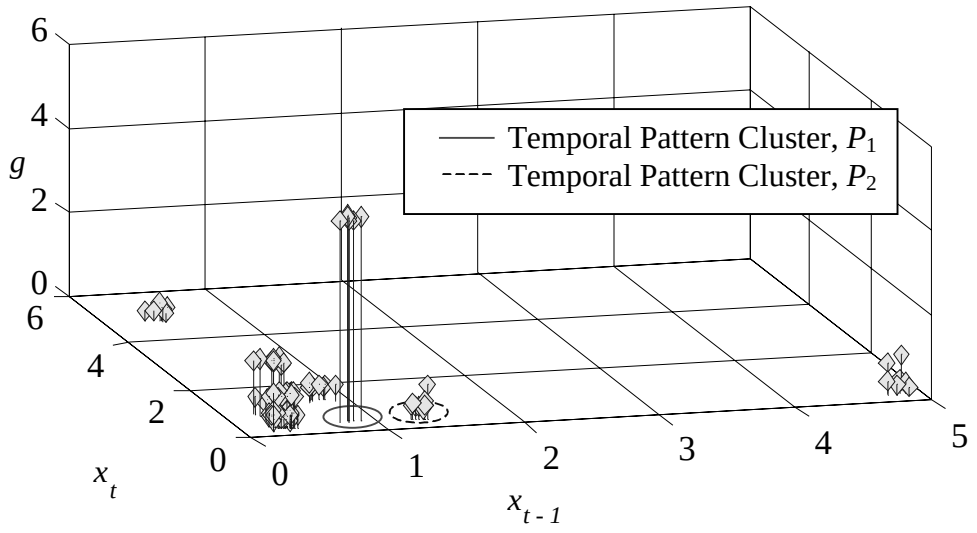

Figure 3.13 – Synthetic Seismic Augmented Phase Space with Highlighted Temporal Pattern Clusters

Figure 3.13 illustrates the requirement of the TSDM objective function. The temporal pattern cluster P_1 is obviously the best temporal pattern cluster for identifying events, while the temporal pattern cluster P_2 is not. The objective function must map the temporal pattern clusters such that $f(P_1) > f(P_2)$.

The form of the objective functions is application dependent, and several different objective functions may achieve the same TSDM goal. Before presenting example objective functions, several definitions are required.

The index set Λ is the set of all time indices *t* of phase space points.

$$\Lambda = \{t : t = (Q-1)\tau + 1, \ldots, N\}, \tag{3.7}$$

where $(Q-1)\tau$ is the largest embedding time-delay, and *N* is the number of observations in the time series. The index set *M* is the set of all time indices *t* when $\mathbf{x}_t$ is within the temporal pattern cluster, i.e.

$$M = \{t : \mathbf{x}_t \in P, t \in \Lambda\}. \tag{3.8}$$

Similarly, $\tilde{M}$, the complement of *M*, is the set of all time indices *t* when $\mathbf{x}_t$ is outside the temporal pattern cluster.

The average value of *g*, also called the average eventness, of the phase space points within the temporal pattern cluster *P* is

$$\mu_M = \frac{1}{c(M)} \sum_{t \in M} g(t), \tag{3.9}$$

where $c(M)$ is the cardinality of *M*. The average eventness of the phase space points not in *P* is

$$\mu_{\tilde{M}} = \frac{1}{c(\tilde{M})} \sum_{t \in \tilde{M}} g(t). \tag{3.10}$$

Consequently, the average eventness of all phase space points is given by

$$\mu_X = \frac{1}{c(\Lambda)} \sum_{t \in \Lambda} g(t). \tag{3.11}$$

The corresponding variances are

$$\sigma_M^2 = \frac{1}{c(M)} \sum_{t \in M} (g(t) - \mu_M)^2, \tag{3.12}$$

$$\sigma_{\tilde{M}}^2 = \frac{1}{c(\tilde{M})} \sum_{t \in \tilde{M}} (g(t) - \mu_{\tilde{M}})^2, \text{ and} \tag{3.13}$$

$$\sigma_X^2 = \frac{1}{c(\Lambda)} \sum_{t \in \Lambda} (g(t) - \mu_X)^2. \tag{3.14}$$

Using these definitions, several examples of objective functions are defined below. The first objective function is the *t* test for the difference between two independent means [53, 54].

$$f(P) = \frac{\mu_M - \mu_{\tilde{M}}}{\sqrt{\frac{\sigma_M^2}{c(M)} + \frac{\sigma_{\tilde{M}}^2}{c(\tilde{M})}}}, \tag{3.15}$$

where *P* is a temporal pattern cluster. This objective function is useful for identifying temporal pattern clusters that are statistically significant and have a high average eventness.

The next example objective function orders temporal pattern clusters according to their ability to characterize time series observations with high eventness and characterize at least a minimum number of events. The objective function

$$f(P) = \begin{cases} \mu_M & \text{if } c(M)/c(\Lambda) \geq \beta \\ (\mu_M - g_0)\dfrac{c(M)}{\beta c(\Lambda)} + g_0 & \text{otherwise} \end{cases}, \tag{3.16}$$

where β is the desired minimum percentage cardinality of the temporal pattern cluster, and g_0 is the minimum eventness of the phase space points, i.e.

$$g_0 = \min\{g(t) : t \in \Lambda\}. \tag{3.17}$$

The parameter β in the linear barrier function in (3.16) is chosen so that $c(M)$ is non-trivial, i.e., the neighborhood around **p** includes some percentage of the total phase space points. If $\beta = 0$, then $c(M) = 0$ or $g(i) = g(j)\ \forall i, j \in M$, i.e., the eventness value of all points in the temporal pattern cluster are identical. If $\beta = 0$, the temporal pattern cluster will be maximal when it contains only one point in the phase space – the point with the highest eventness. If there are many points with the highest eventness, the optimal temporal pattern cluster may contain several of these points. When $\beta = 0$, (3.16) is still defined, because $c(M)/c(\Lambda) \geq 0$ is always true.

The next objective function is useful when the TSDM goal requires that every event is predicted, e.g., the best solution to the welding problem will predict every droplet release. With this goal in mind, the objective function must capture the accuracy with which a temporal pattern cluster predicts all events. Since it may be impossible for a single temporal pattern cluster to perfectly predict all events, a collection $\mathcal{C}$ of temporal pattern clusters is used for this objective function. The objective function $f(\mathcal{C})$ is the ratio of correct predictions to all predictions, i.e.

$$f(\mathcal{C}) = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}, \tag{3.18}$$

where t_p (true positive), t_n (true negative), f_p (false positive), and f_n (false negative) are respectively defined as

$$t_p = c(\{\mathbf{x}_t : \exists P_i \in \mathcal{C} \ni \mathbf{x}_t \in P_i \wedge g(t) = 1\}), \tag{3.19}$$

$$f_p = c(\{\mathbf{x}_t : \exists P_i \in \mathcal{C} \ni \mathbf{x}_t \in P_i \wedge g(t) = 0\}), \tag{3.20}$$

$$t_n = c(\{\mathbf{x}_t : \mathbf{x}_t \notin P_i \, \forall P_i \in \mathcal{C} \wedge g(t) = 0\}), \text{ and} \tag{3.21}$$

$$f_n = c(\{\mathbf{x}_t : \mathbf{x}_t \notin P_i \, \forall P_i \in \mathcal{C} \wedge g(t) = 1\}) \tag{3.22}$$

This objective function would be used to achieve maximum event characterization and prediction accuracy for binary $g(t)$ (1 for an event, 0 for a nonevent) as with the welding time series shown in Figure 3.2.

3.7 Optimization

The key concept of the TSDM framework is to find optimal temporal pattern clusters that characterize and predict events. Thus, an optimization algorithm represented by

$$\max_{\mathbf{p},\delta} f(P) \tag{3.23}$$

is necessary.

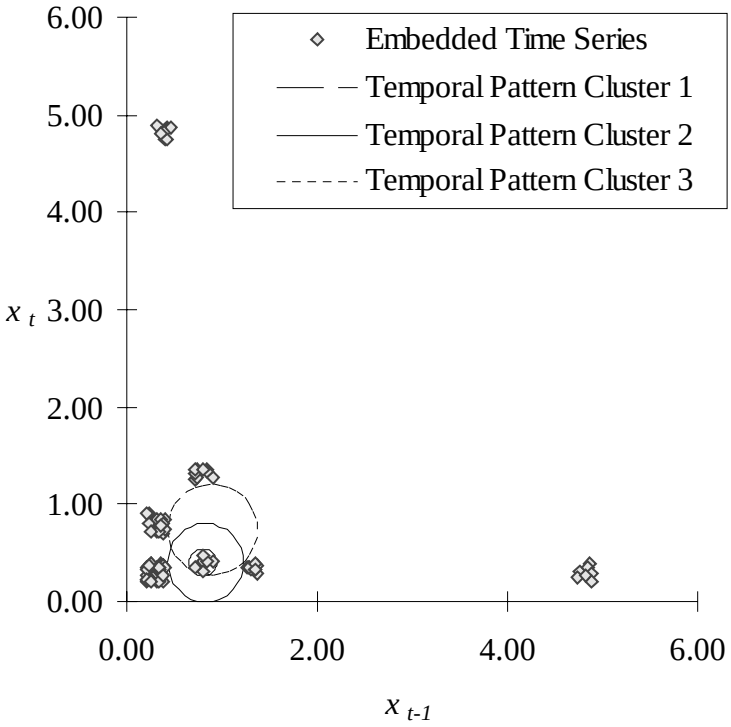

Figure 3.14 – Synthetic Seismic Phase Space with Alternative Temporal Pattern Clusters

Since different temporal pattern clusters may contain the same phase space points, as illustrated in Figure 3.14, a bias may be placed on δ, the radius of the temporal pattern cluster hypersphere. Three possible biases are minimize, maximize, or moderate δ. The choice of the bias is based on the types of prediction errors to be minimized. To minimize the false positive prediction errors, the error of classifying a non-event as an event, δ is minimized subject to $f(P)$ remaining constant. This will cause the temporal pattern cluster to have as small a coverage as possible while not changing the value of the objective function. To minimize the false negative prediction errors, the error of classifying an event as a non-event, δ is maximized subject to $f(P)$ remaining constant. This will cause

the temporal pattern cluster to have as large a coverage as possible while not changing the value of the objective function. A moderating bias would balance between the false positives and false negatives.

Thus, an optimization formulation for (3.15) and (3.16), is $\max f(P)$ subject to $\min \delta$ such that minimizing δ does not change the value of $f(P)$. This formulation places a minimization bias on δ. An optimization formulation for (3.18) is $\max f(\mathcal{C})$ subject to $\min c(\mathcal{C})$ and $\min \delta_i \; \forall P_i \in \mathcal{C}$ such that minimizing $c(\mathcal{C})$ and δ_i does not change the value of $f(P)$. This formulation searches for a minimal set of temporal pattern clusters that is a maximizer of the objective function, and each temporal pattern cluster has a minimal radius.

3.8 Summary of Concepts in Time Series Data Mining

To review, some the key concepts of TSDM follow. An event is defined as an important occurrence in time. The associated event characterization function $g(t)$, defined *a priori*, represents the value of future eventness for the current time index. Defined as a vector of length Q or equivalently as a point in a Q-dimensional space, a temporal pattern is a hidden structure in a time series that is characteristic and predictive of events.

A phase space is a Q-dimensional real metric space into which the time series is embedded. The augmented phase space is defined as a Q+1 dimensional space formed by extending the phase space with the additional dimension of $g(\cdot)$. The objective function represents a value or fitness of a temporal pattern cluster or a collection of temporal pattern clusters. Finding optimal temporal pattern clusters that characterize and predict events is the key of the TSDM framework.

With the concepts of the TSDM framework defined, the next chapter formulates the TSDM method that searches for a single optimal temporal pattern cluster in a single dimensional time series.

Chapter 4 Fundamental Time Series Data Mining Method

This chapter details the fundamental Time Series Data Mining (TSDM) method. After reviewing the problem statement, the TSDM method will be discussed. The chapter presents a method based on an electrical field for moderating the temporal pattern cluster threshold δ. Statistical tests for temporal pattern cluster significance are discussed as a means for validating the results. The chapter also presents an adaptation of a genetic algorithm to the TSDM framework. Extensions and variations of the TSDM method are presented in Chapter 6.

The key to the TSDM method is that it forgoes the need to characterize time series observations at all time indices for the advantages of being able to identify the optimal local temporal pattern clusters for predicting important events. This allows prediction of complex real-world time series using small-dimensional phase spaces.

4.1 Time Series Data Mining Method

The first step in applying the TSDM method is to define the TSDM goal, which is specific to each application, but may be stated generally as follows. Given an observed time series

$$X = \{x_t, t = 1, \ldots, N\}, \tag{4.1}$$

the goal is to find hidden temporal patterns that are characteristic of events in X, where events are specified in the context of the TSDM goal. Likewise, given a testing time series

$$Y = \{x_t, t = R, \ldots, S\} \quad N < R < S, \tag{4.2}$$

the goal is to use the hidden temporal patterns discovered in X to predict events in Y.

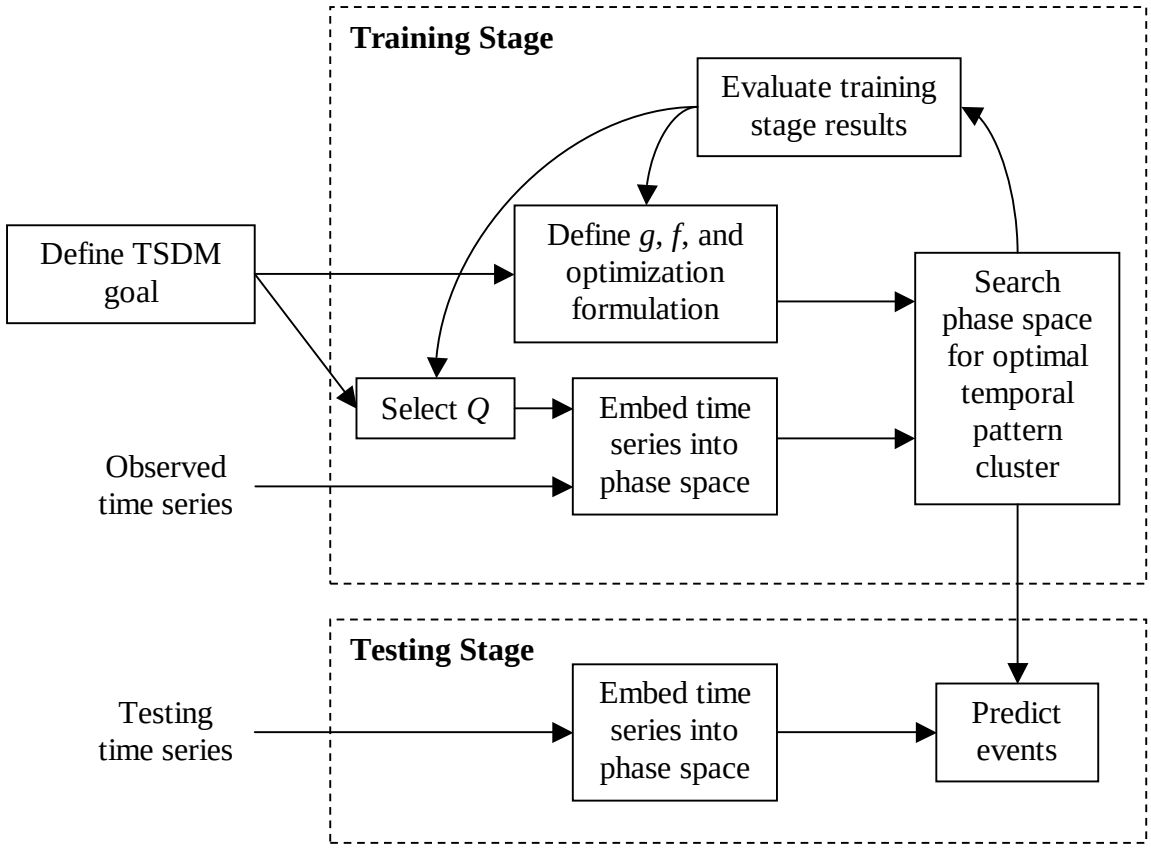

Figure 4.1 – Block Diagram of TSDM Method

Figure 4.1 presents a block diagram of the TSDM method. Given a TSDM goal, an observed time series to be characterized, and a testing time series to be predicted, the steps in the TSDM method are:

I. Training Stage (Batch Process)

1. Frame the TSDM goal in terms of the event characterization function, objective function, and optimization formulation.

   a. Define the event characterization function *g*.

   b. Define the objective function *f*.

c. Define the optimization formulation, including the independent variables over which the value of the objective function will be optimized and the constraints on the objective function.

2. Determine *Q*, i.e., the dimension of the phase space and the length of the temporal pattern.
3. Transform the observed time series into the phase space using the time-delayed embedding process.
4. Associate with each time index in the phase space an eventness represented by the event characterization function. Form the augmented phase space.
5. In the augmented phase space, search for the optimal temporal pattern cluster, which best characterizes the events.
6. Evaluate training stage results. Repeat training stage as necessary.

II. Testing Stage (Real Time or Batch Process)

1. Embed the testing time series into the phase space.
2. Use the optimal temporal pattern cluster for predicting events.
3. Evaluate testing stage results.

With the TSDM method defined, the next section presents an example to further clarify the method's mechanisms.

4.2 TSDM Example

This section applies the TSDM method to the synthetic seismic time series as illustrated in Figure 4.2. The TSDM goal is to characterize and predict the "earthquakes", i.e., the large spikes.

4.2.1 TSDM Training Step 1 – Frame the TSDM Goal in Terms of TSDM Concepts

The first step in the TSDM method is to frame the data mining goal in terms of the event characterization, objective function, and optimization formulation. Since the goal is to characterize the synthetic earthquakes, the event characterization function is $g(t) = x_{t+1}$, which allows prediction one time-step in the future.

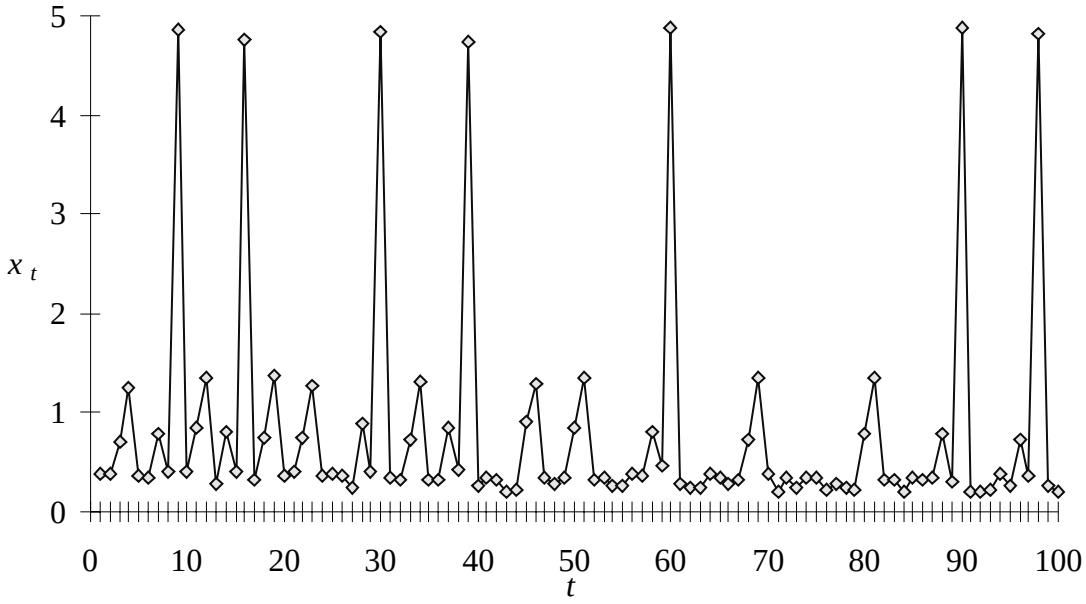

Figure 4.2 – Synthetic Seismic Time Series (Observed)

Since the temporal patterns that characterize the events are to be statistically different from other temporal patterns, the objective function is

$$f(P) = \frac{\mu_M - \mu_{\tilde{M}}}{\sqrt{\frac{\sigma_M^2}{c(M)} + \frac{\sigma_{\tilde{M}}^2}{c(\tilde{M})}}}, \tag{4.3}$$

which orders temporal pattern clusters according to their ability to statistically differentiate between events and non-events.

The optimization formulation is to $\max f(P)$ subject to $\min b(P)$ such that minimizing $b(P)$ does not change the value of $f(P)$. This optimization formulation will identify the most statistically significant temporal pattern cluster with a moderate radius. The function *b* determines a moderate δ based on an electrical field with each phase space point having a unit charge. The function *b* measures the cumulative force applied on the surface of the temporal pattern cluster. The details of *b* are provided later in this chapter.

4.2.2 TSDM Training Step 2 – Determine Temporal Pattern Length

The length of the temporal pattern *Q*, which is also the dimension of the phase space, is chosen ad hoc. Recall that Takens' Theorem proves that if $Q = 2m+1$, where *m* is the original state space dimension, the reconstructed phase space is guaranteed to be topologically equivalent to the original state space, but Takens' Theorem provides no mechanism for determining *m*. Using the principle of parsimony, temporal patterns with small *Q* are examined first. For this example, $Q = 2$, which allows a graphical presentation of the phase space.

4.2.3 TSDM Training Step 3 – Create Phase Space

For this example, Figure 4.3 illustrates the phase space.

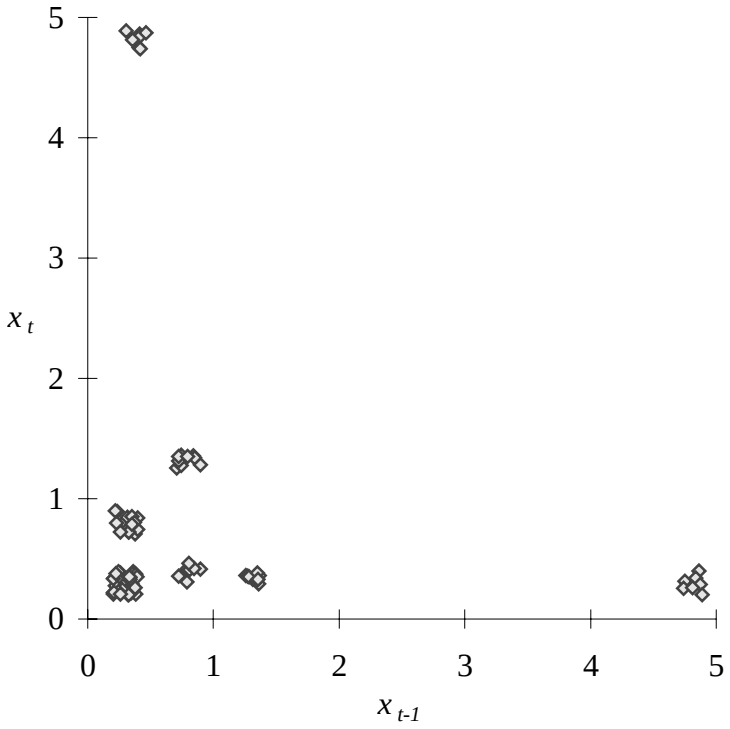

Figure 4.3 – Synthetic Seismic Phase Space (Observed)

The time series X is embedded into the phase space using the time-delay embedding process where each pair of sequential points (x_{t-1}, x_t) in X generates a two-dimensional phase space point. If the phase space were three-dimensional, every triplet of sequential points (x_{t-2}, x_{t-1}, x_t) could be selected to form the phase space. The Manhattan or l_1 distance is chosen as the metric for this phase space.

4.2.4 TSDM Training Step 4 – Form Augmented Phase Space

The next step is to form the augmented phase space by extending the phase space with the $g(\cdot)$ dimension as illustrated by Figure 4.4, a stem-and-leaf plot. The vertical lines represent the dimension g associated with the pairs of (x_{t-1}, x_t). The next step will find an optimal cluster of leaves with high eventness.

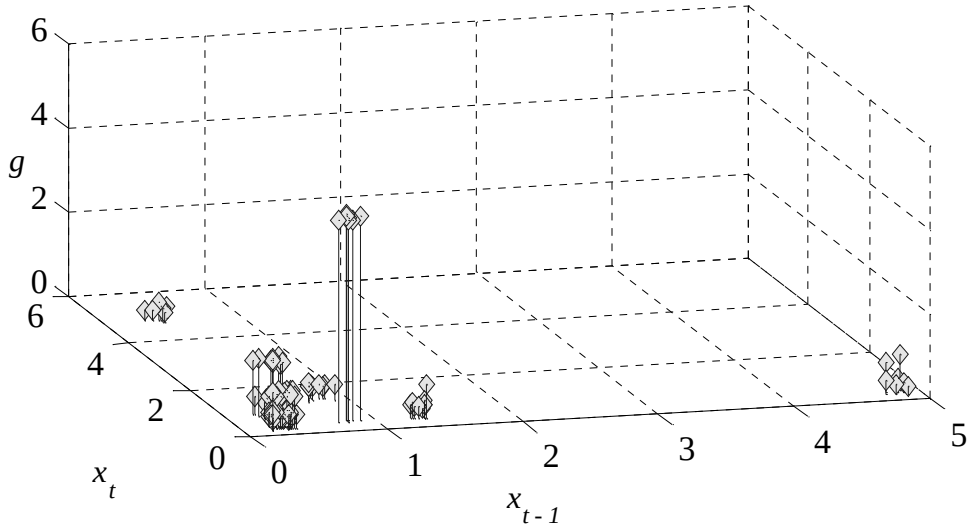

Figure 4.4 – Synthetic Seismic Augmented Phase Space (Observed)

4.2.5 TSDM Training Step 5 – Search for Optimal Temporal Pattern Cluster

A genetic algorithm searches for the optimal temporal pattern cluster, where a temporal pattern cluster P is a hypersphere with a center defined by a temporal pattern $\mathbf{p}$ and a radius δ. In Figure 4.5, the temporal pattern cluster found by the genetic algorithm is highlighted in the phase space. By comparing Figure 4.4 and Figure 4.5, it is obvious that the optimal temporal pattern cluster is identified. The "circle" P (recall the phase space distance is Manhattan) in Figure 4.5 has its center at $\mathbf{p}$ with radius δ.

In Figure 4.6, the temporal pattern and events are highlighted on the time series. The δ is not present in this view, but the relationship between the time series observations matched by the temporal pattern cluster and the event observation is obvious.

It is clear from Figures 4.4, 4.5, and 4.6 that the TSDM training stage has been successful. The process of evaluating the training stage results is explained later in this chapter. Next, the testing stage applies the temporal pattern cluster P to the testing time series.

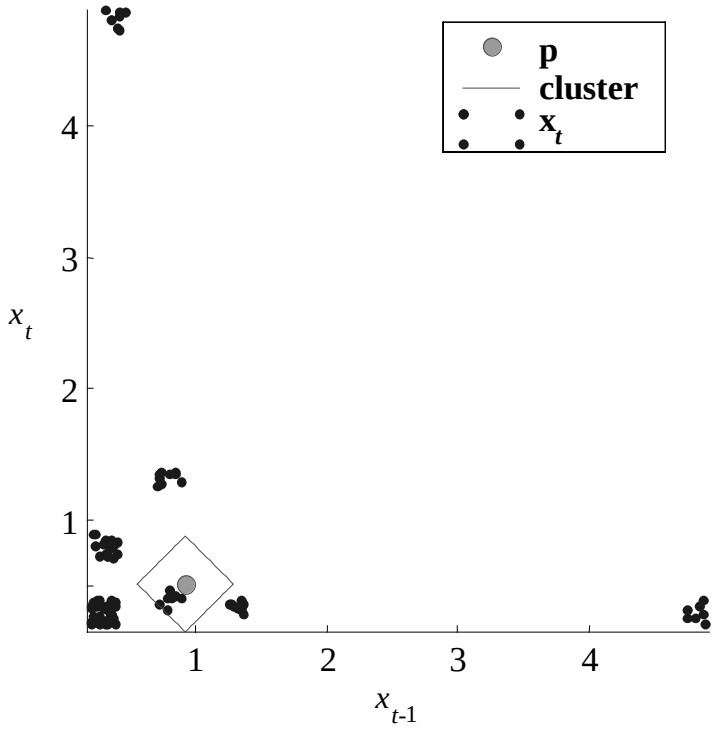

Figure 4.5 – Synthetic Seismic Phase Space with Temporal Pattern Cluster (Observed)

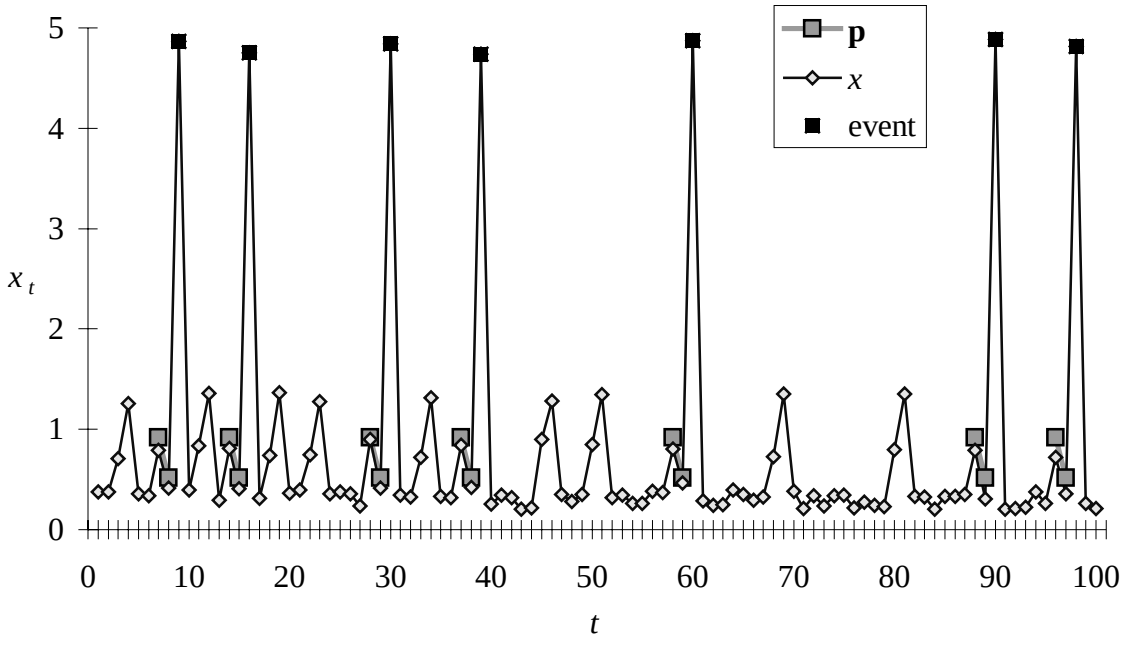

Figure 4.6 – Synthetic Seismic Time Series with Temporal Patterns and Events Highlighted (Observed)

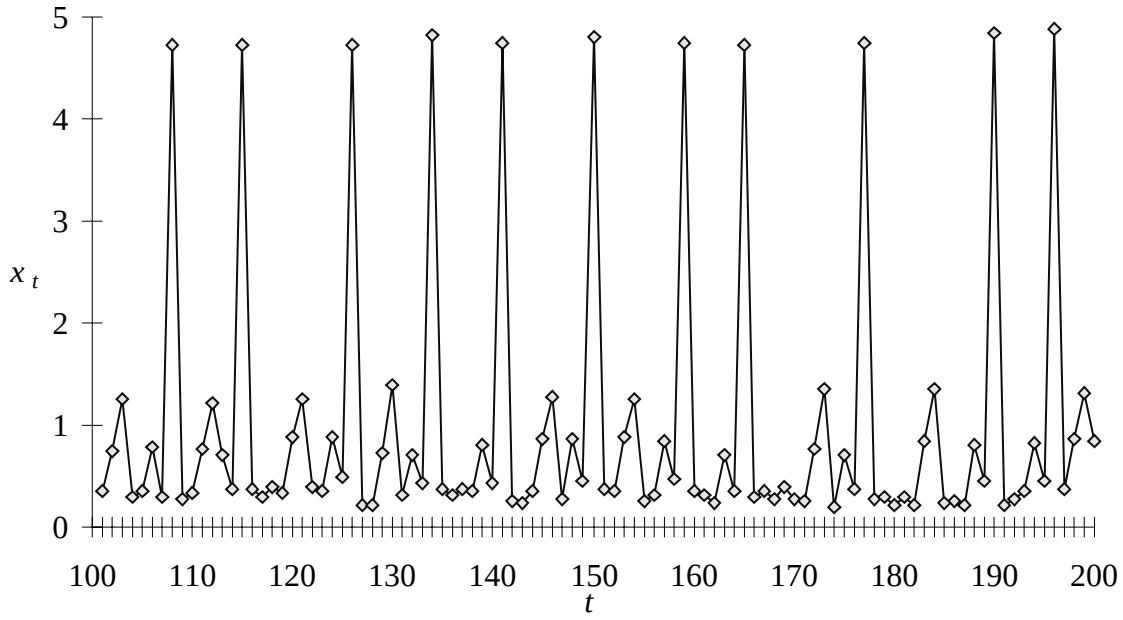

Figure 4.7 – Synthetic Seismic Time Series (Testing)

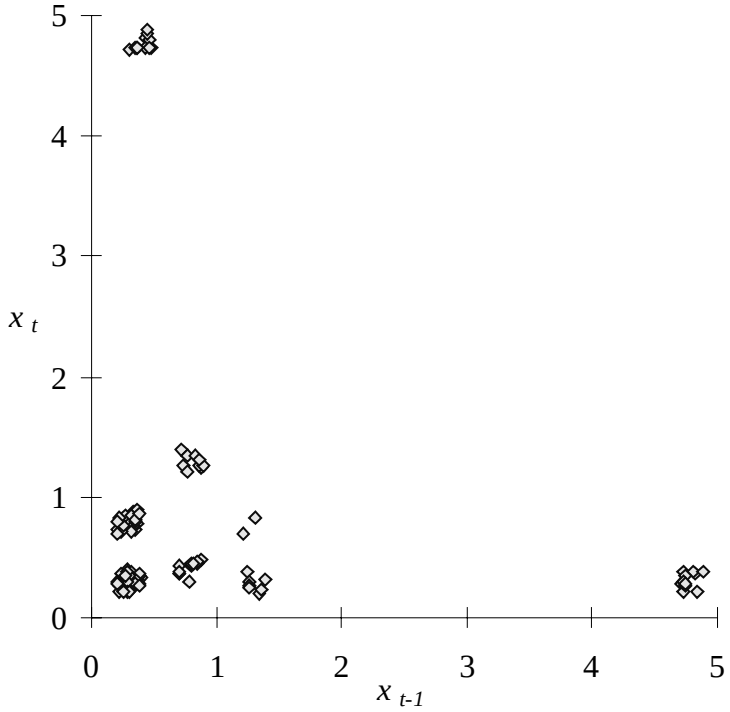

Figure 4.8 – Synthetic Seismic Phase Space (Testing)

4.2.6 TSDM Testing Step 1 – Create Phase Space

The testing time series *Y*, which is shown in Figure 4.7, is the nonstationary, non-periodic continuation of the observed time series. The time series *Y* is embedded into the phase space as shown in Figure 4.8 using the time-delay embedding process performed in the training stage.

4.2.7 TSDM Testing Step 2 – Predict Events

The last step in the TSDM method is to predict events by applying the discovered temporal pattern cluster to the testing phase space. For this example, Figure 4.9 clearly illustrates the accuracy of the temporal pattern in predicting events. The pair of connected gray squares that match sequences of time series observations before events is the temporal pattern. The black squares indicate predicted events.

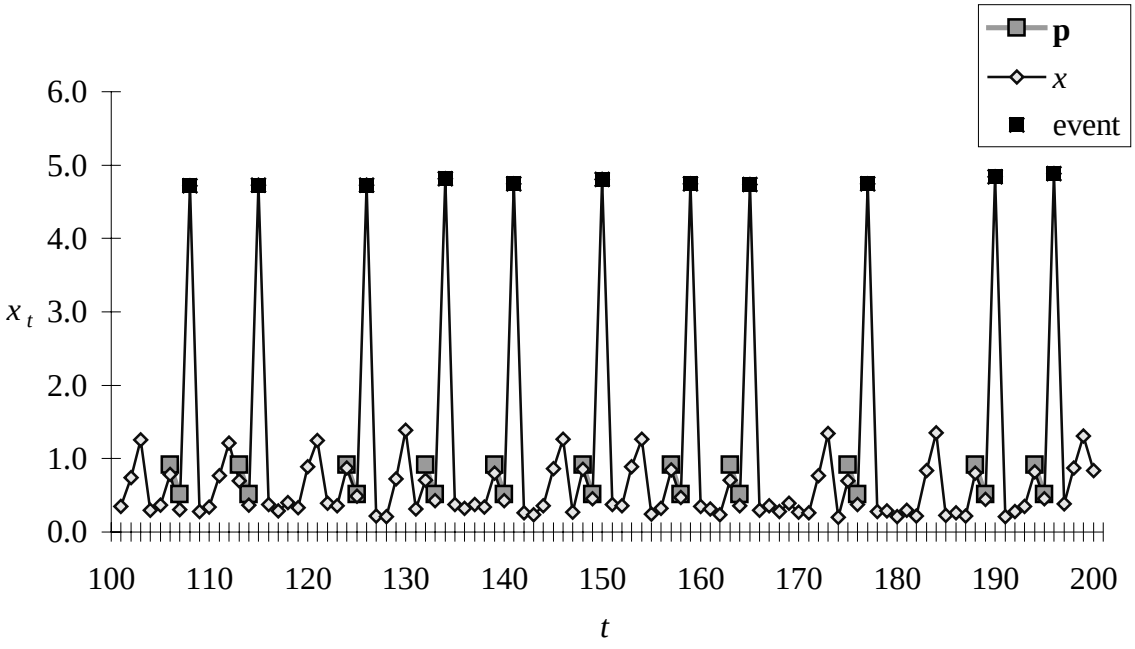

Figure 4.9 – Synthetic Seismic Time Series with Temporal Patterns and Events Highlighted (Testing)

This section has presented an example application of the TSDM method to the synthetic seismic time series. The next section describes in detail the function *b* used in this example to find a moderate δ.

4.3 Repulsion Function for Moderating δ

The optimization formulation in the previous section was to $\max f(P)$ subject to $\min b(P)$ such that minimizing *b*(*P*) does not change the value of *f*(*P*). This section explains the repulsion function *b*, which is based on the concept of an electrical field.

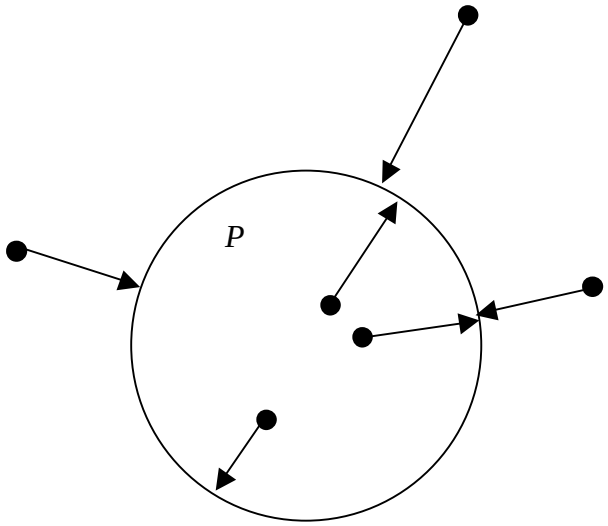

Figure 4.10 – Repulsion Force Illustration

The minimizer of *b* is a temporal pattern cluster with a moderate δ. More precisely, $\delta^*_{\min} \le \delta^*_b \le \delta^*_{\max}$, where $\delta^*_{\min}$ is the radius of $P^*_{\min}$ (the optimal temporal pattern cluster with the smallest radius); δ^*_b is the radius of P^*_b (the optimal temporal pattern cluster with the smallest $b(P)$); and $\delta^*_{\max}$ is the radius of $P^*_{\max}$ (the temporal pattern cluster with the largest radius), where $P^*_{\min}, P^*_b, P^*_{\max} \in \mathcal{C}$, the collection of optimal temporal pattern clusters that all contain the same phase space points. The function *b* represents a repulsion force on the surface of the hypersphere defined by a temporal

pattern cluster *P*. The points in the phase space are treated like fixed electrons that exert a force on the nearest point on the surface of the hypersphere as illustrated in Figure 4.10

Several intermediate results are needed to define *b*. Recall the set of all time indices of phase space points $\Lambda = \{t : t = (Q-1)\tau + 1, \ldots, N\}$. The vector

$$\mathbf{v}_t = \mathbf{x}_t - \mathbf{p},\ t \in \Lambda \tag{4.4}$$

is the vector from the center of the hypersphere to each phase space point. The distances to the surface of the hypersphere are

$$h_t = \left| \delta - \|\mathbf{v}_t\|_p \right|, t \in \Lambda, \tag{4.5}$$

using the *p* norm of the phase space. The

$$m_t = \frac{1}{h_t^p}, t \in \Lambda \tag{4.6}$$

is the force magnitude of the *t*th phase space point. The force

$$\mathbf{f}_t = \begin{cases} m_t \dfrac{\mathbf{v}_t}{\|\mathbf{v}_t\|_p} & \text{if } h_t \le \delta \\ -m_t \dfrac{\mathbf{v}_t}{\|\mathbf{v}_t\|_p} & \text{if } h_t > \delta \end{cases} \qquad t = \tau_{Q\text{-}1}, \ldots, N \tag{4.7}$$

is the *t*th phase space point's force on the hypersphere surface.

Finally,

$$b(P) = \left\| \sum_{t=\tau_{Q-1}}^{N} \mathbf{f}_t \right\|_p + \left| \sum_{t \in M} m_t - \sum_{t \in M} m_t \right| \tag{4.8}$$

is the magnitude of the sum of all forces added to the absolute value of the difference between the sum of the force magnitudes inside the temporal pattern cluster and the sum of the force magnitudes outside the temporal pattern cluster. The minimizer of *b* is both the minimizer of the overall force and the minimizer of the difference between the forces

inside and outside the temporal pattern cluster. The δ_b^* has a value between the $\delta_{\min}^*$ and $\delta_{\max}^*$.

The next section discusses the tests used for evaluating the statistical significance of the temporal pattern clusters.

4.4 Statistical Tests for Temporal Pattern Cluster Significance

Two statistical tests are used to verify that the TSDM goal is met. Recall that the goal was to find hidden temporal patterns that are characteristic of events in the observed time series and predictive of events in the testing time series.

The first statistical test is the runs test. The runs test measures whether a binary sequence is random [54, pp. 135-149]. A binary sequence is formed by assigning a 0 to time series observations classified as non-events and a 1 to those classified as events. Sorting the binary sequence according to associated eventnesses of the binary sequence forms the test sequence. For large sample sizes

$$z = \frac{r - \left[\frac{2n_0 n_1}{n_0 + n_1} + 1\right]}{\sqrt{\frac{2n_0 n_1 (2n_0 n_1 - n_0 - n_1)}{(n_0 + n_1)^2 (n_0 + n_1 - 1)}}}, \tag{4.9}$$

where *r* the number of runs of the same element in a sequence, n_0 is the number of occurrences of a 0, and n_1 is the number of occurrences of a 1.

The test hypothesis is:

H_0: The set of eventnesses associated with the temporal pattern cluster P $\{g(t): t \in M\}$ is not different from the set of eventnesses not associated with the temporal pattern cluster P $\{g(t): t \in \tilde{M}\}$.

H_a: The sets $\{g(t): t \in M\}$ and $\{g(t): t \in \tilde{M}\}$ are different.

The complementary error function and a two-tailed normal distribution are used to find the probability value α associated with *z*. The probability values are typically much better than $\alpha = 0.01$, where α is the probability of making a Type I error. A Type I error is when the null hypothesis is incorrectly rejected [53, pp. 274-276, 54, p. 16].

The second statistical test is the *z* test for two independent samples [53, pp. 336–338, 54, pp. 153–174].

$$z = \frac{\left(\bar{X} - \bar{Y}\right)}{\sqrt{\frac{{\sigma_X}^2}{n_X} + \frac{{\sigma_Y}^2}{n_Y}}}, \tag{4.10}$$

where $\bar{X}$ is the mean of *X*, $\bar{Y}$ is the mean of *Y*, σ_X is the standard deviation of *X*, σ_Y is the standard deviation of *Y*, n_X is the number of elements in *X*, and n_Y is the number of elements in *Y*. As with the runs test, the probability values are typically much better than $\alpha = 0.01$.

The test hypothesis is:

H_0: The mean of the eventnesses $\{g(t): t \in M\}$ associated with the temporal pattern cluster *P* is not greater than the mean of the eventnesses $\{g(t): t \in \tilde{M}\}$ not associated with the temporal pattern cluster *P*.

H_a: The mean of $\{g(t): t \in M\}$ is greater than the mean of $\{g(t): t \in \tilde{M}\}$.

A single-tailed distribution is used. The next section discusses the adaptation of the genetic algorithm for the TSDM method.

4.5 Optimization Method – Genetic Algorithm

In Chapter 2, a review of the basic genetic algorithm was provided. Here the basic genetic algorithm is adapted to the TSDM framework. These adaptations include an initial Monte Carlo search and hashing of fitness values. Additionally, the multi-objective optimization capabilities of the tournament genetic algorithm are discussed.

The basic genetic algorithm presented in Chapter 2 is modified as follows.

Create an elite population

Randomly generate large population (n times normal population size)

Calculate fitness

Select the top 1/n of the population to continue

While all fitnesses have not converged

Selection

Crossover

Mutation

Reinsertion

Initializing the genetic algorithm with the results of a Monte Carlo search has been found to help the optimization's rate of convergence and in finding a good optimum.

The hashing modification reduces the computation time of the genetic algorithm by 50%. This modification is discussed in detail in [20]. Profiling the computation time of the genetic algorithm reveals that most of the computation time is used evaluating the fitness function. Because the diversity of the chromosomes diminishes as the population evolves, the fitness values of the best individuals are frequently recalculated. Efficiently storing fitness values in a hash table dramatically improves genetic algorithm performance [20].

The objective function $\max f(P)$ subject to $\min b(P)$ such that minimizing $b(\delta)$ does not change the value of $f(P)$, presents two separate optimization objectives. The two optimization objectives could be reduced to a single objective problem using a barrier function, or the tournament genetic algorithm could then be applied directly. The second method is applied because the different objectives have different priorities. The primary objective is to maximize $f(P)$. The secondary objective is to minimize $b(P)$ such that minimizing $b(\delta)$ does not change the value of $f(P)$. The primary TSDM goal of finding an optimal temporal pattern cluster should never be compromised to achieve a better temporal pattern cluster shape.

This is accomplished with a tournament tiebreaker system. The chromosomes compete on the primary objective of finding optimal temporal pattern clusters. If, in the

tournament, two chromosomes have the same primary objective function value, the winner is determined by a tiebreaker, where the tiebreaker is the secondary optimization objective.

This chapter presented the TSDM method and through an example showed how hidden temporal patterns can be identified. Additionally, the repulsion force function, statistical characterization of the temporal pattern cluster, and adaptation of the genetic algorithm were discussed. The next chapter further illustrates the method through a series of examples.

Chapter 5 Basic and Explanatory Examples

This chapter presents four examples that help elicit the capabilities and limitations of the TSDM method while clarifying its mechanisms. The first example characterizes the maximal values of a constant frequency sinusoid. The second example applies the TSDM method to a uniform density stochastic time series. The third uses a combination of a sinusoid and uniform density noise to illustrate the TSDM method's capabilities with noisy time series. The fourth example is the synthetic seismic time series.

5.1 Sinusoidal Time Series

The first observed time series, $X = \{x_t = \sin(\omega t), t = 1, \ldots, N\}$, where $\omega = \pi/8$ and $N = 100$, is illustrated in Figure 5.1. For this time series, the TSDM goal is to predict the maximal points of the time series. To achieve this objective, the event characterization function is $g(t) = x_{t+1}$, which will be used for all remaining examples. The objective function (described in Chapter 3) is

$$f(P) = \begin{cases} \mu_M & \text{if } c(M)/c(\Lambda) \geq \beta \\ (\mu_M - g_0)\dfrac{c(M)}{\beta c(\Lambda)} + g_0 & \text{otherwise} \end{cases}, \qquad (5.1)$$

where $\beta = 0.05$. This objective function is useful for finding temporal pattern clusters with a high average eventness, where β is the desired minimum percentage cardinality of the temporal pattern cluster. The optimization formulation is $\max f(P)$ subject to $\min b(\delta)$ such that minimizing $b(\delta)$ does not change the value of $f(P)$. The function b is described in Chapter 4.

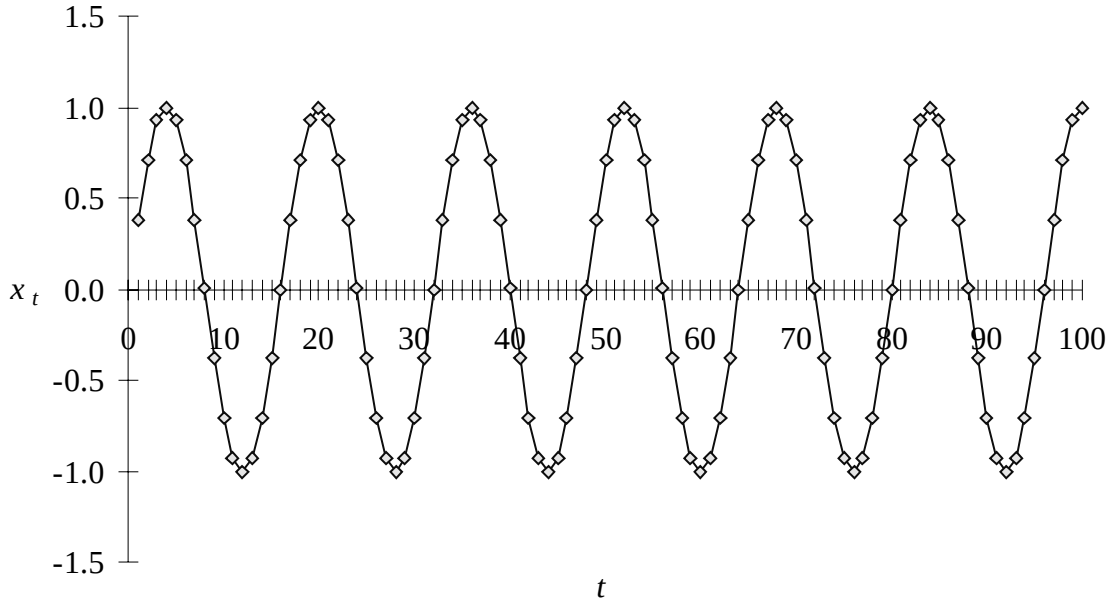

Figure 5.1 – Sinusoidal Time Series (Observed)

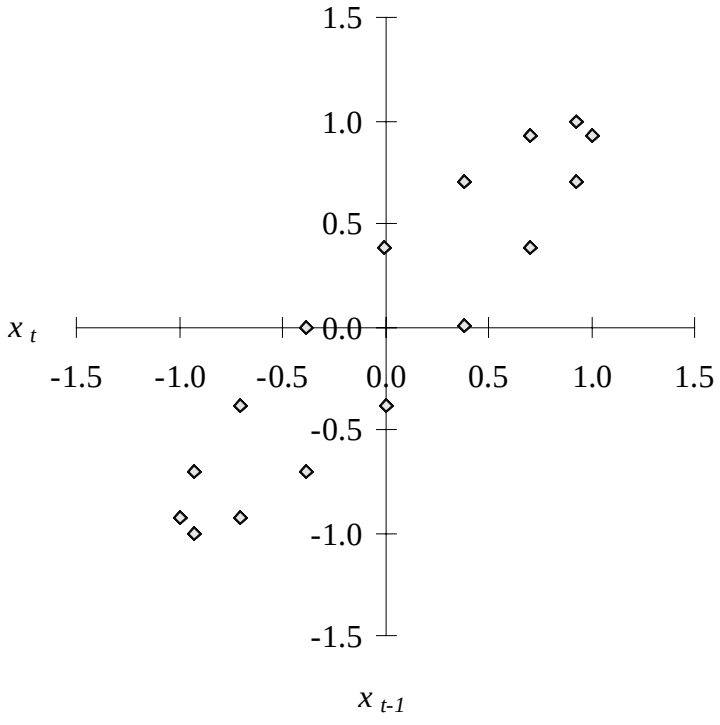

Figure 5.2 – Sinusoidal Phase Space (Observed)

Figure 5.2 presents the training stage phase space with an l_2 distance metric. Since the time series varies sinusoidally, it embeds to an ellipse. Figure 5.3 illustrates the augmented phase space, which further shows the elliptical nature of the phase space points.

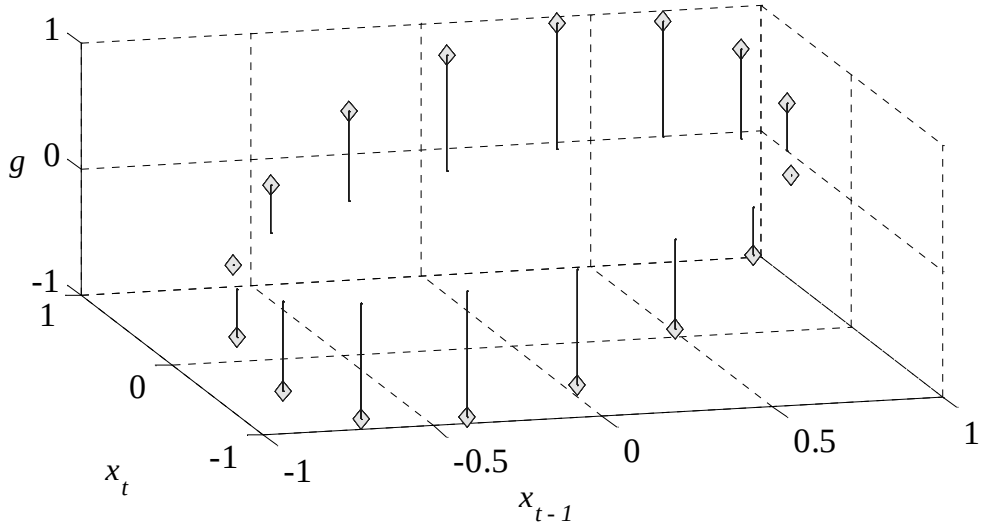

Figure 5.3 – Sinusoidal Augmented Phase Space (Observed)

The tournament genetic algorithm search parameters are presented in Table 5.1. The *random search multiplier* specifies the size of the Monte Carlo search used to create the initial genetic algorithm population. The *population size* is the number of chromosomes in the genetic algorithm population. The *elite count* specifies the number of chromosomes that bypass the selection, mating, and mutation steps. The *gene length* is the number of bits used to represent each dimension of the search space. For a $Q = 2$, the chromosome is formed from three genes. The first gene is the x_{t-1} dimension, the second gene is the x_t dimension, and the third is the threshold δ. Hence, the chromosome will have a length of 3 (genes) x 8 (gene length) = 24 (bits). The *tournament size* specifies the number of chromosomes that will participate in one round of the tournament selection

process. The *mutation rate* specifies the likelihood a particular bit in a chromosome will be mutated. The *convergence* criterion with a range of $[0,1]$ is used to decide when to halt the genetic algorithm. The convergence criterion is the minimum ratio of the worst chromosome's fitness to the best chromosome's fitness. When the ratio is equal to or greater than the convergence criterion, the genetic algorithm is halted.

Parameter	Value
Random search multiplier	1
Population size	100
Elite count	1
Gene length	8
Tournament size	2
Mutation rate	0.2%
Convergence criteria	1

Table 5.1 – Genetic Algorithm Parameters for Sinusoidal Time Series

Result	Value
Temporal pattern, **p**	[0.57 1.0]
Threshold, δ	0.25
Cluster cardinality, $c(M)$	7
Cluster mean eventness, μ_M	1.0
Cluster standard deviation eventness, σ_M	0.0
Non-cluster cardinality, $c(\tilde{M})$	91
Non-cluster mean eventness, $\mu_{\tilde{M}}$	-0.056
Non-cluster standard deviation eventness, $\sigma_{\tilde{M}}$	0.69

Result	Value
z_r	-9.5
α_r	$3.0\text{x}10^{-21}$
z_m	15
α_m	$5.2\text{x}10^{-49}$

Table 5.2 – Sinusoidal Results (Observed)

The search results are shown in Table 5.2. The first two results, *temporal pattern* and *threshold*, define the temporal pattern cluster. The *cluster cardinality* is the count of phase space points in the temporal pattern cluster. The *cluster mean eventness* is the average value of *g* for each phase space point in the cluster. The *cluster standard deviation eventness* is the standard deviation of *g* for the phase space points in the cluster.

The *non-cluster cardinality* is the number of phase space points not in the temporal pattern cluster. The *non-cluster mean eventness* is the average value of *g* for each phase space point not in the temporal pattern cluster. The *non-cluster standard deviation eventness* is the standard deviation of *g* for the phase space points not in the temporal pattern cluster.

The last four results describe the statistical significance of the temporal pattern cluster using the runs test and the *z* test for two independent samples, which were discussed in Chapter 4. The runs test uses a 0.01 probability of Type I error ($\alpha = 0.01$). The $\alpha_r = 3.0\text{x}10^{-21} < 0.01$ means the null hypothesis can be rejected for the observed time series results.

The second statistical test is the *z* test for two independent samples. The two populations are the eventness of the points in the temporal pattern cluster and the

eventness of the points not in the temporal pattern cluster. The *z* test uses a 0.01 probability of Type I error ($\alpha = 0.01$). Again, $\alpha_m = 5.2\text{x}10^{-49} < 0.01$ shows that the null hypothesis can be rejected for the observed time series temporal pattern cluster.

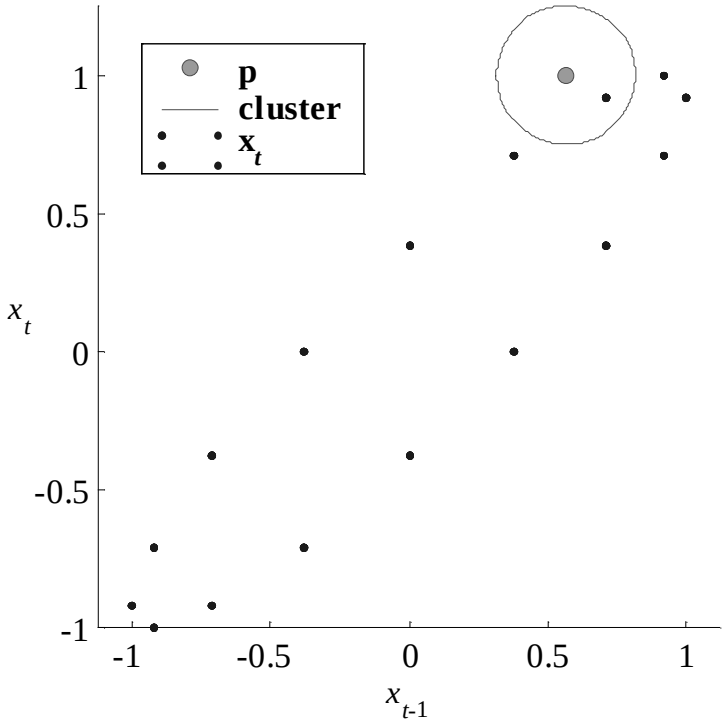

Figure 5.4 – Sinusoidal Phase Space with Temporal Pattern Cluster (Observed)

Figure 5.4 highlights the temporal pattern $\mathbf{p} = [0.57\ 1.0]$ with threshold $\delta = 0.25$ in the phase space. By comparing the temporal pattern cluster seen in Figure 5.4 to the augmented phase space in Figure 5.3, it is obvious that the best temporal pattern cluster is identified. When the temporal pattern cluster matches a subsequence of the time series, the next time series observation is a maximal value of the sinusoid.

In the testing stage, the temporal pattern cluster is used to predict events. The testing stage time series $Y = \{x_t = \sin(\omega t), t = S, \ldots, R\}$, where $\omega = \pi/8$, $S = 101$, and $R = 200$, is shown in Figure 5.5.

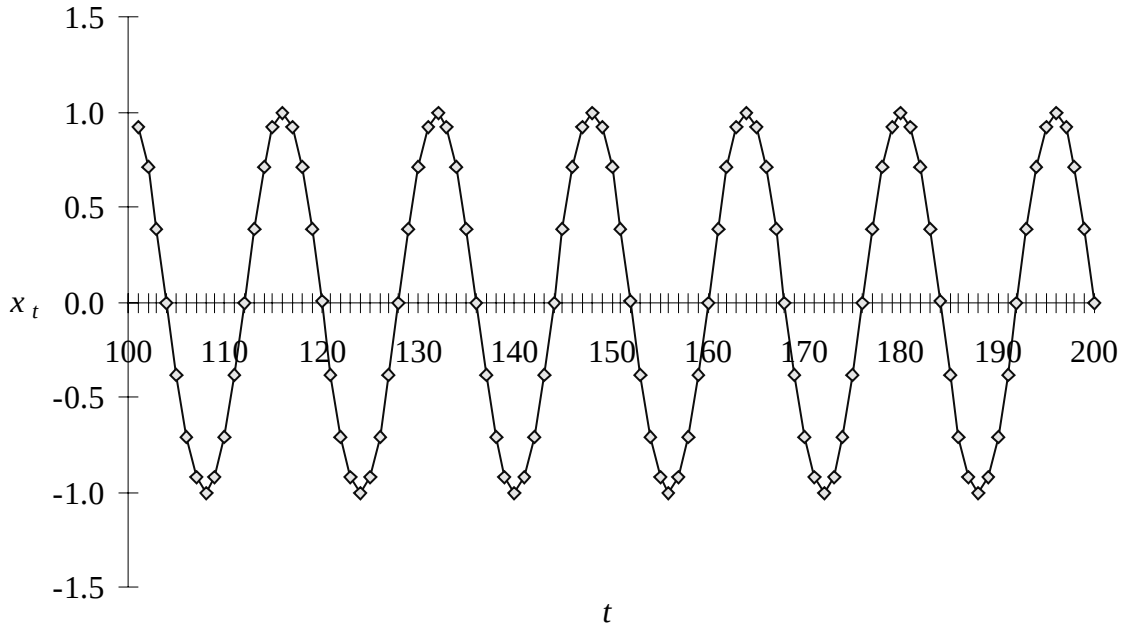

Figure 5.5 – Sinusoidal Time Series (Testing)

Since the testing time series is identical to the observed time series except for a time shift, the phase space and augmented phase spaces are identical to Figure 5.2 and Figure 5.3, respectively.

Result	Value
Cluster cardinality, $c(M)$	6
Cluster mean eventness, μ_M	1.0
Cluster standard deviation eventness, σ_M	0.0
Non-cluster cardinality, $c(\tilde{M})$	92
Non-cluster mean eventness, $\mu_{\tilde{M}}$	-0.061
Non-cluster standard deviation eventness, $\sigma_{\tilde{M}}$	0.68
z_r	-9.4
α_r	5.4×10^{-21}

Result	Value
z_m	15
α_m	2.0×10^{-51}

Table 5.3 – Sinusoidal Results (Testing)

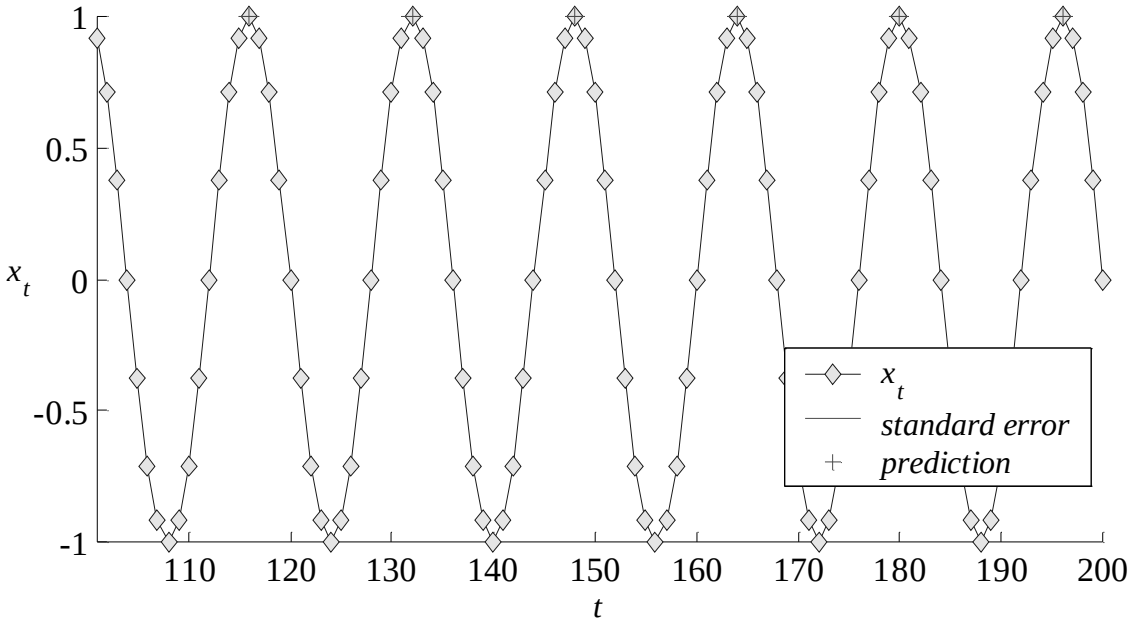

Figure 5.6 – Sinusoidal Time Series with Predictions (Testing)

The testing stage demonstrates that the TSDM goal of predicting all maximal values in the sinusoid is met, as illustrated in Table 5.3 and Figure 5.6. The patterns discovered in the training phase and applied in the testing phase are statistically significant according to the α_r and α_m statistics. The null hypothesis can be rejected in both cases.

The data mining nature of the TSDM method is clearly demonstrated by this example. The temporal pattern cluster characterizes the sequences that lead to the

observations with the highest eventness. The next example applies the TSDM method to a noise time series.

5.2 Noise Time Series

A random variable **x** with a uniform density function generates the second example time series, where

$$f(x) = \begin{cases} 1 & 0 \le x \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

is the density function [55, p. 75]. The time series $X = \{x_t = \mathbf{x}(t), t = 1, \ldots, 100\}$ is illustrated in Figure 5.7.

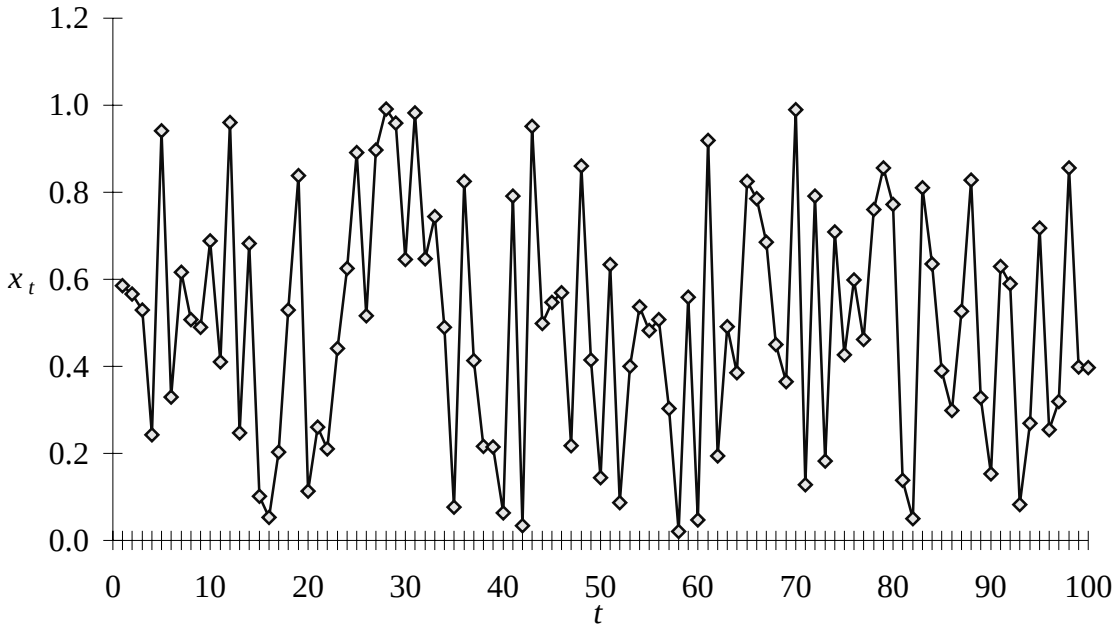

Figure 5.7 – Noise Time Series (Observed)

For this time series, the TSDM goal is to find a temporal pattern that is characteristic and predictive of time series observations that have high values. Because

the time series is a random sequence, the expectation is that any temporal pattern cluster discovered in the training phase will not be predictive in the testing phase.

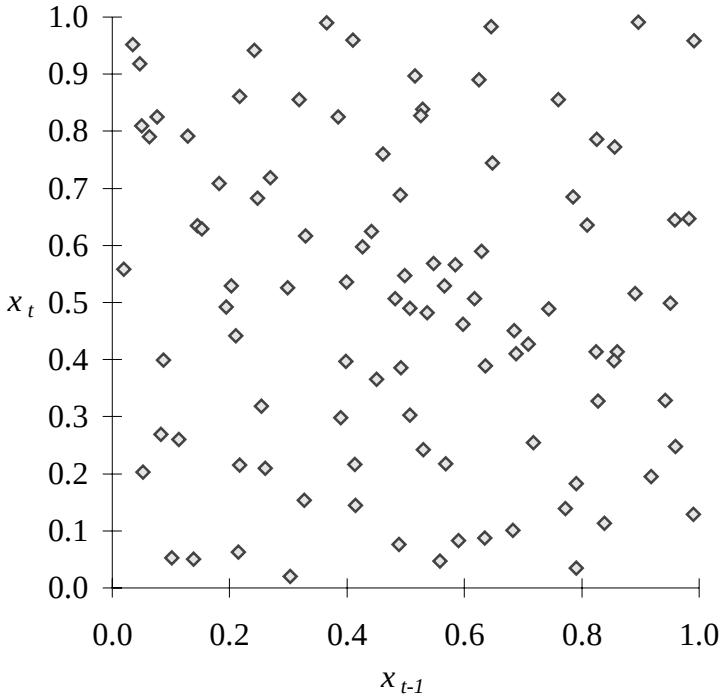

Figure 5.8 – Noise Phase Space (Observed)

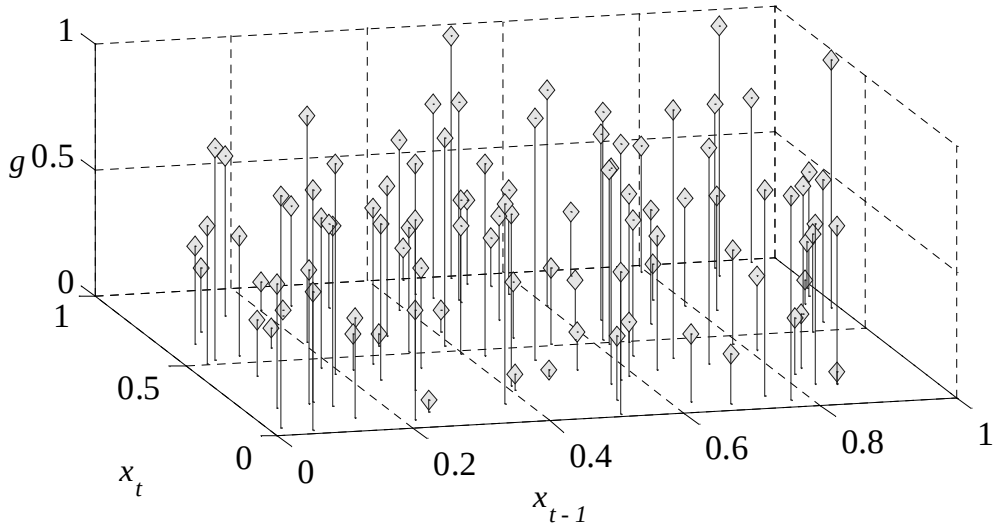

Figure 5.9 – Noise Augmented Phase Space (Observed)

The event characterization, objective function, and optimization formulation are the same as in the previous section. Figure 5.8 presents the Euclidean phase space. Since the time series varies randomly in a uniform manner over the range [0,1], it embeds to an evenly scattered pattern. Figure 5.9 shows the augmented phase space, which further illustrates the scattered nature of the embedded time series.

The search parameters are described previously in Table 5.1. The training stage results are shown in Table 5.4.

Result	Value
Temporal pattern, **p**	[0.72 0.97]
Threshold, δ	0.21
Cluster cardinality, $c(M)$	5
Cluster mean eventness, μ_M	0.78
Cluster standard deviation eventness, σ_M	0.20
Non-cluster cardinality, $c(\tilde{M})$	93
Non-cluster mean eventness, $\mu_{\tilde{M}}$	0.48
Non-cluster standard deviation eventness, $\sigma_{\tilde{M}}$	0.28
z_r	-0.54
α_r	5.9×10^{-1}
z_m	3.1
α_m	8.2×10^{-4}

Table 5.4 – Noise Results (Observed)

Finding a statistically significant temporal pattern in random noise is counterintuitive. However, the TSDM method found a temporal pattern cluster containing

five phase space points with a mean eventness greater than the mean eventness of phase space points not contained in the temporal pattern cluster. According to $\alpha_m = 8.2\text{x}10^{-4}$, the null hypothesis may be rejected, i.e., the two sets are statistically different. However, according to the runs statistical test $\alpha_r = 5.9\text{x}10^{-1}$, the two sets cannot be said to be statistically different. This means that there is some evidence that the temporal pattern is statistically significant. Figure 5.10 highlights the temporal pattern $\mathbf{p} = [0.72\ 0.97]$ with threshold $\delta = 0.21$ illustrated in the phase space.

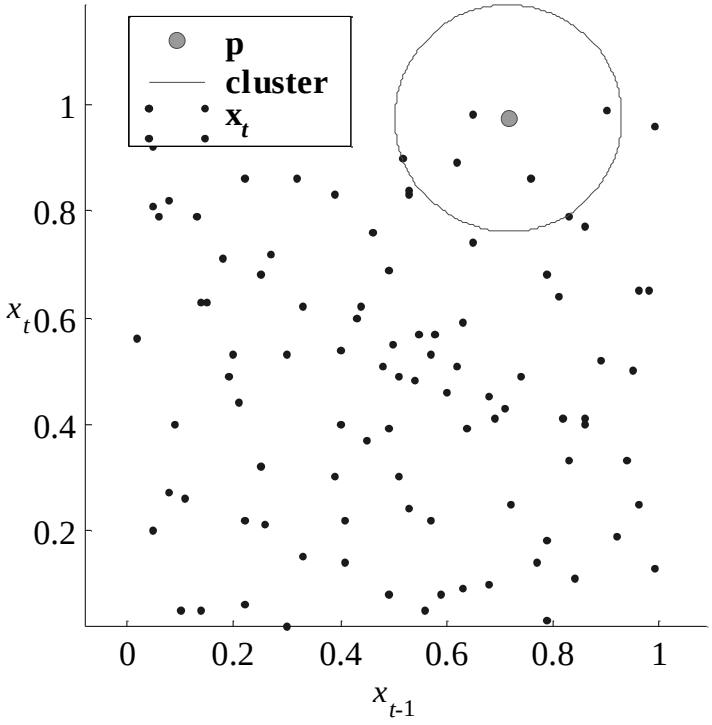

Figure 5.10 – Noise Phase Space with Temporal Pattern Cluster (Observed)

The testing stage time series $X = \{x_t = \mathbf{x}(t), t = 101, \ldots, 200\}$, which is a continuation of the training stage time series, is illustrated in Figure 5.11. The testing time series is transformed into the phase space as shown in Figure 5.12, and the augmented phase space is seen in Figure 5.13.

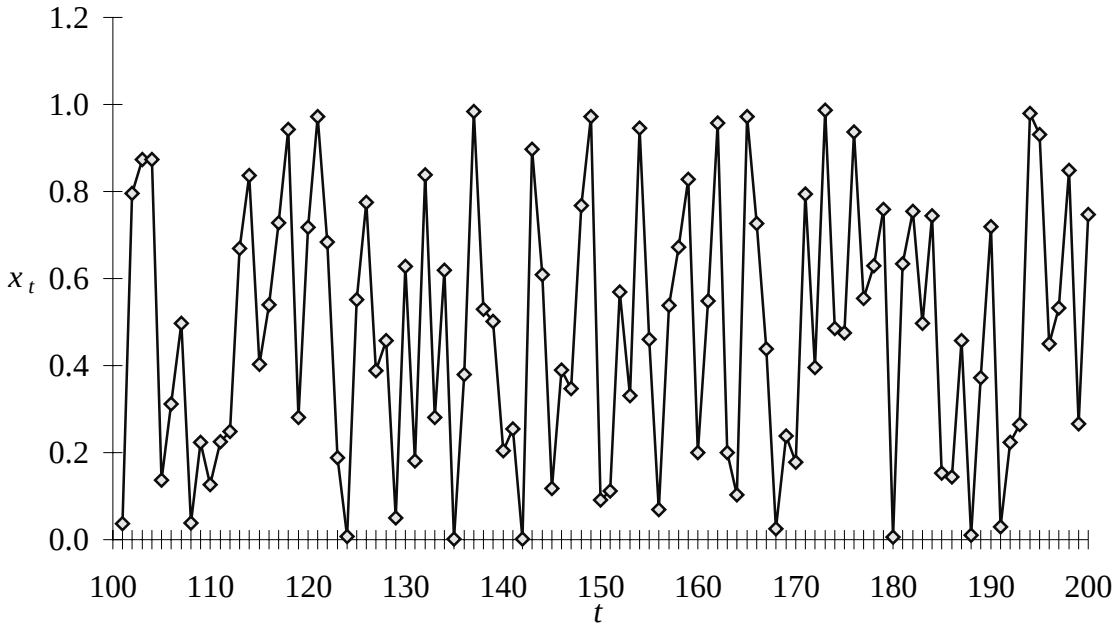

Figure 5.11 – Noise Time Series (Testing)

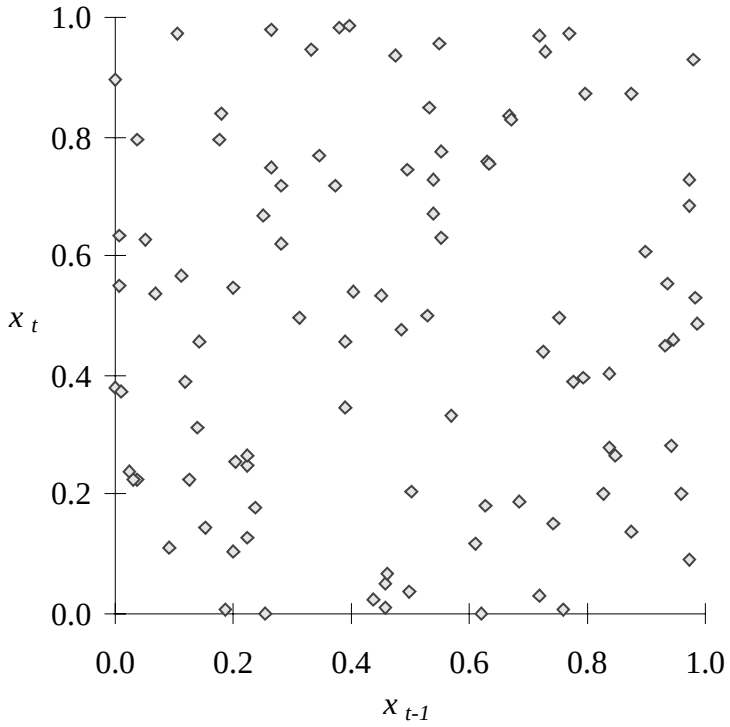

Figure 5.12 – Noise Phase Space (Testing)

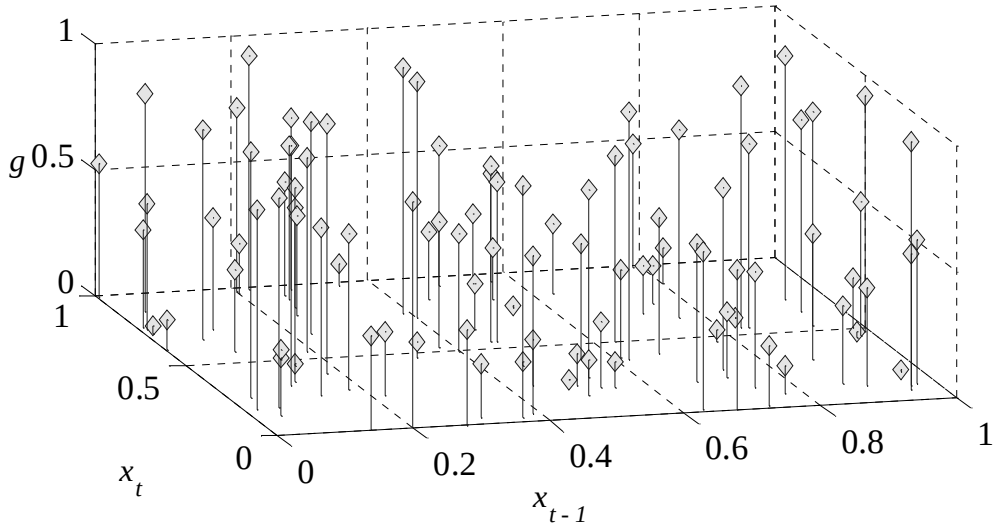

Figure 5.13 – Noise Augmented Phase Space (Testing)

Table 5.5 shows the statistical characterization of the testing stage results.

Result	Value
Cluster cardinality, $c(M)$	8
Cluster mean eventness, μ_M	0.36
Cluster standard deviation eventness, σ_M	0.28
Non-cluster cardinality, $c(\tilde{M})$	90
Non-cluster mean eventness, $\mu_{\tilde{M}}$	0.49
Non-cluster standard deviation eventness, $\sigma_{\tilde{M}}$	0.30
z_r	-0.48
α_r	6.3×10^{-1}
z_m	-1.3
α_m	9.1×10^{-1}

Table 5.5 – Noise Results (Testing)

The temporal pattern cluster discovered in the training stage and applied in the testing stage is not statistically significant as seen by the α_r and α_m statistics. The null hypothesis cannot be rejected. This is illustrated in Figure 5.14, which shows the predictions made by the testing stage.

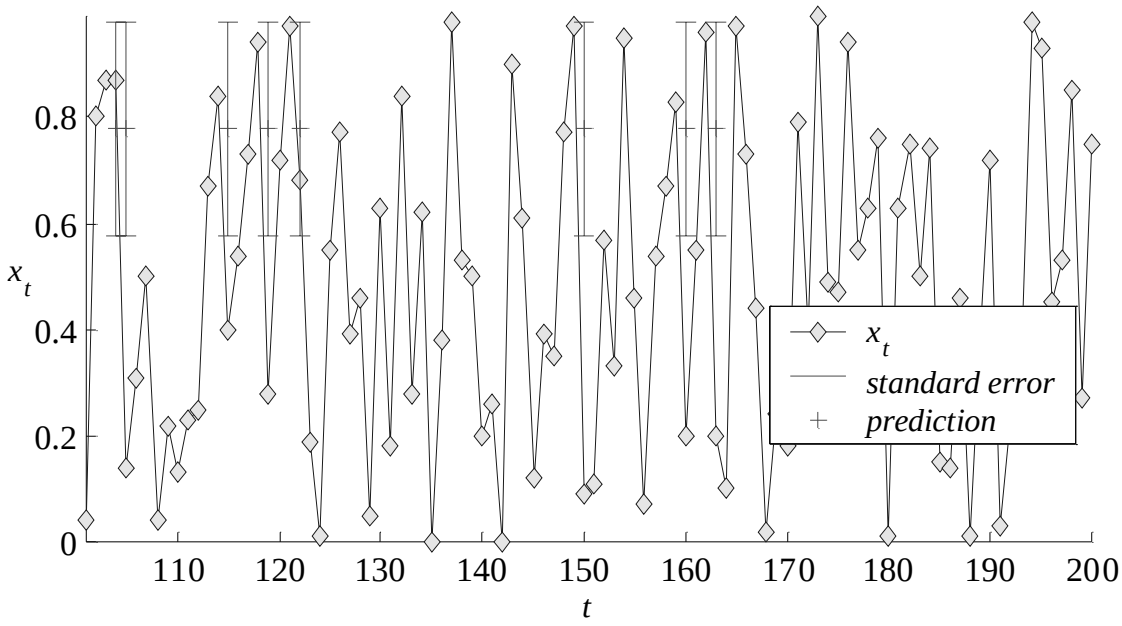

Figure 5.14 – Noise Time Series with Predictions (Testing)

In this example, the TSDM method cannot find temporal pattern clusters that are both characteristic and predictive of events in a noise time series. Figure 5.14 along with the results from Table 5.5, show that the TSDM goal of finding a temporal pattern cluster that is predictive of time series observations whose mean value is greater than the mean value of the not predicted observations has not been met.

Although according to one statistical measure, the training stage results were significant in their ability to characterize events, these results did not carry over to

predicting events in the testing stage. However, the next section shows that a sinusoidal contaminated with noise is still predictable.

5.3 Sinusoidal with Noise Time Series

A sinusoid combined with a random variable **x** (5.2) is illustrated by Figure 5.15, where, $X = \{x_t = \sin(t\pi/8) + 1/5\,\mathbf{x}(t), t = 1,\ldots,100\}$.

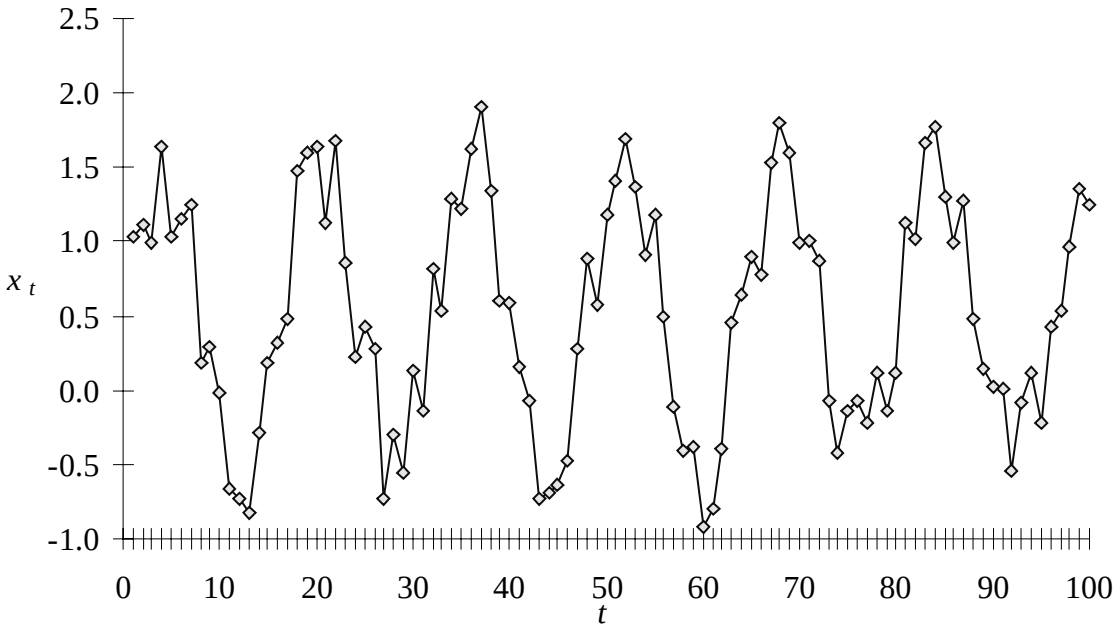

Figure 5.15 - Sinusoidal with Noise Time Series (Observed)

To further characterize this time series, the signal-to-noise-ratio (SNR) is measured and determined analytically. The theoretical SNR is the ratio of the signal variance to the noise variance. This would be the measured SNR for an ergodic time series as the length of the time series approached infinity. The variance of the random variable [55, p. 107] **x** is

$$\frac{1}{5}\cdot\frac{1}{2}\int_0^1 x^2\,dx = \frac{1}{15}. \tag{5.3}$$

The variance of the sinusoid is

$$\frac{1}{16}\int_0^{16}\sin^2(t\pi/8)\,dt - 0 = \frac{1}{2\pi}\left(\frac{t\pi}{16} - \frac{1}{4}\sin\left(\frac{t\pi}{4}\right)\right)\Bigg|_0^{16} = \frac{1}{2}, \tag{5.4}$$

making the theoretical SNR 7.5 (8.8dB). The measured variance of the noise is 0.069 and of the sinusoid is 0.51, making the measured SNR 7.4 (8.7dB) for the finite length observed time series.

For this time series, the TSDM goal is to predict the maximal values of the time series. The objective function, event characterization function, and optimization formulation remain the same as in the two previous sections.

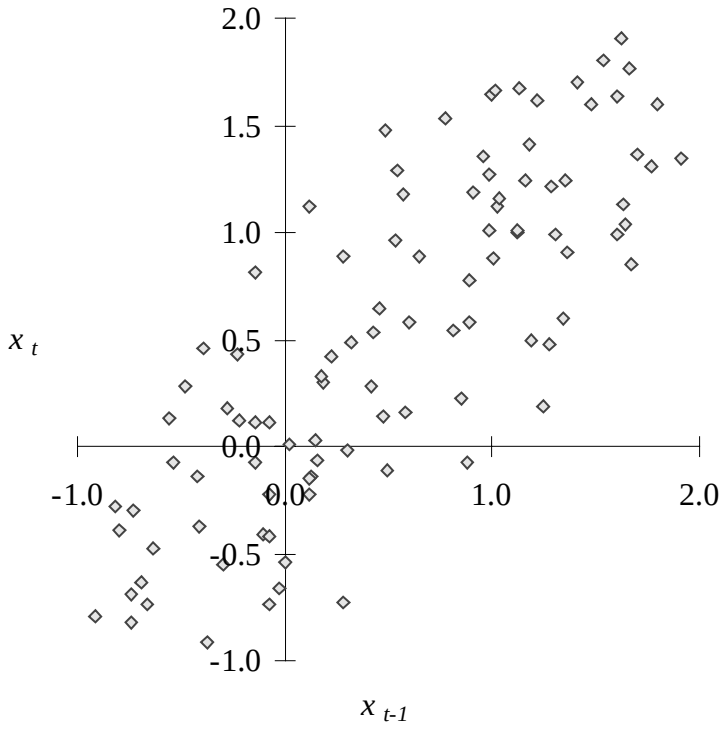

Figure 5.16 - Sinusoidal with Noise Phase Space (Observed)

Figure 5.16 presents the Euclidean phase space. Since the time series is composed of a sinusoid and a uniform density random variable, the embedding is expected to be a

scattered ellipse. Figure 5.16 shows exactly this type of pattern. Figure 5.17 shows the augmented phase space, which further illustrates the scattered elliptical nature of the embedded time series.

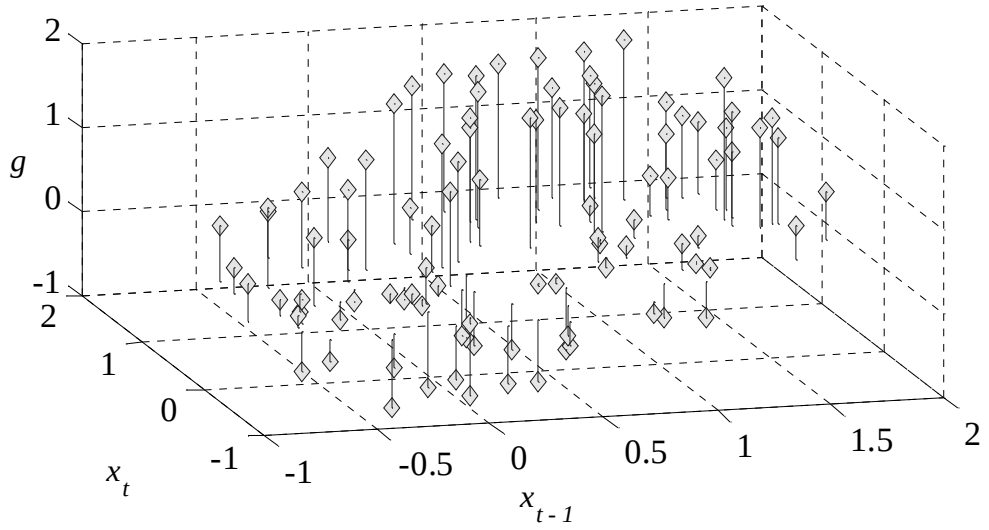

Figure 5.17 - Sinusoidal with Noise Augmented Phase Space (Observed)

The genetic algorithm search parameters are described previously in Table 5.1. The training stage results are shown in Table 5.6.

Result	Value
Temporal pattern	[1.1 1.8]
Threshold	0.46
Cluster cardinality	9
Cluster mean eventness	1.5
Cluster standard deviation eventness	0.36
Not cluster cardinality	89
Not cluster mean eventness	0.41
Not cluster standard deviation eventness	0.72

Result	Value
z_r	-3.3
α_r	8.8×10^{-4}
z_m	7.7
α_m	5.1×10^{-15}

Table 5.6 - Sinusoidal with Noise Results (Observed)

According to both statistical tests, the training results are statistically significant. Figure 5.18 highlights the temporal pattern $\mathbf{p} = [1.1\ 1.8]$ with threshold $\delta = 0.46$ in the phase space. Comparing the temporal pattern cluster seen in Figure 5.18 to the augmented phase space in Figure 5.17 demonstrates that the TSDM method found a good temporal pattern cluster.

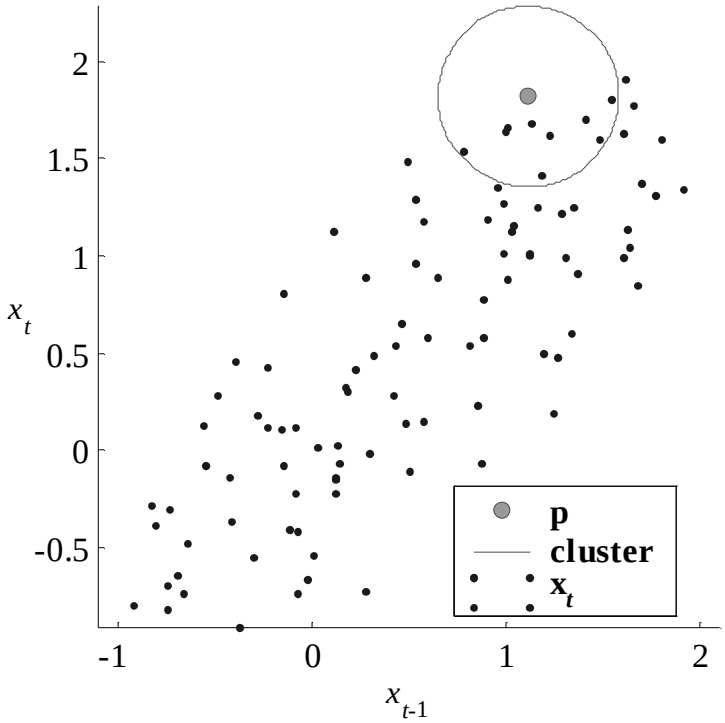

Figure 5.18 - Sinusoidal with Noise Phase Space with Temporal Pattern Cluster (Observed)

Figure 5.19 illustrates the testing stage time series, which is a continuation of the observed time series. The measured variance of the noise is 0.084 and of the sinusoid is 0.50, yielding a measured SNR is 6.0 (7.8dB). Figure 5.20 and Figure 5.21 illustrate the phase space and the augmented phase space, respectively.

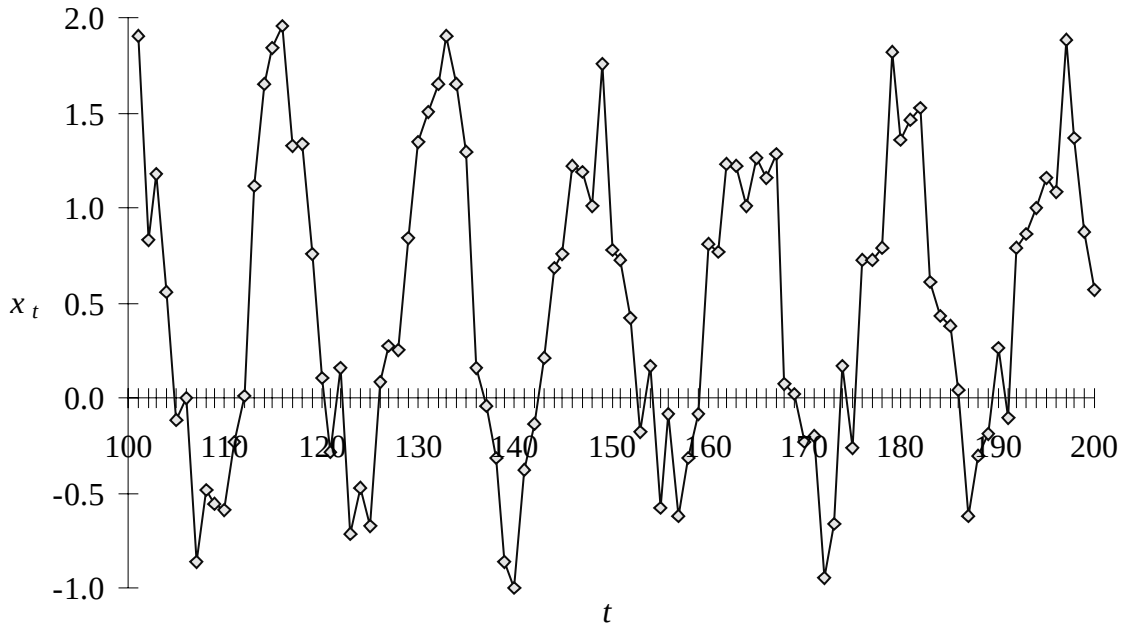

Figure 5.19 - Sinusoidal with Noise Time Series (Testing)

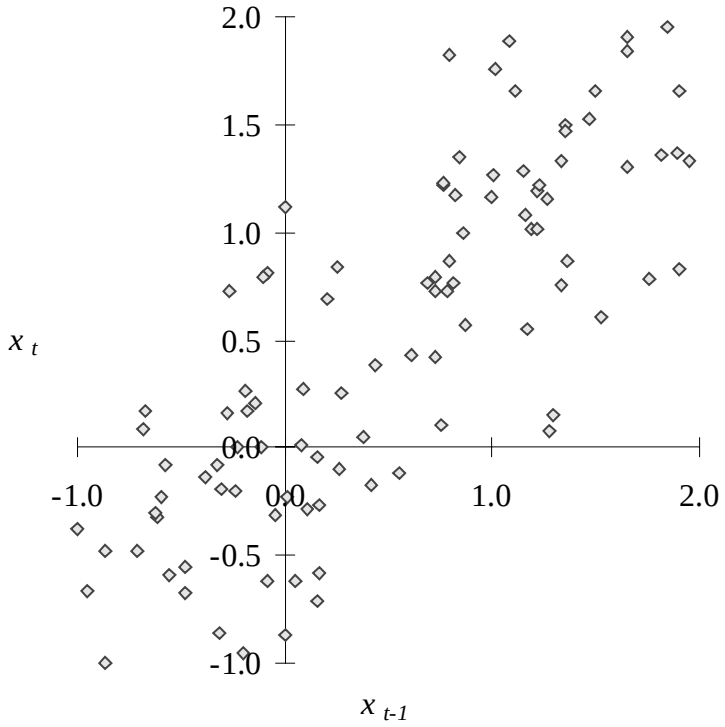

Figure 5.20 - Sinusoidal with Noise Phase Space (Testing)

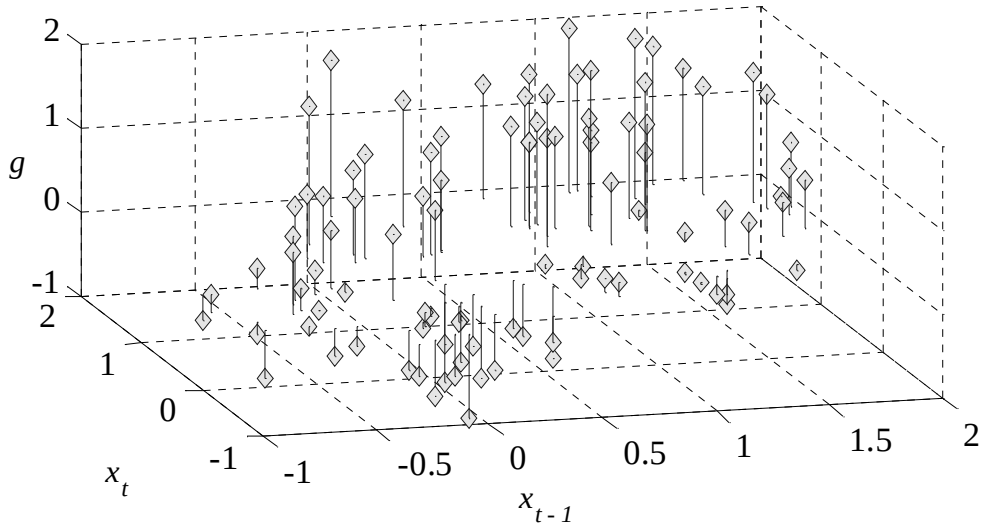

Figure 5.21 - Sinusoidal with Noise Augmented Phase Space (Testing)

Result	Value
Cluster cardinality	8
Cluster mean eventness	1.4
Cluster standard deviation eventness	0.47
Not cluster cardinality	90
Not cluster mean eventness	0.41
Not cluster standard deviation eventness	0.76
z_r	-0.48
α_s	6.3×10^{-1}
z_m	5.3
α_m	6.1×10^{-8}

Table 5.7 - Sinusoidal with Noise Results (Testing)

The patterns discovered in the training phase and applied in the testing phase are statistically significant as seen by the α_m statistic, but not the α_r statistic. The cluster mean eventness also is greater than the non-cluster mean eventness. Therefore, even though one of the statistical tests is not significant, the TSDM method was able to find a significant temporal pattern cluster (although because of the noise not every maximal point is accurately predicted).

This is illustrated in Figure 5.22, which shows the predictions and error range when the temporal pattern cluster is applied to the testing time series.

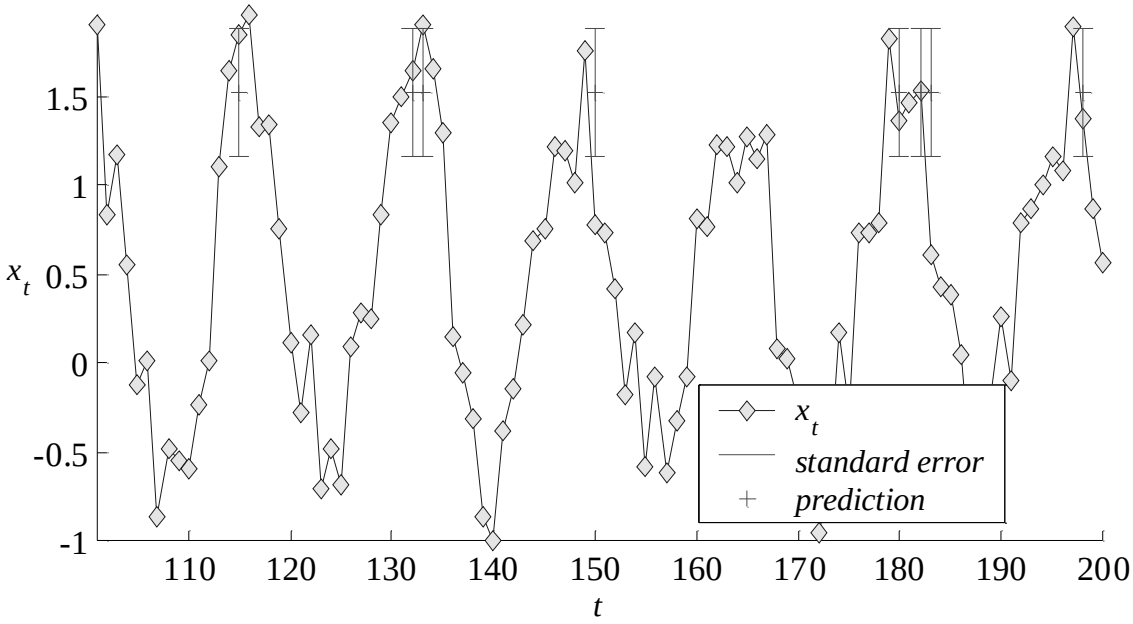

Figure 5.22 - Sinusoidal with Noise Time Series with Predictions (Testing)

This example further reveals the data mining nature of the TSDM method. The temporal pattern cluster does not characterize the whole time series or every highest value; rather it characterizes a sequence that leads to an observation with high eventness. The next section provides a further example of the TSDM methods capabilities.

5.4 Synthetic Seismic Time Series

This example analyzes in detail the previously presented synthetic seismic time series, which is generated from a randomly occurring temporal pattern, synthetic earthquake, and a contaminating noise signal. The noise is defined by (5.2).

The observed time series is illustrated in Figure 5.23. The measured variance of the contaminating noise is 3.3×10^{-3} and of the temporal pattern with synthetic earthquake is 1.3. Without the synthetic earthquake, the variance of the temporal pattern is 0.10. The measured SNR is 396 (26.0dB) for the temporal pattern and synthetic earthquake and 30.2 (14.8dB) for the temporal pattern without the synthetic earthquake.

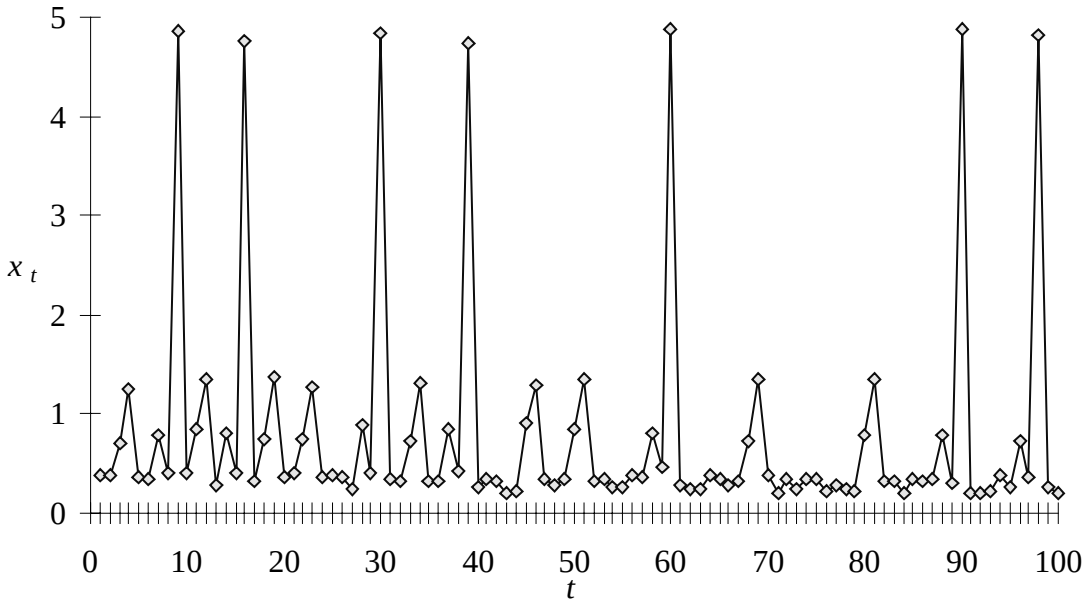

Figure 5.23 – Synthetic Seismic Time Series (Observed)

The TSDM goal for this time series is to characterize the synthetic earthquakes one time-step ahead. To capture this goal, the event characterization function is $g(t) = x_{t+1}$, and the objective function is

$$f(P) = \frac{\mu_M - \mu_{\tilde{M}}}{\sqrt{\frac{\sigma_M^2}{c(M)} + \frac{\sigma_{\tilde{M}}^2}{c(\tilde{M})}}}. \tag{5.5}$$

This objective function is useful for identifying temporal pattern clusters that are statistically significant and have a high average eventness. The optimization formulation is $\max f(P)$ subject to $\min b(P)$ such that minimizing $b(\delta)$ does not change the value of $f(P)$.

Composed of a temporal pattern, synthetic earthquake, and noise, the time series embeds to a set of small clusters in the phase space as illustrated in Figure 5.24. Figure

5.25 shows the augmented phase space, which clearly indicates the different eventness values associated with the small clusters of phase space points.

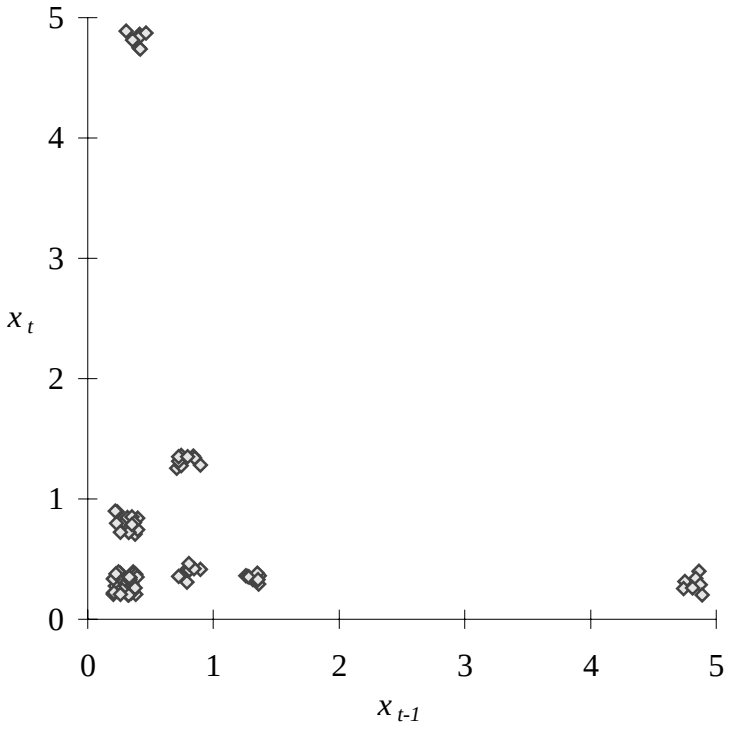

Figure 5.24 – Synthetic Seismic Phase Space (Observed)

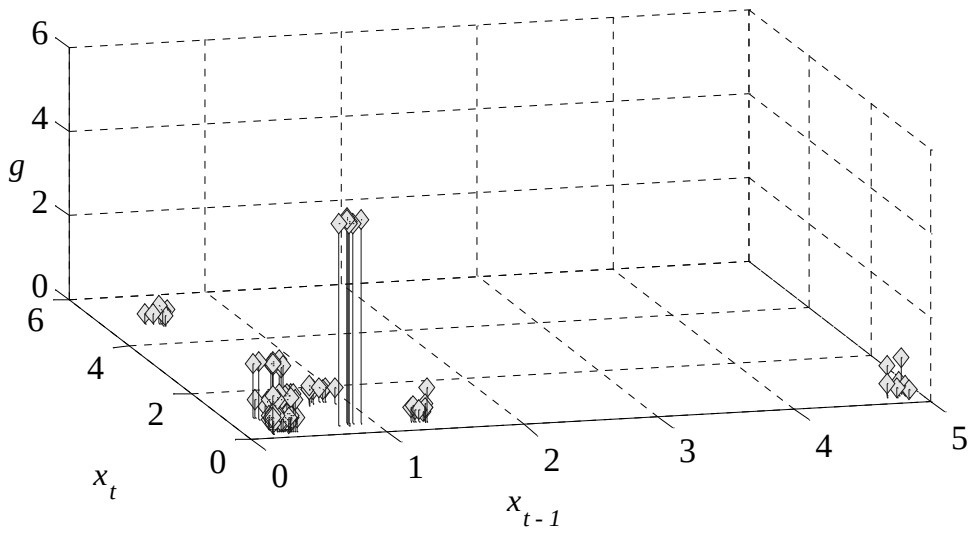

Figure 5.25 – Synthetic Seismic Augmented Phase Space (Observed)

The search parameters are presented in Table 5.1. The training stage results are shown in Table 5.8.

Result	Value
Temporal pattern, **p**	[0.92 0.51]
Threshold, δ	0.37
Cluster cardinality, $c(M)$	7
Cluster mean eventness, μ_M	4.8
Cluster standard deviation eventness, σ_M	0.058
Non-cluster cardinality, $c(\tilde{M})$	91
Non-cluster mean eventness, $\mu_{\tilde{M}}$	0.50
Non-cluster standard deviation eventness, $\sigma_{\tilde{M}}$	0.33
z_r	-9.5
α_r	3.0×10^{-21}
z_m	104
α_m	0

Table 5.8 – Synthetic Seismic Results (Observed)

The discovered temporal pattern cluster is statistically significant by both statistical tests. Figure 5.26 illustrates the temporal pattern $\mathbf{p} = [0.92\ 0.51]$ with threshold $\delta = 0.37$ in the phase space. A comparison of Figure 5.25 and Figure 5.26 demonstrates that the training stage found the best temporal pattern cluster, i.e., when a sequence of time series observations match the temporal pattern cluster, the next observation is a synthetic earthquake.

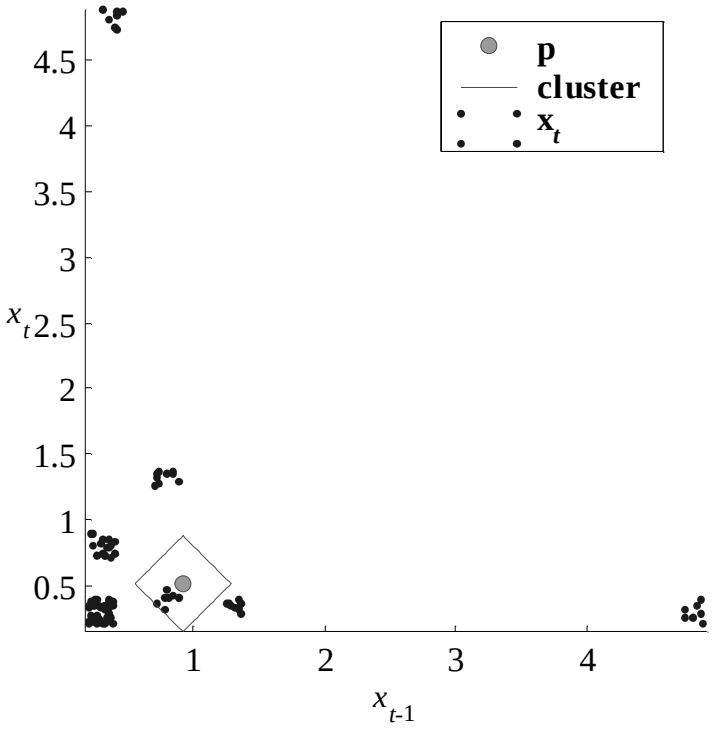

Figure 5.26 – Synthetic Seismic Phase Space with Temporal Pattern Cluster (Observed)

The synthetic seismic testing time series, a continuation of the observed time series, is illustrated in Figure 5.27. The measured variance of the noise is 3.5×10^{-3} and of the temporal pattern with synthetic earthquake is 1.9. The measured variance of the temporal pattern without synthetic earthquake is 0.10. The measured SNR is 536 (27dB) for the temporal pattern with synthetic earthquake, and 29.0 (14.6dB) for the temporal pattern without synthetic earthquake.

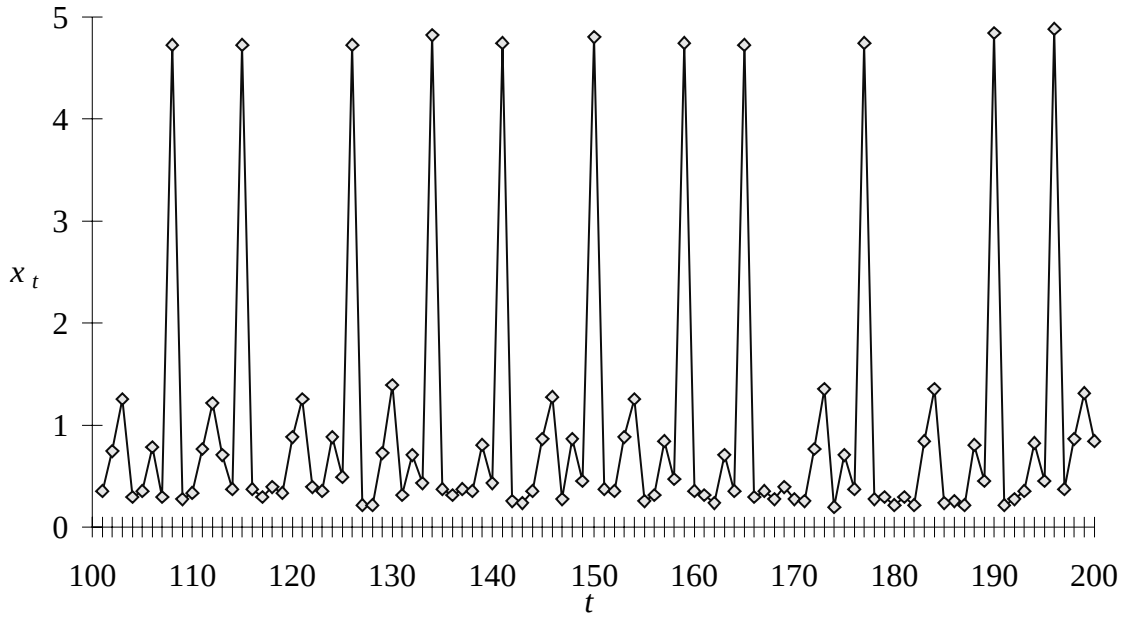

Figure 5.27 – Synthetic Seismic Time Series (Testing)

The testing time series is transformed into the phase space as shown in Figure 5.28. The augmented phase space for the testing time series is seen in Figure 5.29.

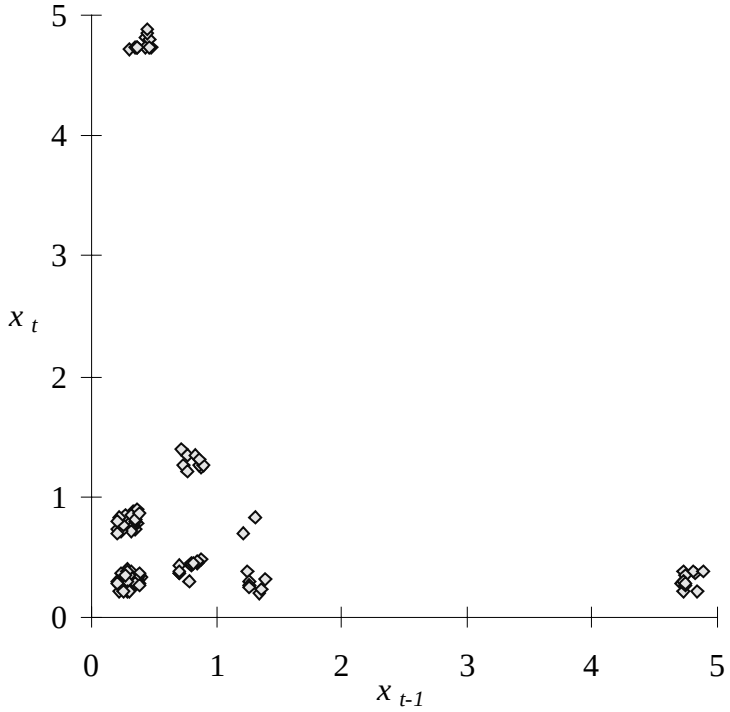

Figure 5.28 – Synthetic Seismic Phase Space (Testing)

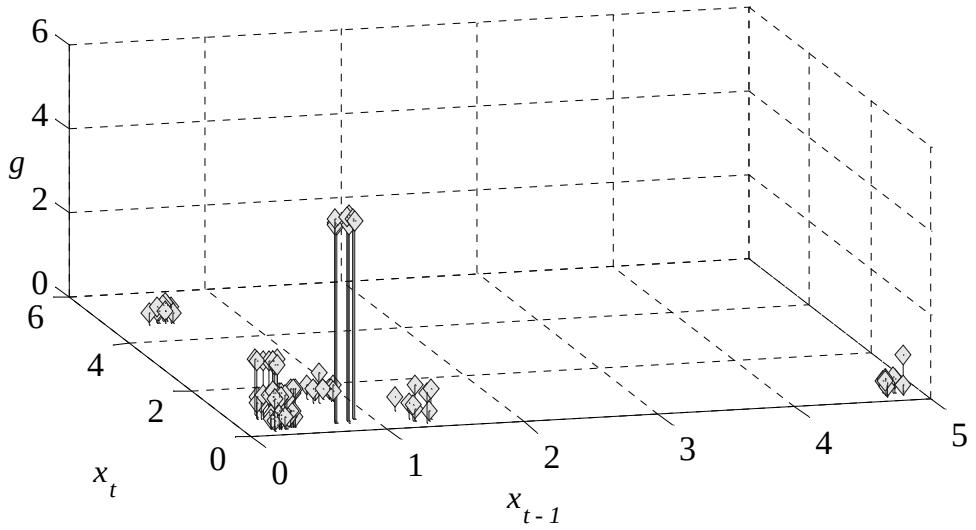

Figure 5.29 – Synthetic Seismic Augmented Phase Space (Testing)

The testing stage results presented in Table 5.9 are statistically significant as seen by the α_r and α_m statistics.

Result	Value
Cluster cardinality, $c(M)$	11
Cluster mean eventness, μ_M	4.8
Cluster standard deviation eventness, σ_M	0.056
Non-cluster cardinality, $c(\tilde{M})$	87
Non-cluster mean eventness, $\mu_{\tilde{M}}$	0.53
Non-cluster standard deviation eventness, $\sigma_{\tilde{M}}$	0.33
z_r	-9.6
α_r	8.5×10^{-22}
z_m	107
α_m	0

Table 5.9 – Synthetic Seismic Results (Testing)

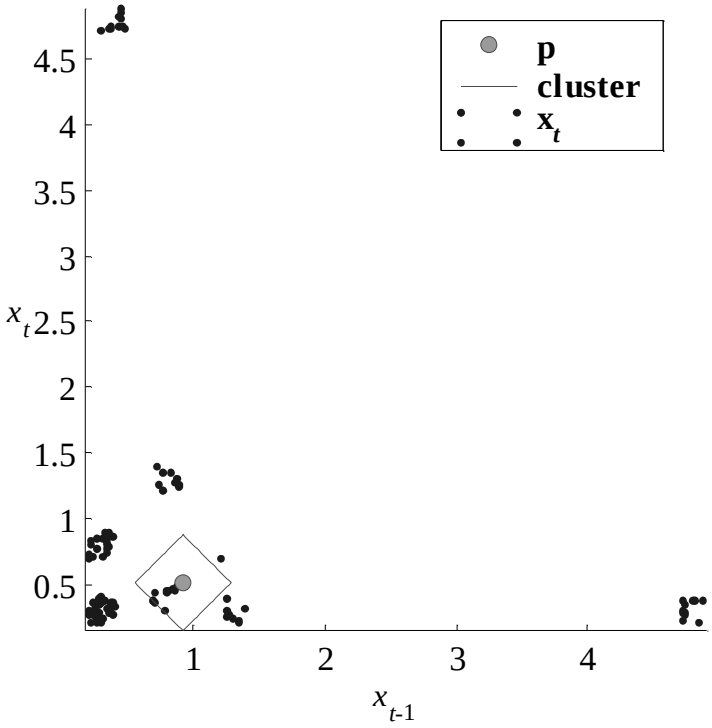

Figure 5.30 – Synthetic Seismic Phase Space with Temporal Pattern Cluster (Testing)

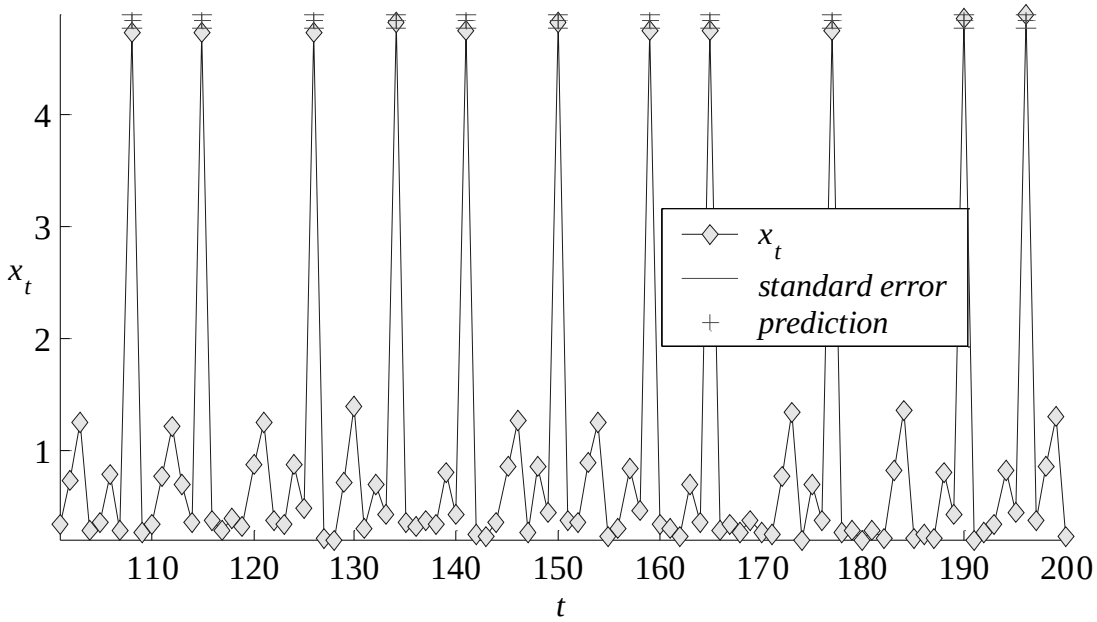

Figure 5.31 – Synthetic Seismic Time Series with Predictions (Testing)

Figure 5.30 highlights the temporal pattern cluster in the testing phase space. Figure 5.31 clearly illustrates the prediction accuracy of the testing stage by highlighting the predictions and error range on the testing time series. This example further reveals the strength of the TSDM method – its ability to predict events.

In this chapter, the TSDM method has been applied successfully to the sinusoidal, random noise, sinusoidal with noise, and synthetic seismic example time series. Each example time series highlighted the capabilities of the TSDM method. The sinusoidal time series highlighted the event-capturing capability of the TSDM method. With the sinusoidal time series, each peak point in the time series was characterized and predicted as an event. The noise time series showed that the method correctly determined that there are no temporal patterns in random noise. The sinusoidal with noise time series showed that the method, although affected by noise, can still predict maximal values.

The synthetic seismic time series demonstrates the full power of the TSDM method. The time series is the composite of a temporal pattern, a synthetic earthquake that occur non-periodically, and contaminating noise. With this time series, the method accurately characterized and predicted all of the events.

Chapter 6 presents several extensions to the TSDM method, including variations that search for temporal patterns in multi-dimensional time series and find multiple temporal pattern clusters. In Chapters 7 and 8, the TSDM method is applied to real world problems.

Chapter 6 Extended Time Series Data Mining Methods

This chapter presents three extensions to the Time Series Data Mining (TSDM) method. The first variation extends the TSDM method to multi-dimensional time series by adapting the time-delay embedding process. For simplicity, it is called the TSDM-M/x (Time Series Data Mining multi-dimensional time series) method. The second TSDM extension searches for multiple temporal pattern clusters. It is called the Time Series Data Mining multiple temporal pattern (TSDM-x/M) method, where the x may be either S or M depending on the dimensionality of the time series.

Additionally, this chapter discusses alternative clustering methods and temporal pattern stationarity. In Chapter 4, the TSDM method employed a temporal pattern cluster that was formed with a hypersphere in a Manhattan phase space. By changing the distance metric associated with the phase space, alternative cluster shapes are achieved.

Nonstationary temporal patterns are addressed with two techniques. The first is by applying the integrative techniques from the ARIMA method to transform nonstationary temporal pattern clusters into stationary ones. The second is through an extension to the TSDM method, called the Time Series Data Mining evolving (TSDMe) method.

The chapter concludes with a discussion of diagnostics for improving TSDM results.

6.1 Multiple Time Series (TSDM-M/x)

This section discusses the TSDM-M/x method [2], which allows data from multiple sensors to be fused. The TSDM method is adapted by modifying the time-delay embedding process to incorporate observations from each dimension of a multi-

dimensional time series. Intuitively, additional sensors on a system will provide additional information assuming they are not sensing the same state variable. Therefore, the time series generated by these sensors will provide a richer set of observations from which to form the reconstructed phase space. This has been shown experimentally by Povinelli and Feng [2].

The multi-dimensional time series

$$\mathbf{X} = \{\vec{x}_t, t = 1, \ldots, N\} \tag{6.1}$$

is a sequence of *N* vector observations, where $\vec{x}_t$ is an *n*-dimensional vector. This collection of observed time series may be represented as a matrix

$$\mathbf{X} = \begin{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_1 & \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_2 & \cdots & \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_N \end{bmatrix}. \tag{6.2}$$

The corresponding multi-dimensional testing time series **Y** takes the form

$$\mathbf{Y} = \{\vec{x}_t, t = R, \ldots, S\} \quad N < R < S \text{, or} \tag{6.3}$$

$$\mathbf{Y} = \begin{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_R & \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_{R+1} & \cdots & \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_S \end{bmatrix}. \tag{6.4}$$

Since the vector time series is *n*-dimensional, the dimension of the phase space is *n*·*Q*. As with the TSDM method, a metric *d* is defined on the phase space. The observed time series are embedded into the phase space yielding phase space points or $(n \cdot Q) \times 1$ phase space vectors

$$\mathbf{x}_t = \left(\vec{x}^T_{t-(Q-1)\tau}, \ldots, \vec{x}^T_{t-\tau}, \vec{x}^T_t\right)^T, t \in \Lambda, \tag{6.5}$$

where $\Lambda = \{t : t = (Q-1)\tau + 1, \ldots, N\}$. Likewise, the collection of testing time series is embedded yielding $\mathbf{y}_t$. The dimensionality of the phase space and modified embedding process are adaptations of the TSDM method required to yield the TSDM-M/x method.

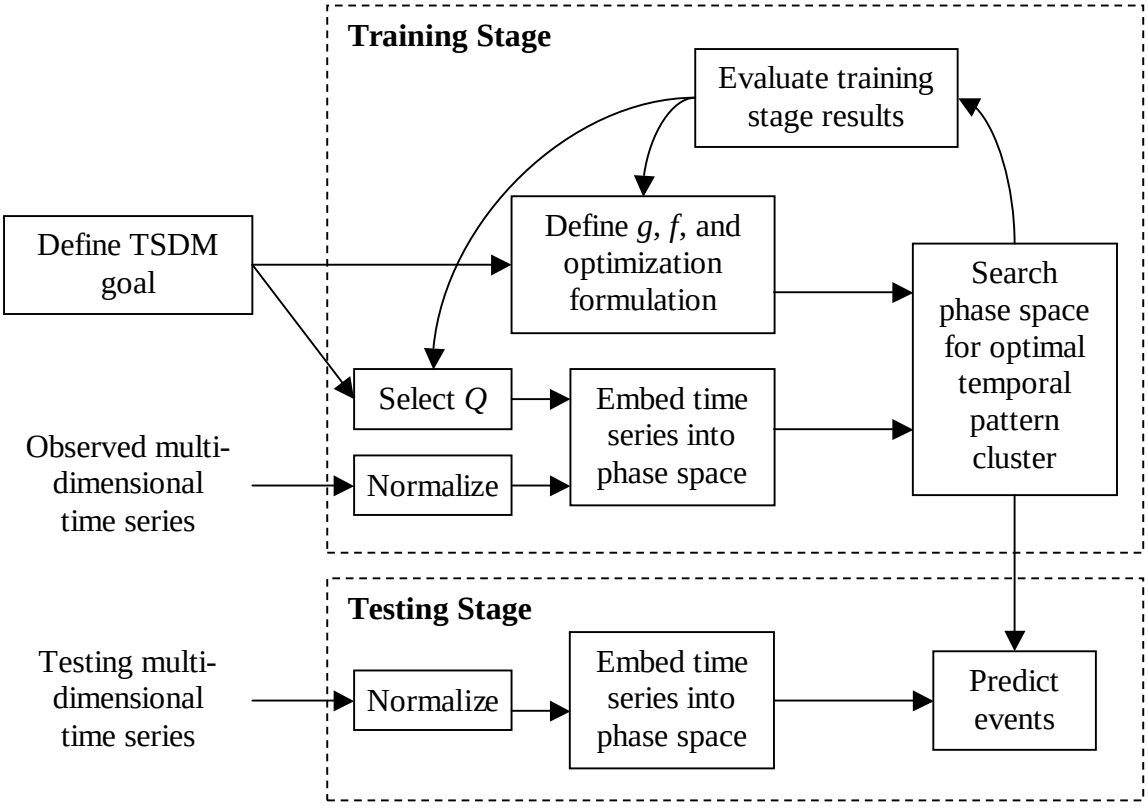

Figure 6.1 – Block Diagram of TSDM-M/x Method

As illustrated in Figure 6.1, a normalization step may be added to force each dimension of the multi-dimensional time series to have the same range. Normalization does not change the topology of the phase space, but mapping each time series onto the same range allows the use of similar search step sizes for each phase space dimension. This normalization assists the optimization routines. The normalization constant used in the training stage is retained for use in predicting events in the testing stage.

The next section present a variation of the TSDM method that searches for multiple temporal pattern clusters.

6.2 Multiple Temporal Patterns (TSDM-x/M)

The TSDM method finds a single hyperspherical temporal pattern cluster. The temporal patterns to be characterized may not conform to a hyperspherical shape or may consist of multiple disjoint regions, as shown in Figure 6.2.

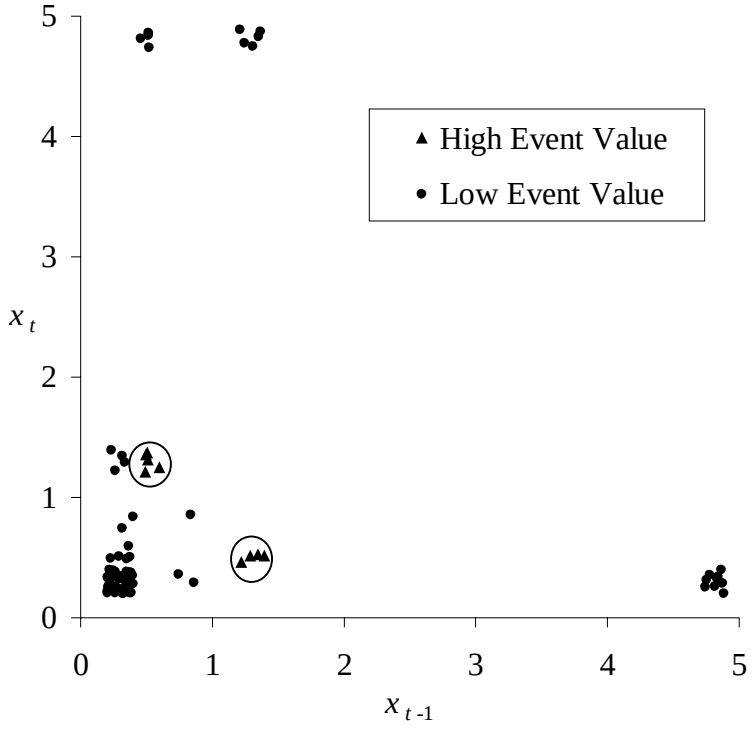

Figure 6.2 – Multiple Temporal Pattern Cluster Phase Space

The triangles have high eventness values and the dots have low eventness values. However, there is not a single hypersphere that can contain all the high eventness phase space points and exclude all of the low eventness ones. Two temporal pattern clusters are needed. A new method for finding a collection of temporal pattern clusters also is needed.

In order to find a collection of temporal patterns, the objective function is modified to include the phase space points within each of the temporal pattern clusters $P_i \in \mathcal{C}, i = 1, 2, \ldots$. The example objective function given by (3.15) is extended to yield

$$f(\mathcal{C}) = \frac{\mu_M - \mu_{\tilde{M}}}{\sqrt{\frac{\sigma_M^2}{c(M)} + \frac{\sigma_{\tilde{M}}^2}{c(\tilde{M})}}}, \tag{6.6}$$

where the index set M is defined more generally, i.e.

$$M = \{t : \mathbf{x}_t \in P_i, t \in \Lambda\}, \tag{6.7}$$

where $P_i \in \mathcal{C}, i = 1, 2, \ldots$. Similarly, $\tilde{M}$, the complement of *M*, is the set of all time indices *t* when $\mathbf{x}_t$ is not in any $P_i \in \mathcal{C}$. This objective function is useful for identifying temporal pattern clusters that are statistically significant and have a high average eventness.

Another example objective function, the ratio of correct predictions to all predictions,

$$f(\mathcal{C}) = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{6.8}$$

was first defined in (3.18) and requires no modification to work in the TSDM-x/M method.

The optimization formulation

$$\max_{P_i} f(\mathcal{C}) \tag{6.9}$$

may be used, but it may lead to the following set of temporal pattern clusters illustrated in Figure 6.3. A simpler and therefore more preferable solution is illustrated in Figure 6.4.

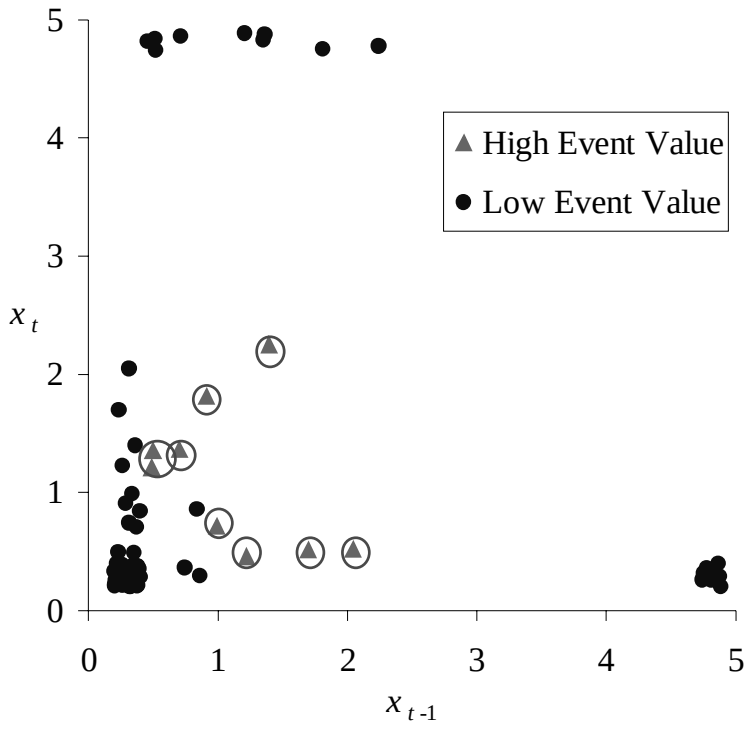

Figure 6.3 – Multiple Cluster Solution With Too Many Temporal Pattern Clusters

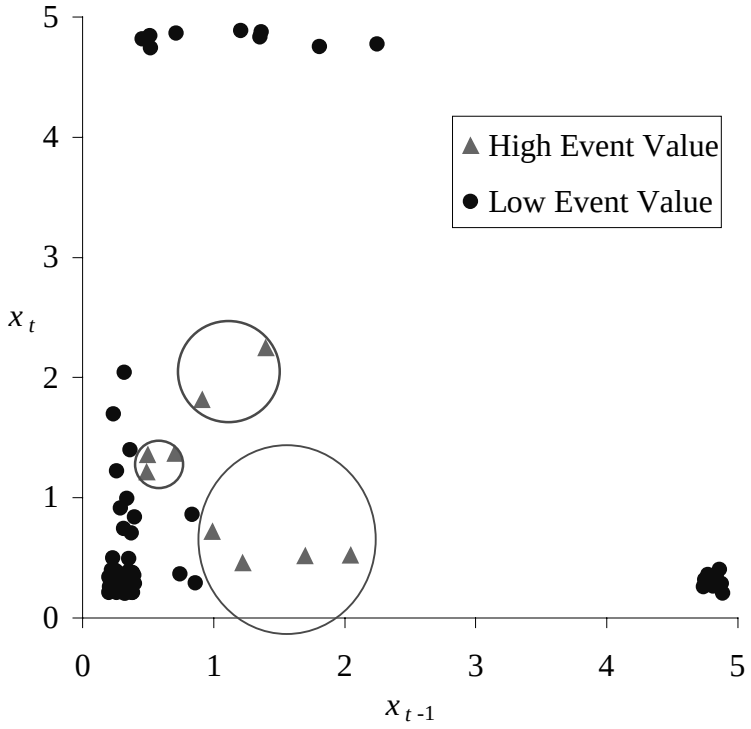

Figure 6.4 – Multiple Cluster Solution

To achieve the preferred solution the optimization formulation is $\max f(\mathcal{C})$ subject to $\min c(\mathcal{C})$ such that minimizing $c(\mathcal{C})$ does not change the value of $f(\mathcal{C})$. A bias also may be placed on the δs yielding the optimization formulation $\max f(\mathcal{C})$ subject to $\min c(\mathcal{C})$ such that minimizing $c(\mathcal{C})$ does not change the value of $f(\mathcal{C})$ and $\min \delta_i \; \forall P_i \in \mathcal{C}$ such that minimizing $\delta_i \; \forall P_i \in \mathcal{C}$ does not change the value of $c(\mathcal{C})$. These staged optimizations are resolved through the genetic algorithm tournament tiebreaker system introduced in Chapter 4.

Given a TSDM goal, a target observed time series to be characterized, and a testing time series to be predicted, the steps in the TSDM-x/M method are essentially the same as the steps in the TSDM method. The modifications are that a range of phase space dimensions is chosen, and the search processes is iterative. The steps of the TSDM-x/M method are given below.

I. Training Stage (Batch Process)

1. Frame the TSDM goal in terms of the event characterization function, objective function, and optimization formulation.
    a. Define the event characterization function, *g*.
    b. Define the objective function, *f*.
    c. Define the optimization formulation, including the independent variables over which the value of the objective function will be optimized and the constraints on the objective function.
    d. Define the criteria to accept a temporal pattern cluster.
2. Determine the range of *Q*'s, i.e., the dimension of the phase space and the length of the temporal pattern.

3. Embed the observed time series into the phase space using the time-delayed embedding process.
4. Associate with each time index in the phase space an eventness represented by the event characterization function. Form the augmented phase space.
5. Search for the optimal temporal pattern cluster in the augmented phase space using the following algorithm.

   if the temporal pattern cluster meets the criteria set in 1.d then,

   repeat step 5 after removing the clustered phase space points from the phase space.

   elseif the range of Q is not exceeded, increment Q and goto step 2

   else goto step 6
6. Evaluate training stage results. Repeat training stage as necessary.

II. Testing Stage (Real Time or Batch Process)

1. Embed the testing time series into the phase spaces.
2. Apply the temporal pattern clusters to predict events.
3. Evaluate testing stage results.

This section presented an extension of the TSDM method that allows multiple temporal pattern clusters to be discovered. The next section presents a set of techniques that allow more complicated temporal pattern clusters to be identified.

6.3 Other Useful TSDM Techniques

This section presents three techniques that are useful in the process of identifying optimal temporal pattern clusters. The first is a method for changing the temporal pattern cluster shape by employing different phase space metrics. The next two techniques are useful for time series with nonstationary temporal pattern clusters.

6.3.1 Clustering Technique

The phase space metric used in the synthetic seismic time series example from Chapter 4 was the Manhattan or l_1 distance. Obviously, this is not the only applicable metric. With alternative metrics, the shape of the temporal pattern cluster can be changed. The l_p norms provide a simple mechanism for changing the temporal pattern cluster shape without increasing the search space dimensionality. The l_p norm is defined as

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \quad \text{[56, p. 29].} \tag{6.10}$$

Figure 6.5 illustrates five different norms: $l_{0.5}$, l_1, l_2, l_3, and l_∞. The temporal pattern cluster is located in a two-dimensional space at (0,0) with $\delta = 1$.

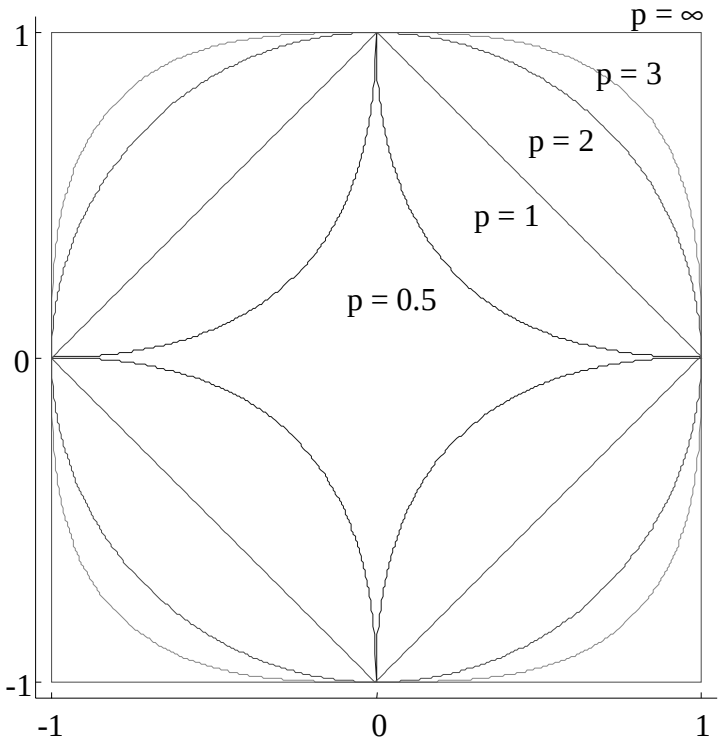

Figure 6.5 – Cluster Shapes of Unit Radius for Various l_p Norms

When the l_2, Euclidean, norm is used the cluster is a circle. Using the l_1 and l_∞ norms, the temporal pattern cluster is a square. These alternative cluster shapes are incorporated into the method by simply defining the phase space using the desired l_p norm. The next section presents a technique for identifying nonstationary temporal pattern clusters.

6.3.2 Filtering Technique

In Chapter 2, ARIMA time series analysis was discussed. ARIMA modeling requires that the time series be stationary. TSDM's requirement is less stringent. Only the temporal pattern cluster must be stationary, i.e., the phase space points characteristic of events must remain within the temporal pattern cluster. In Chapter 2, a set of filters were presented that could transform linear and exponential trend time series, which are nonstationary, into stationary ones. These same filters also are useful for transforming

time series with nonstationary temporal pattern clusters into time series with stationary temporal pattern clusters.

The following example shows how a nonstationary time series can be made stationary and the appearance of a nonstationary time series in the phase space and augmented phase space. The observed time series $X = \{x_t = .02t, t = 1,\ldots,100\}$ is illustrated in Figure 6.6.

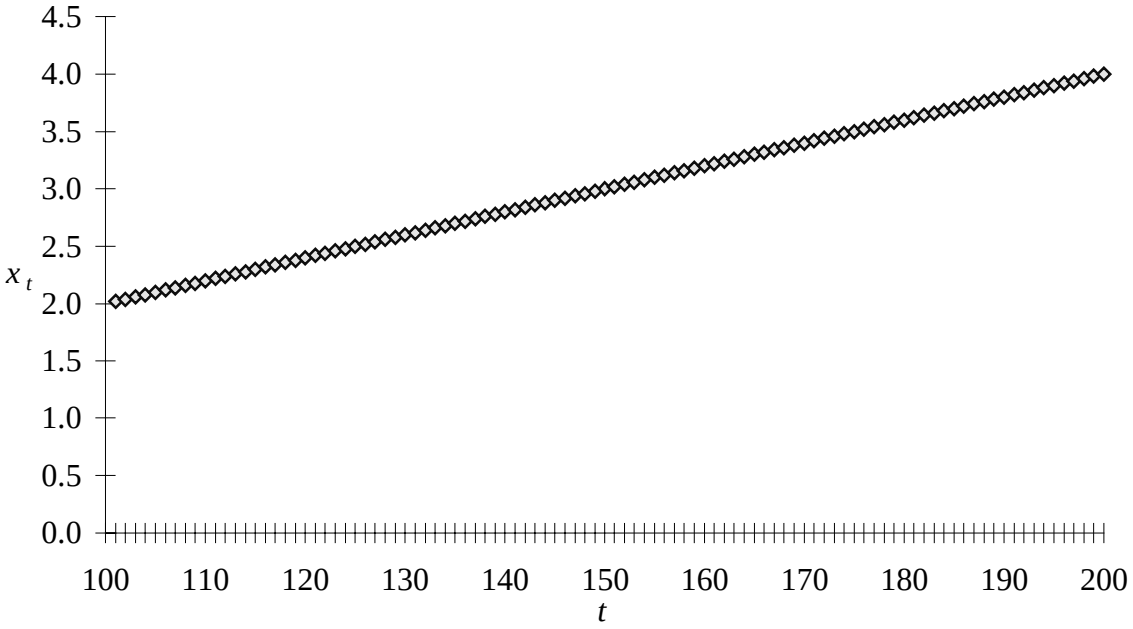

Figure 6.6 – Linearly Increasing Time Series (Observed)

The TSDM goal is to characterize and predict all observations. Thusly, the event characterization function is $g(t) = x_{t+1}$. The corresponding objective function (described in Chapter 3) is

$$f(P) = \begin{cases} \mu_M & \text{if } c(M)/c(\Lambda) \geq \beta \\ (\mu_M - g_0)\dfrac{c(M)}{\beta c(\Lambda)} + g_0 & \text{otherwise} \end{cases}, \qquad (6.11)$$

where $\beta = 0.05$. The optimization formulation is $\max f(P)$ subject to $\min \delta$.

Figure 6.7 presents the Euclidean phase space, and Figure 6.8 illustrates the augmented phase space. Since the time series has a linearly increasing value, it embeds as a line in both spaces. The linear feature of the phase space points indicates nonstationarity.

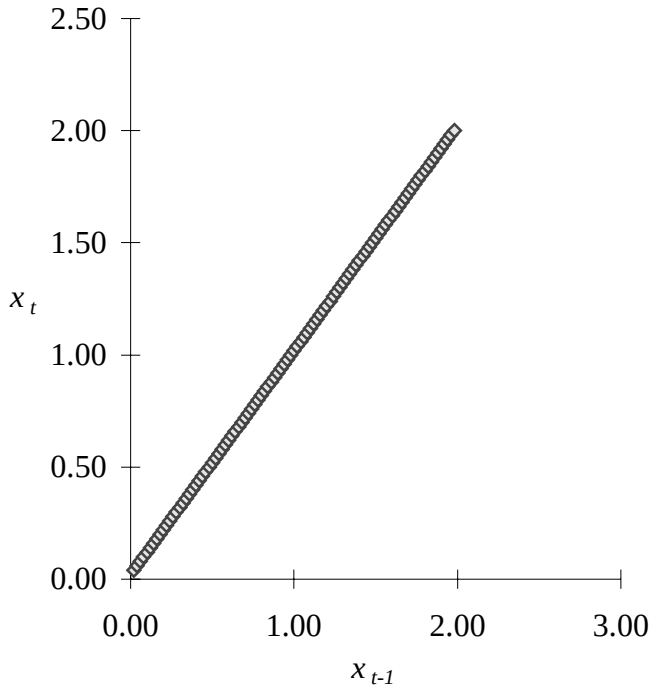

Figure 6.7 – Linearly Increasing Phase Space (Observed)

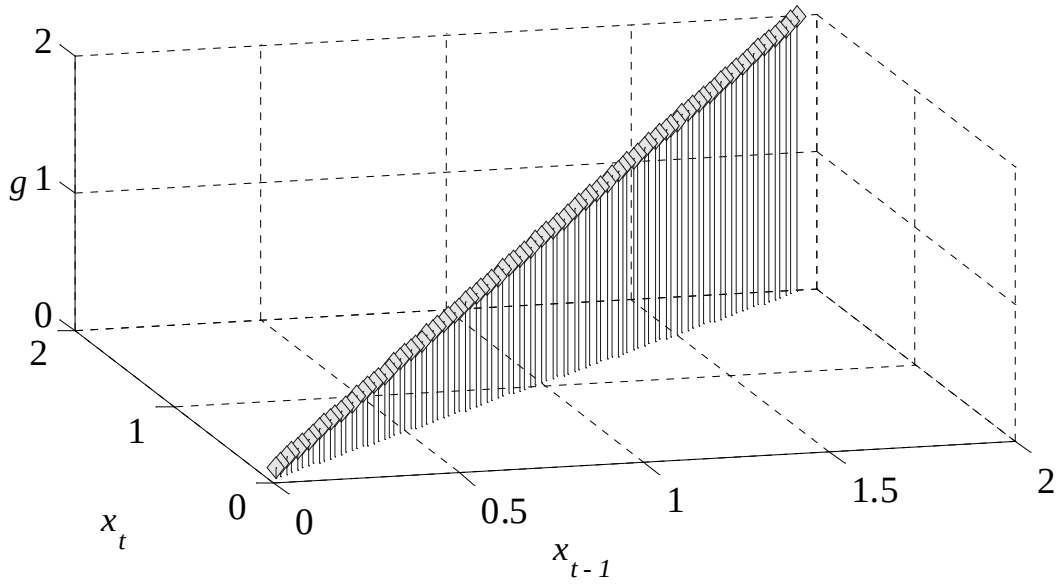

Figure 6.8 – Linearly Increasing Augmented Phase Space (Observed)

The genetic algorithm search parameters are presented in Table 6.1.

Parameter	Value
Random search multiplier	1
Population size	20
Elite count	1
Gene length	8
Tournament size	2
Mutation rate	0.2%
Convergence criteria	1

Table 6.1 – Genetic Algorithm Parameters for Linearly Increasing Time Series

The training stage results are shown in Figure 6.9, which demonstrates that the temporal pattern cluster does not capture the linearly increasing nature of the time series. This will become more evident in the testing stage of the TSDM method.

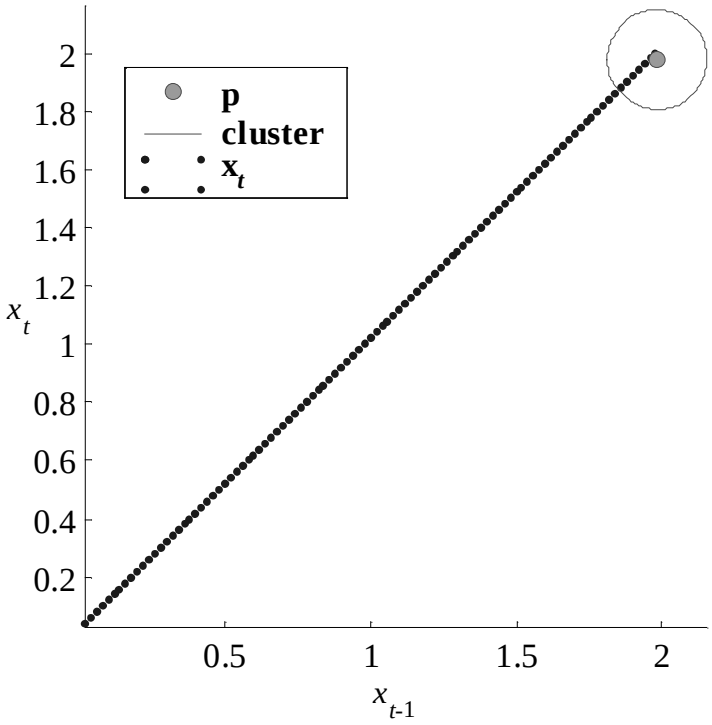

Figure 6.9 – Linearly Increasing Phase Space with Temporal Pattern Cluster (Observed)

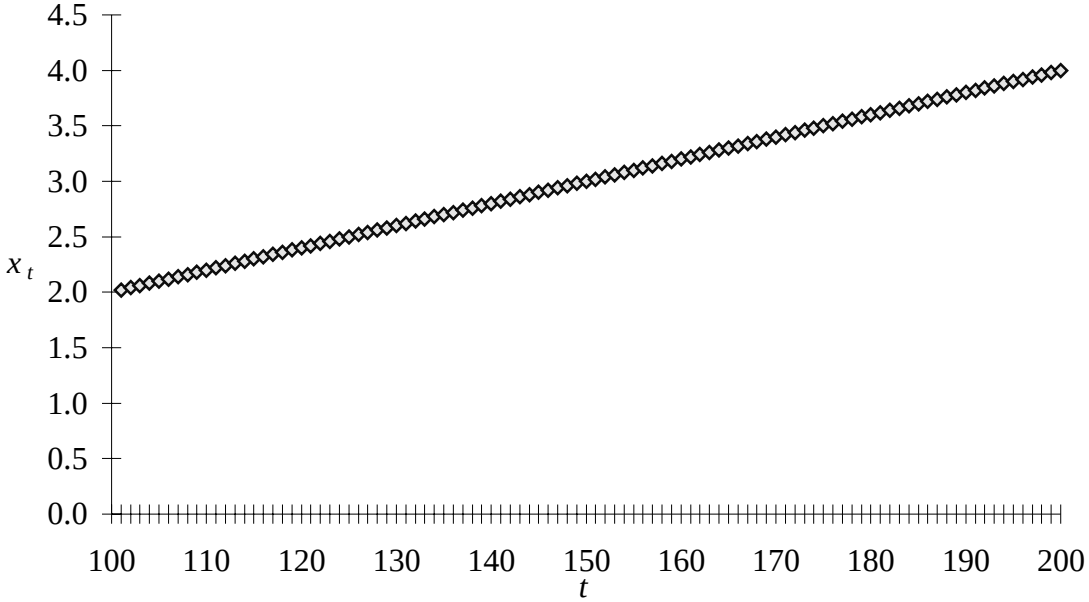

Figure 6.10 – Linearly Increasing Time Series (Testing)

The testing time series is illustrated in Figure 6.10.

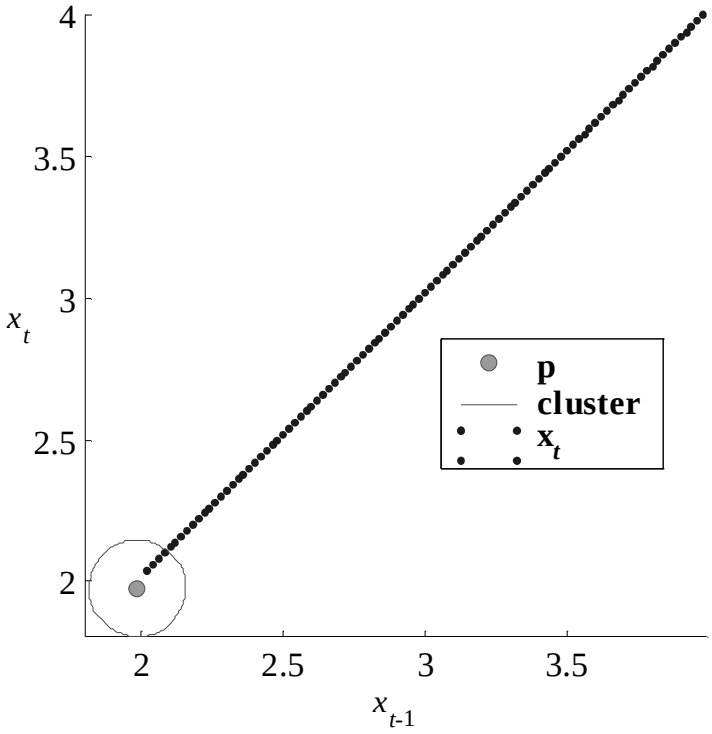

Figure 6.11 – Linearly Increasing Phase Space with Temporal Pattern Cluster (Testing)

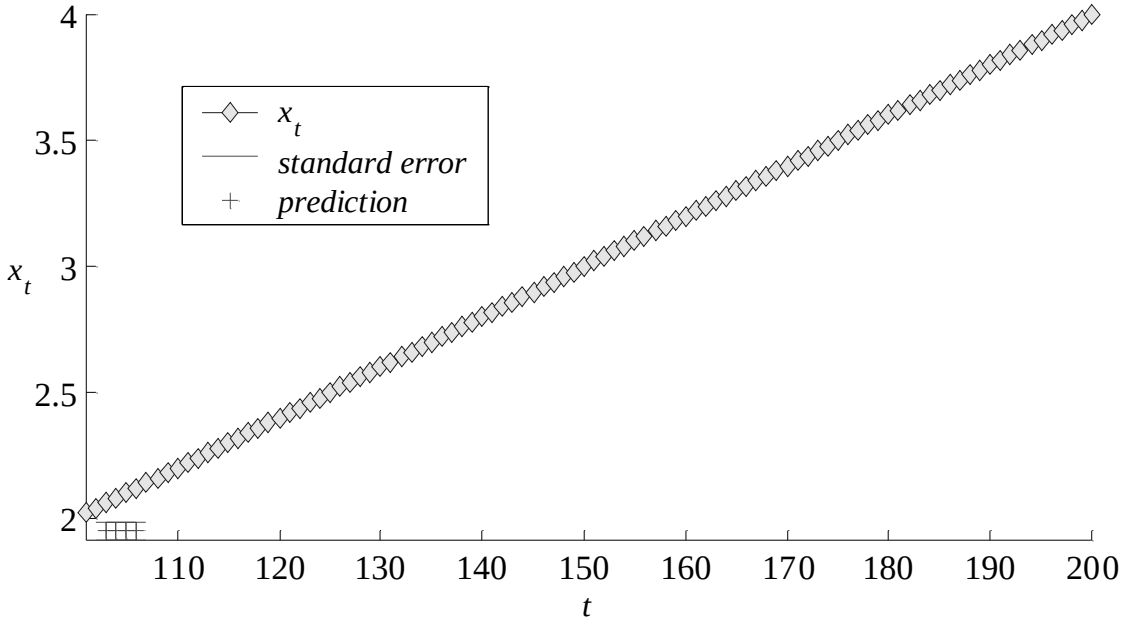

Figure 6.12 – Linearly Increasing Time Series with Predictions (Testing)

Figure 6.11 highlights the temporal pattern cluster in the phase space. Obviously, as illustrated by Figure 6.11, the desired TSDM goal is not met, which is reinforced by Figure 6.12. The cause of the prediction failure is the lack of temporal pattern stationarity, not necessarily because of time series nonstationarity. The resolution to the problem of temporal pattern nonstationarity is achieved by applying the filtering techniques discussed in Chapter 2. Applying the first difference filter to the observed time series *X* yields $Z = \{z_t = .02, t = 2, \ldots, 100\}$, which is a constant-value time series. The problem is now trivial.

Although some time series may be made stationary through filtering techniques, these methods will not convert all nonstationary time series into stationary ones. The next section presents a method for analyzing time series with quasi-stationary temporal pattern clusters.

6.3.3 Non-filtering Techniques

Although stationarity usually describes the statistical characteristics of a stochastic time series [55, pp. 297-298], this dissertation introduces a more general definition. When applied to a deterministic time series, stationarity indicates that the periodicity, if the time series is periodic, and range of the time series are constant. When applied to chaotic time series, stationarity indicates that the attractors remain constant through time. Chaotic time series whose underlying attractors evolve through time are classified as nonstationary chaotic time series.

Beyond filtering to extract nonstationary temporal patterns, there are two TSDM methods presented in this section that address quasi-stationary temporal patterns, i.e., temporal patterns that are characteristic and predictive of events for a limited time

window. They are called the Time Series Data Mining evolving temporal pattern (TSDMe) methods. These methods are useful for analyzing time series generated by adaptive systems such as financial markets with feedback characteristics that counteract systemic predictions.

The first method ($TSDMe_1$) uses a fixed training window and a fixed prediction window. The second method ($TSDMe_2$) uses a fixed training window and a single period prediction window. The TSDMe methods differ from the other TSDM methods in how the observed and testing time series are formed.

The $TSDMe_1$ method divides the time series into equally sized sets $X_j = \{x_t, t = (j-1)N+1, \ldots, jN\}$, where N is the number of observations in a subset of X, and j is the index of the subset. The time series X_j is used in the training stage. The time series X_{j+1} is used in the testing stage. The length of the time window N is determined experimentally such that the temporal patterns clusters remain quasi-stationary between any two adjacent time windows.

The $TSDMe_2$ method creates the overlapping observed time series as follows:

$$X_j = \{x_t, t = j, \ldots, j+N\}. \tag{6.12}$$

The testing time series is formed from a single observation as follows:

$$Y_j = \{x_t, t = j+N+1\}. \tag{6.13}$$

With these changes in the formation of the observed and testing time series, any of the TSDM methods may be applied.

The last section in this chapter presents a set of cases with which to diagnose and adjust the TSDM method.

6.4 Evaluating Results and Adjusting Parameters

In the training stage of the TSDM methods, there is an *evaluate training stage results* step, which is an ad hoc evaluation of the intermediate and final results of the TSDM method. The evaluation may include visualization of the phase space and augmented phase space and review of the statistical results. Based on the ad hoc evaluation, the parameters of the method may be adjusted, alternative TSDM methods selected, and/or appropriate TSDM techniques applied. This section discusses ad hoc evaluation techniques, what issues they might discover, and possible solutions.

By parsimony, the simplest characterization of events possible is desired, i.e., as small a dimensional phase space as possible and as few temporal pattern clusters as required. The first evaluation technique is to visualize, if possible, the phase space and augmented phase space, which allows human insight to identify clustering problems. The cases that may be identified and their potential solutions are listed below.

Case 1: One cluster is identifiable, but not discovered by the TSDM method.

Potential Solution A: Select alternative phase space metric.

Potential Solution B: Increase genetic algorithm population size.

Potential Solution C: Increase genetic algorithm chromosome length.

Potential Solution D: Increase genetic algorithm mutation rate.

Potential Solution E: Use alternative objective function.

Case 2: Multiple clusters are visualized, but not discovered by the TSDM method.

Potential Solution A: Use TSDM-x/M method.

Case 3: No clusters are visualized.

Potential Solution A: Try higher dimensional phase space.

Potential Solution B: Use TSDM-x/M method.

Case 4: Phase space points cluster into a line.

Potential Solution A: Apply filtering techniques.

The second evaluation technique is to review the statistical characteristics of the resulting temporal pattern cluster(s). These statistics include the $c(M)$, $c(\tilde{M})$, μ_M, σ_M, $\mu_{\tilde{M}}$, $\sigma_{\tilde{M}}$, μ_X, α_r, and α_m. The cases that may be identified and their potential solutions are listed below.

Case 5: The cluster cardinality $c(M)$ is too large or small while using the objective function described in (4.3).

Potential Solution A: Use the objective function described in (3.16).

Case 6: The cluster cardinality $c(M)$ is too large or small while using the objective function described in (3.16).

Potential Solution A: Adjust the β as appropriate.

Case 7: Either or both the α_r and α_m do not allow the null hypothesis to be rejected.

Potential Solution A: The null hypothesis holds. No temporal patterns exist in the time series.

Potential Solution B: Use the TSDM-x/M method to find multiple temporal patterns.

Potential Solution C: Use a larger training time series.

Potential Solution D: Use the $TSDMe_1$ or $TSDMe_2$ methods to see if the temporal patterns may be quasi-stationary.

Potential Solution E: Adjust the cluster shape by using an alternative p-norm.

This section presented seven cases where the resulting temporal pattern clusters did not achieve the desired TSDM goal and potential solutions for each of these cases. This is not an exhaustive list of treatments to improve the TSDM results, but a representative sample of the most common adjustments needed.

This chapter has presented extensions to the TSDM method for finding multiple temporal patterns and analyzing multi-dimensional time series. It has also presented a set of techniques for dealing with nonstationary temporal pattern clusters. It concluded with a set of diagnostic cases and their potential resolutions. The next two chapters will apply these extended TSDM methods to real-world applications.

Chapter 7 Engineering Applications

This chapter introduces a set of real-world time series gathered from sensors on a welding station. The problem is to predict when a droplet of metal will release from a welder. The welding process joins two pieces of metal into one by making a joint between them. A current arc is created between the welder and the metal to be joined. Wire is pushed out of the welder. The tip of the wire melts, forming a metal droplet that elongates (sticks out) until it releases. The goal is to predict the moment when a droplet will release, which will allow the quality of the joint to be improved. Because of the irregular, chaotic, and event nature of the droplet release, prediction is impossible using traditional time series methods.

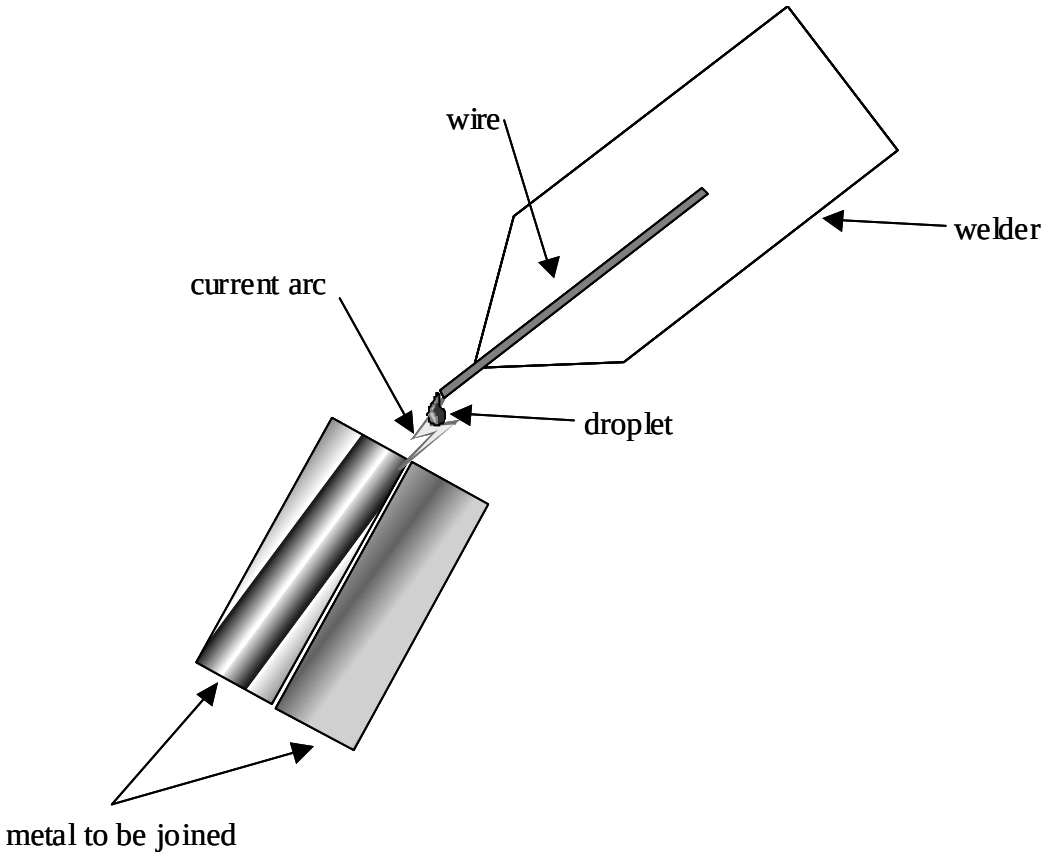

Figure 7.1 - Welder

Samples of the four welding time series are presented in Figure 7.2 and Figure 7.3. Obviously, they are noisy and nonstationary. Sensors on the welding station generate three of the time series. The first is the *stickout* of the droplet measured in pixels by an electronic camera. It is sampled at 1kHz and comprised of approximately 5,000 observations. The second time series is the *voltage* measured in decivolts from the welder to the metal to be joined. The third is the *current* measured in amperes. The voltage and current time series are sampled at 5kHz, synchronized to each other, and each comprised of approximately 35,000 observations. The fourth time series indicates the *release* of the metal droplets. This time series was created after the sensor data was collected using a process at INEEL (Idaho National Engineering & Environmental Laboratory), which also provided the data. It is synchronized with the stickout time series and comprised of approximately 5,000 observations. The release time series indicates the events with a one indicating an event and a zero indicating a non-event.

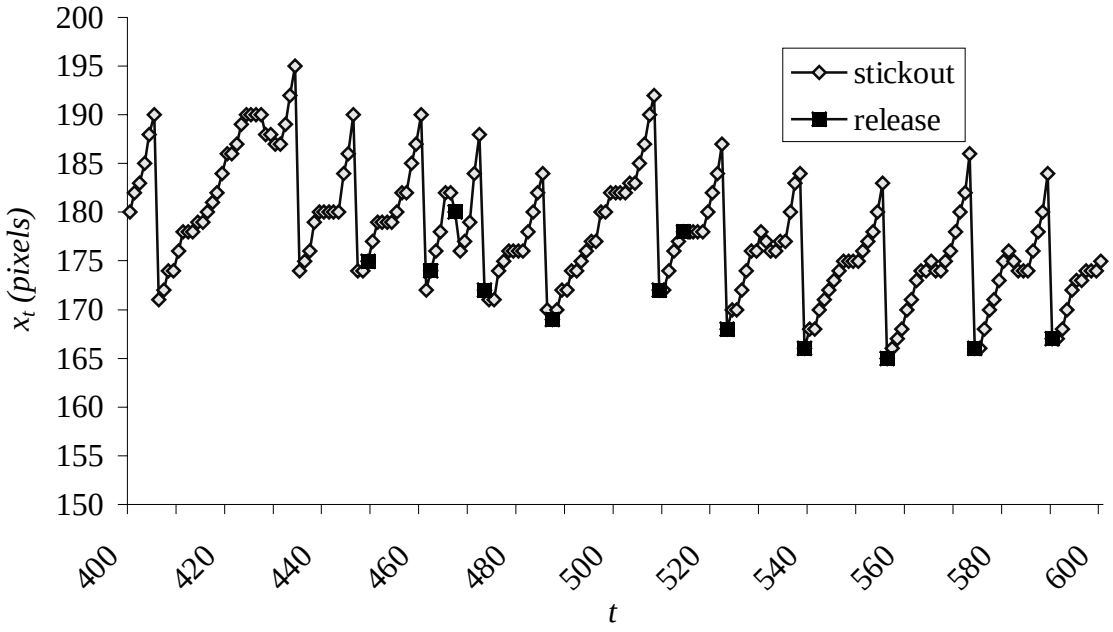

Figure 7.2 – Stickout and Release Time Series

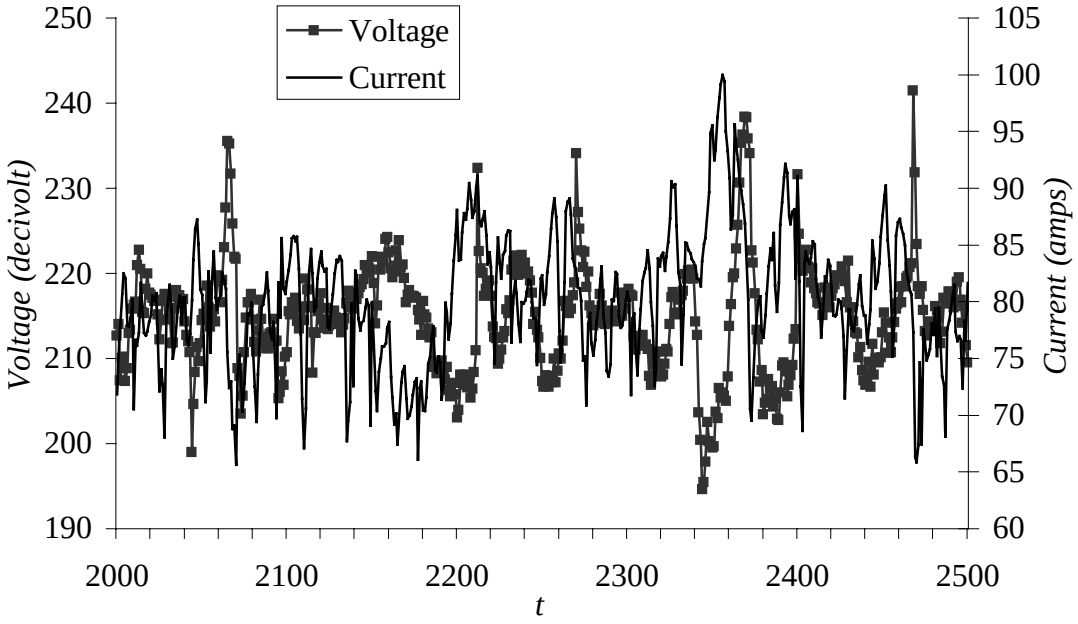

Figure 7.3 – Voltage and Current Time Series

This chapter is organized into six sections. This first section discusses the four time series that comprise the data set and provides an overview of the chapter. The

second section characterizes and predicts the release events using the stickout time series. The third section characterizes and predicts events in an adjusted release time series. The fourth section presents and resolves a time series synchronization problem. As noted above, two of the sensors sampled at approximately 5kHz, while the other sensor sampled at approximately 1kHz. The problem is complicated further because the ratio of the sampling rates is not exactly 5:1. In the fifth section, the TSDM-M/M method is applied to data from all three sensors.

7.1 Release Prediction Using Single Stickout Time Series

This section presents the results of applying the TSDM-S/M method to characterizing and predicting droplet releases using the stickout time series. This application of the TSDM-S/M method does not require the synchronization of the stickout and release time series with the current and voltage time series to be resolved.

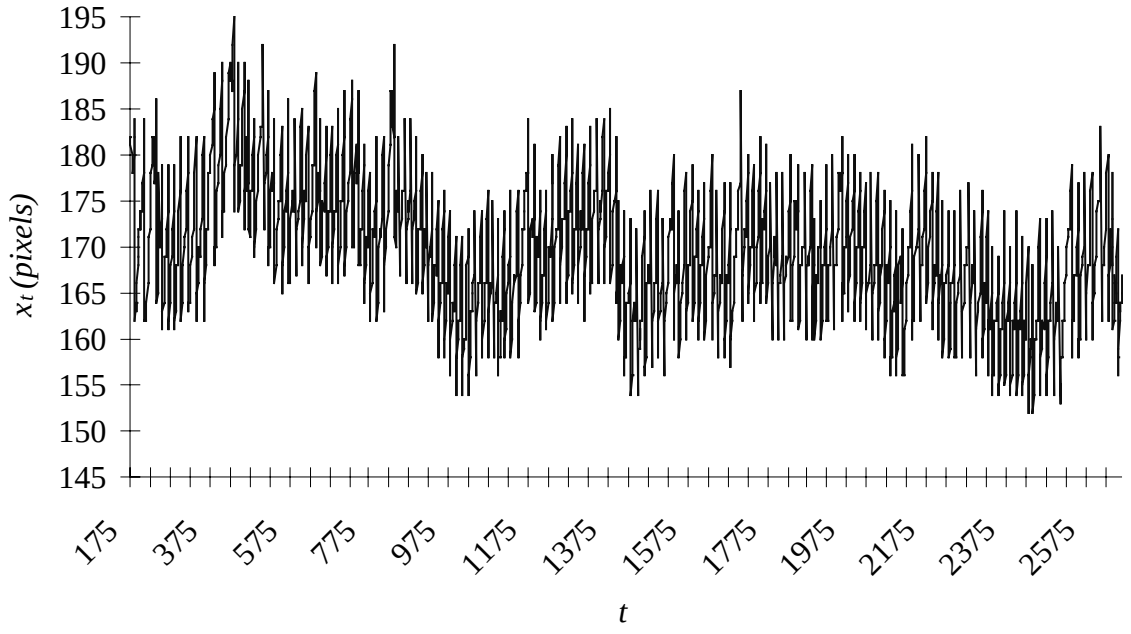

Figure 7.4 – Stickout Time Series (Observed)

The observed stickout time series *X* consists of the 2,492 equally sampled observations, at t = 175 through 2,666. Figure 7.4 illustrates all observations, while Figure 7.2 provides a detailed view of a sample of the time series.

Besides the obvious nonperiodic oscillations, the stickout time series exhibits a large-scale trend. As discussed in Chapter 6, removing trends helps the method find the necessary temporal patterns. A first difference filter could be applied, but that would introduce a new synchronization problem between the release and stickout time series. Instead, a simple recalibration rule is used to removing the trend. When there is a 10-pixel drop between two consecutive observations, the second observation is recalibrated to zero. Figure 7.5 and Figure 7.6 illustrate that the trend in stickout time series has been removed.

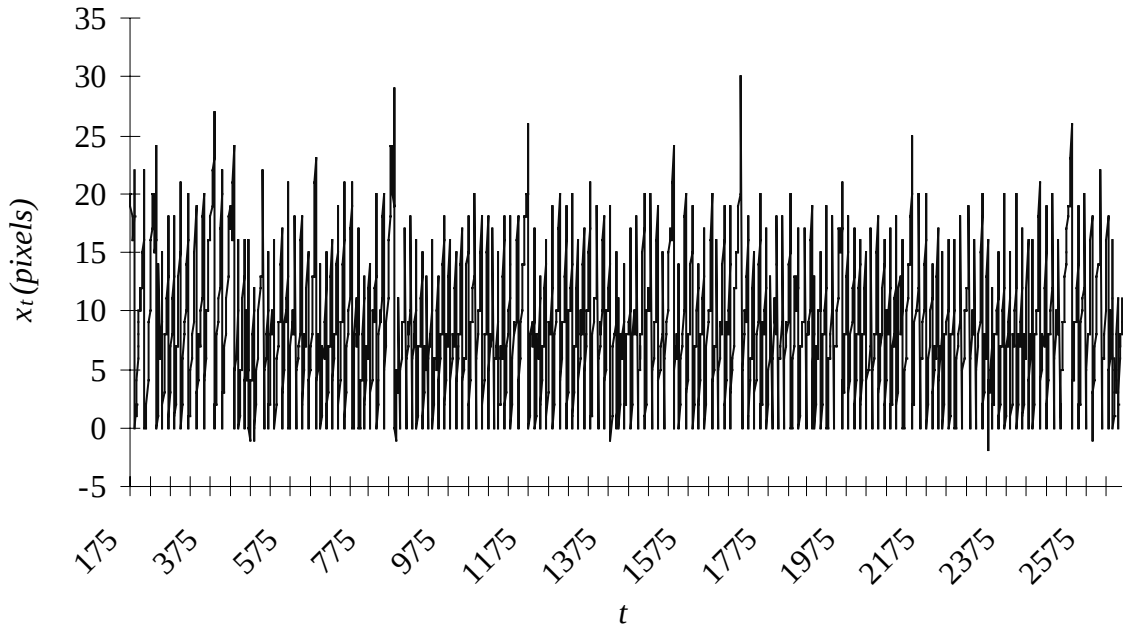

Figure 7.5 – Recalibrated Stickout Time Series (Observed)

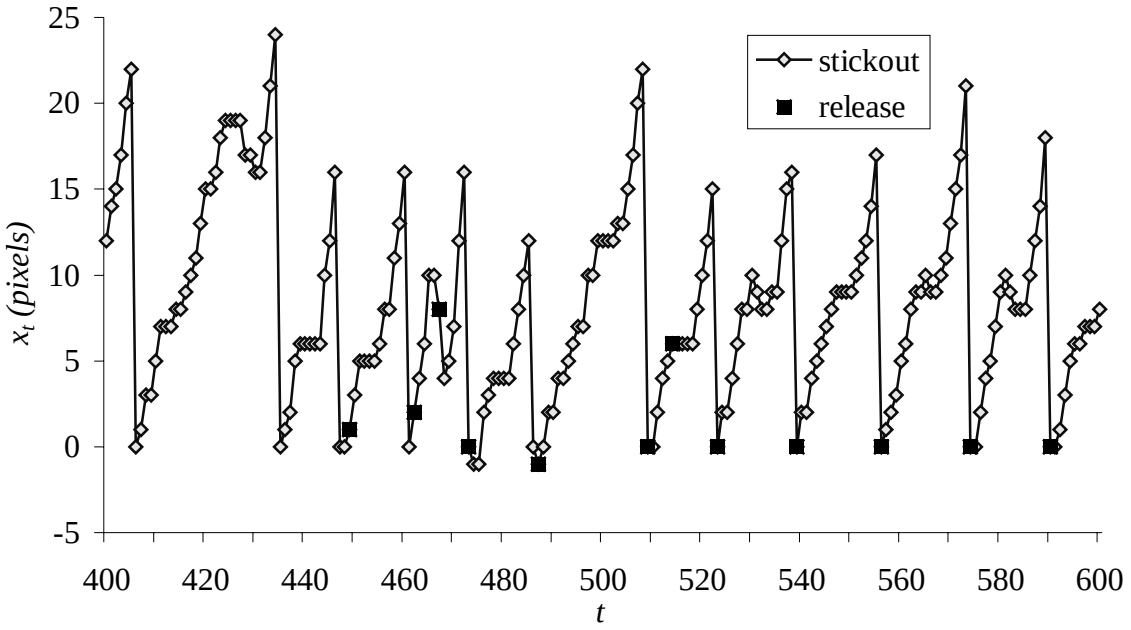

Figure 7.6 – Recalibrated Stickout and Release Time Series (Observed)

Instead of being contained within the stickout time series, the events are captured in the release time series *Y*, as illustrated in Figure 7.6. The release time series is defined as a binary sequence, where the ones indicate a release (event) and the zeros a non-release (non-event). The release usually occurs after a stickout value reaches a local peak and drops 10 pixels or more. However, a study of Figure 7.6 shows there are several times when this does not occur. In this section, the release time series will be used unaltered. In the next section, the release series will be recalculated to more correctly match the stickout length minimums.

Now that the observed time series have been presented, the TSDM goal is restated in terms of the objective and event characterization functions. The TSDM-S/M method requires two objective functions. The first objective function describes the objective for the final result. Introduced in Chapter 3,

$$f_1(\mathcal{C}) = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{7.1}$$

has an optimal value when every event is correctly predicted. The values $t_p, t_n, f_p,$ and f_n are described in Table 7.1.

	Actually an event	Actually a non-event
Categorized as an event	True positive, t_p	False positive, f_p
Categorized as a non-event	False negative, f_n	True negative, t_n

Table 7.1 – Event Categorization

The second objective function,

$$f_2(P) = \frac{t_p}{t_p + f_p}, \tag{7.2}$$

called the positive accuracy, defines how well each $P_i \in \mathcal{C}, i = 1, 2, \ldots$ is at avoiding false positives. It is used as the objective for the intermediate steps in the TSDM-S/M training stage.

The optimization formulation for the whole training stage is $\max f(\mathcal{C})$ subject to $\min c(\mathcal{C})$ and $\min b(\delta_i) \forall P_i \in \mathcal{C}$. The optimization formulation for the intermediate steps is $\max f(P)$ subject to $\min b(\delta)$.

Figure 7.7 presents an illustrative phase space, where the Manhattan or l_1 distance metric is employed. The phase space points are similar to the linearly increasing phase space points, but the increase repeats instead of continuing to grow.

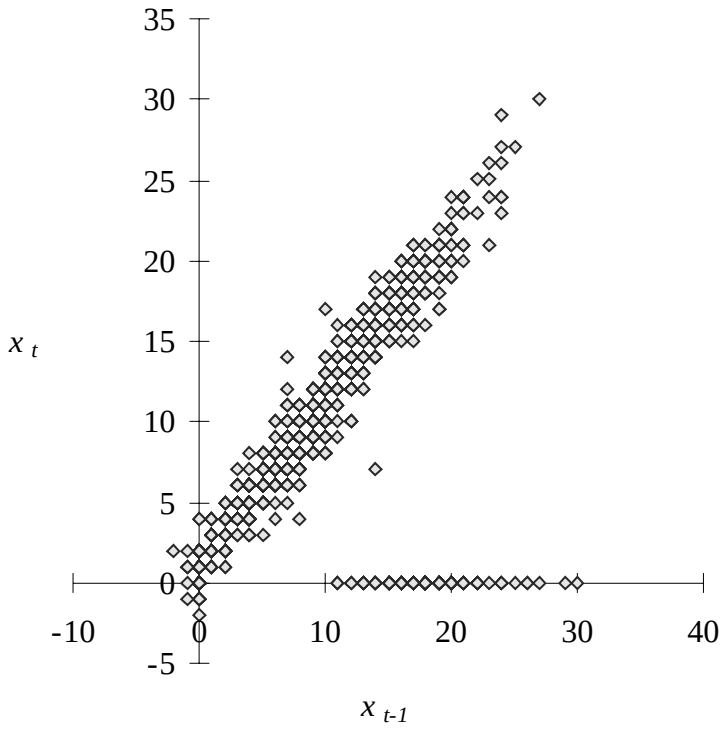

Figure 7.7 – Recalibrated Stickout Phase Space (Observed)

Figure 7.8 clearly shows the complexity of the augmented phase space. The events are not separable from the non-events using a two-dimensional phase space. Hence, the TSDM-S/M method, which finds multiple temporal clusters of varying dimensionality, is applied.

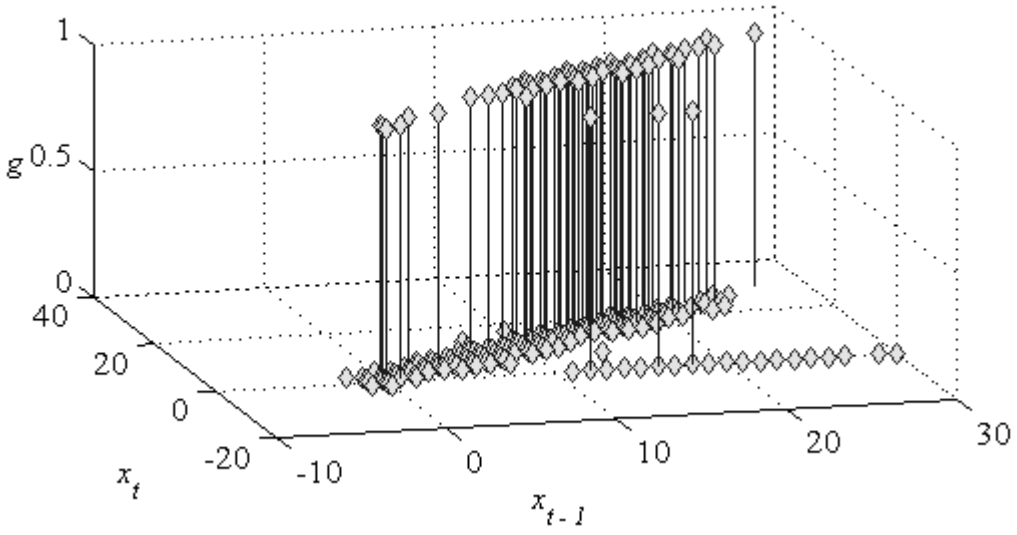

Figure 7.8 – Stickout and Release Augmented Phase Space (Observed)

The augmented phase space is searched using a tournament genetic algorithm. The two sets of search parameters are presented in Table 7.2.

Parameter	Set 1	Set 2
Random search multiplier	10	10
Population size	30	30
Elite count	1	1
Gene length	8	8
Tournament size	2	2
Mutation rate	0.05%	0%
Convergence criteria	0.65	0.5

Table 7.2 – Genetic Algorithm Parameters for Recalibrated Stickout and Release Time Series

The results of the search are shown in Table 7.3.

Result	Value
Temporal pattern cluster count, $c(\mathcal{C})$	14
Temporal pattern cluster dimensions	1~14
Clusters cardinality, $c(M)$	142
Clusters mean eventness, μ_M	0.71
Clusters standard deviation eventness, σ_M	0.45
Non-clusters cardinality, $c(\tilde{M})$	2,349
Non-clusters mean eventness, $\mu_{\tilde{M}}$	0.023
Non-clusters standard deviation eventness, $\sigma_{\tilde{M}}$	0.15
z_r	-49
α_r	0
z_m	18
α_m	$2.4\text{x}10^{-72}$
True positives, t_p	101
False positives, f_p	41
True negatives, t_n	2296
False negatives, f_n	53
Accuracy, $f_1(\mathcal{C})$	96.23%
Positive accuracy, $f_2(\mathcal{C})$	71.13%

Table 7.3 – Recalibrated Stickout and Release Results (Observed)

Fourteen temporal pattern clusters form the temporal pattern cluster collection employed to identify events. This collection contains temporal pattern clusters that vary in dimension from 1 to 14. The runs and *z* tests with $\alpha_r = 0$ and $\alpha_m = 2.4\text{x}10^{-72}$ show that

the two sets, clustered and non-clustered, are statistically different. However, for this problem the goal is to accurately predict droplet releases. The more meaningful statistics are the true/false positives/negatives. The statistics for accuracy indicate that 96.23% of the release observations are correctly characterized. The positive accuracy indicates that 71.13% of the release observations categorized as events are events.

The testing time series is shown in Figure 7.9 and Figure 7.10. The recalibrated stickout and release time series are shown in Figure 7.11 and Figure 7.12. The testing time series is transformed into the phase space as illustrated in Figure 7.13. The augmented phase space for the testing time series is seen in Figure 7.14.

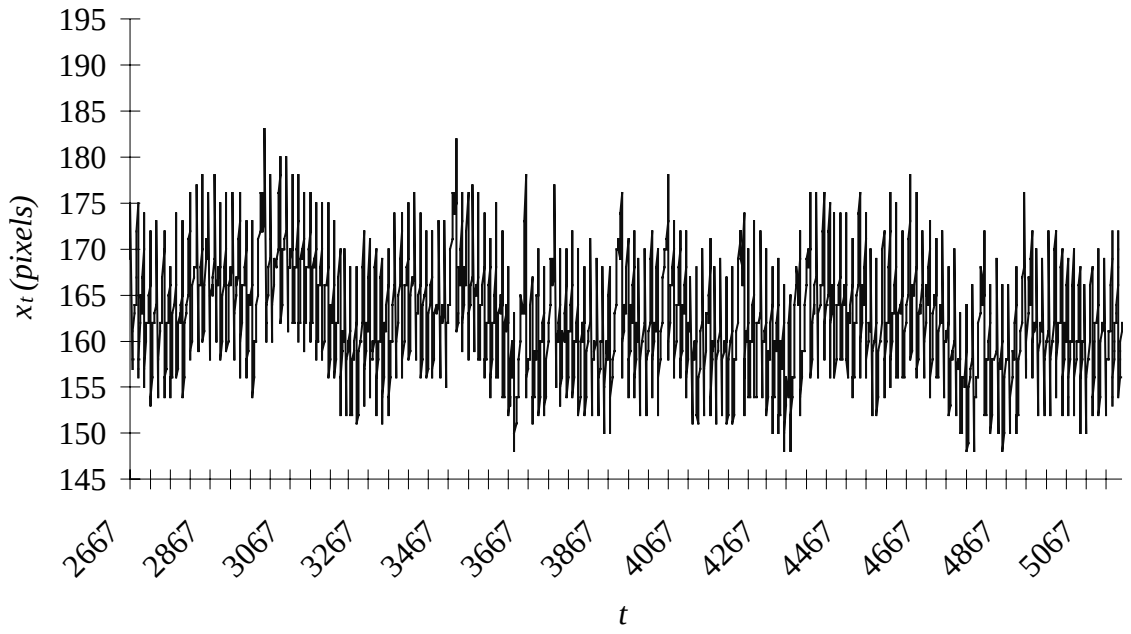

Figure 7.9 – Stickout Time Series (Testing)

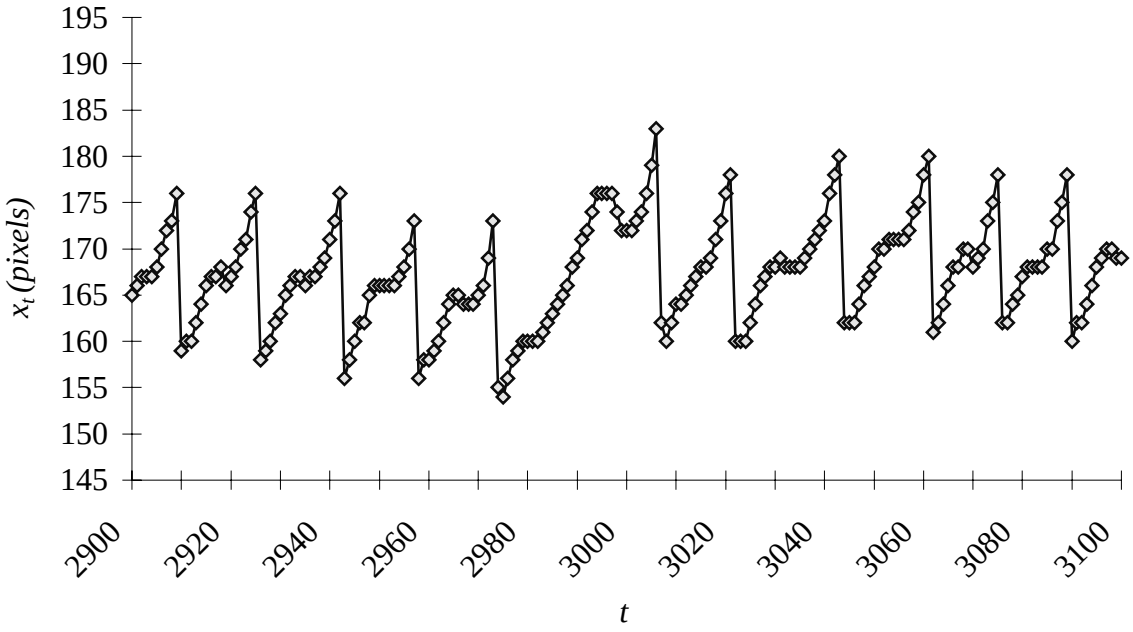

Figure 7.10 – Stickout Sample Time Series (Testing)

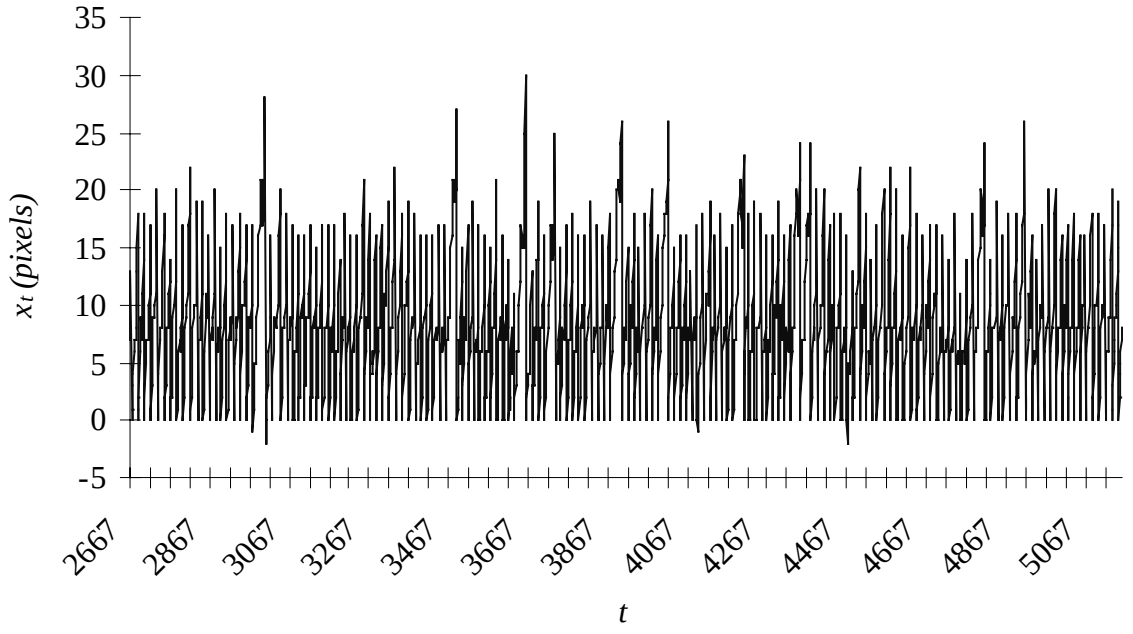

Figure 7.11 – Recalibrated Stickout Time Series (Testing)

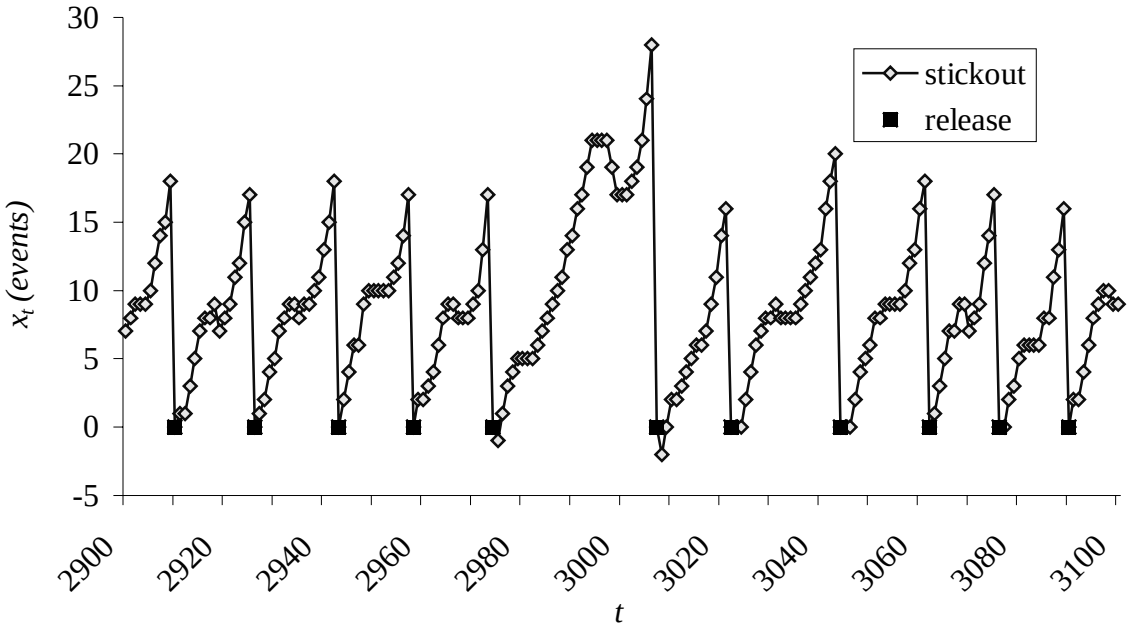

Figure 7.12 – Recalibrated Stickout and Release Time Series (Testing)

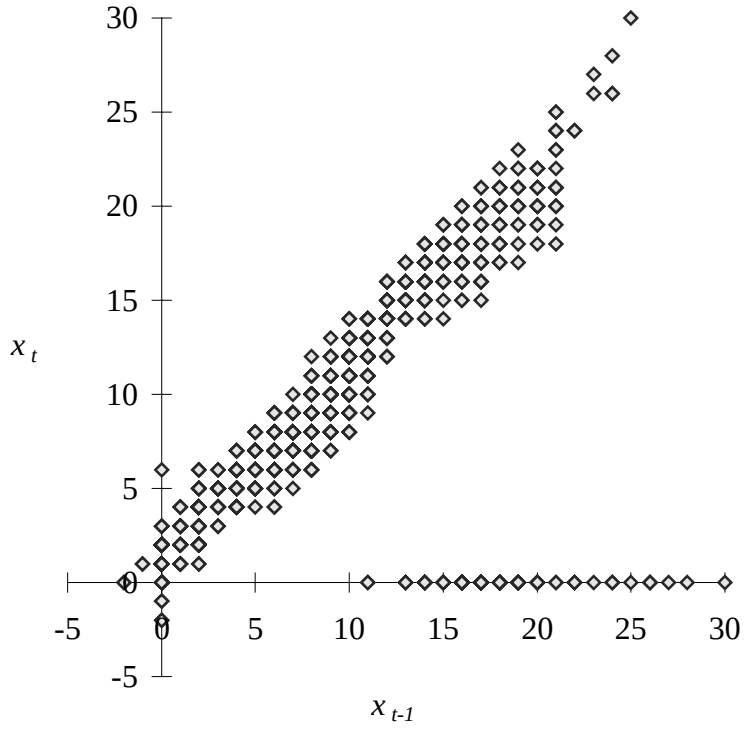

Figure 7.13 – Recalibrated Stickout Phase Space (Testing)

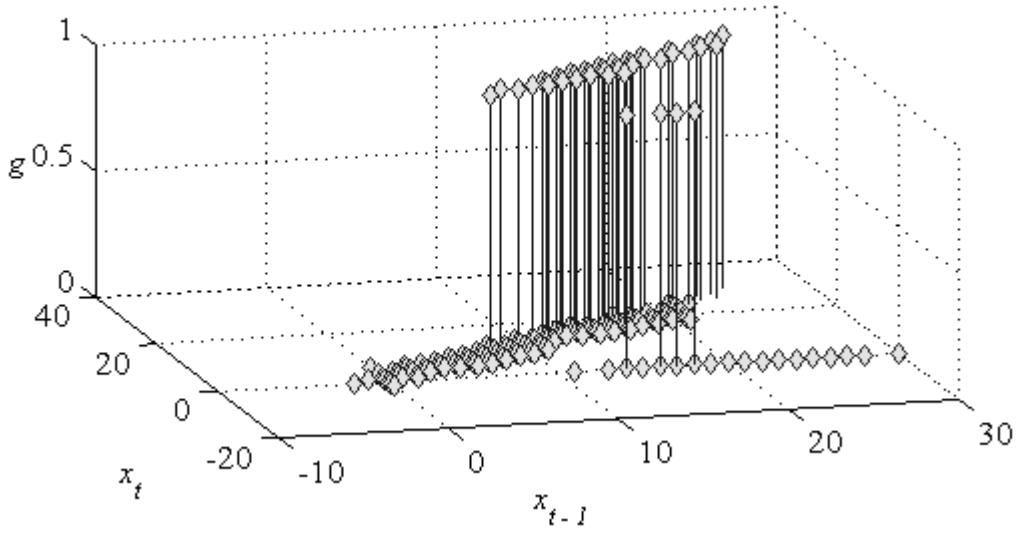

Figure 7.14 – Recalibrated Stickout and Release Augmented Phase Space (Testing)

The results of applying the temporal pattern cluster collection to the testing time series is seen in Table 7.4.

Result	Value
Clusters cardinality, $c(M)$	136
Clusters mean eventness, μ_M	0.74
Clusters standard deviation eventness, σ_M	0.44
Non-clusters cardinality, $c(\tilde{M})$	2,356
Non-clusters mean eventness, $\mu_{\tilde{M}}$	0.022
Non-clusters standard deviation eventness, $\sigma_{\tilde{M}}$	0.15
z_r	-49
α_r	0
z_m	19
α_m	4.0×10^{-78}
True positives, t_p	100

Result	Value
False positives, f_p	36
True negatives, t_n	2,303
False negatives, f_n	53
Accuracy, $f_1(\mathcal{C})$	96.43%
Positive accuracy, $f_2(\mathcal{C})$	73.53%

Table 7.4 – Recalibrated Stickout and Release Results (Testing)

As with the training stage results, the testing stage results are statistically significant as seen by both the runs and *z* tests. The α_r is zero, and the α_m is 4.0×10^{-78}. More importantly, the prediction accuracy is 96.43%, and the positive accuracy is 73.53%. These results are better than those found in the characterization phase. This is significant, especially considering that the data set provider deems the stickout measurements as "not too reliable".

7.2 Adjusted Release Characterization and Prediction Using Stickout

This section presents results using an adjusted release time series rather than the one computed using the INEEL process. As seen in Figure 7.6, the release time series does not always correspond with the stickout data. It also does not correspond with the voltage time series presented later in the chapter. The adjusted release time series is created using a simple rule – a release has occurred after a ten-pixel drop in the stickout time series. This rule is identifying events a posteriori, while the TSDM method is predicting events a priori. A sample of the adjusted release time series is shown in Figure 7.15.

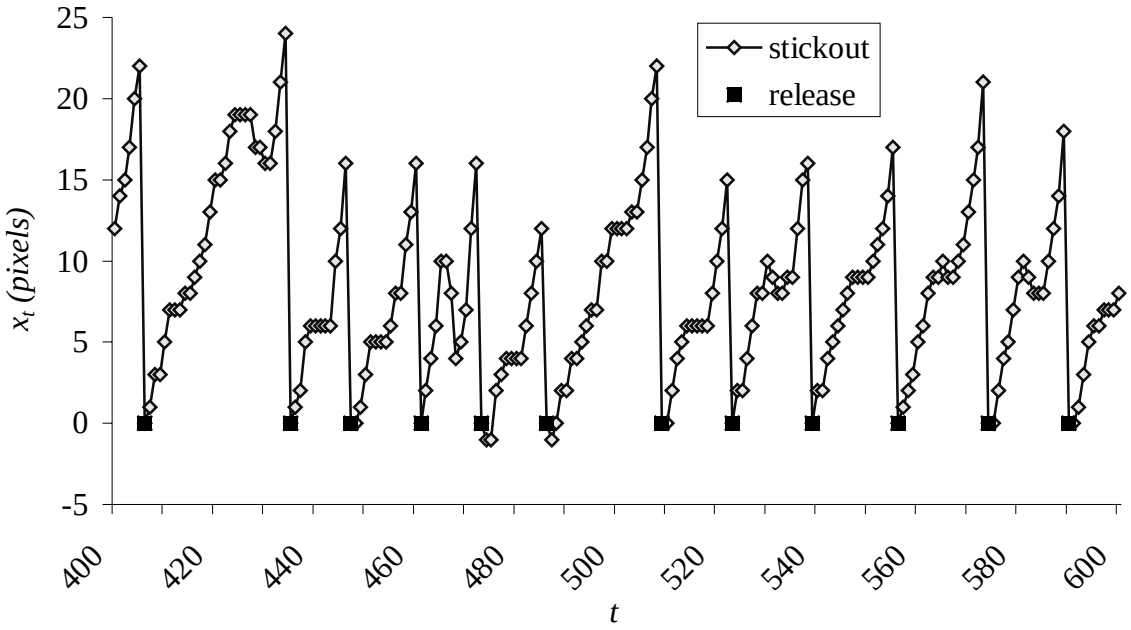

Figure 7.15 – Recalibrated Stickout and Adjusted Release Time Series (Observed)

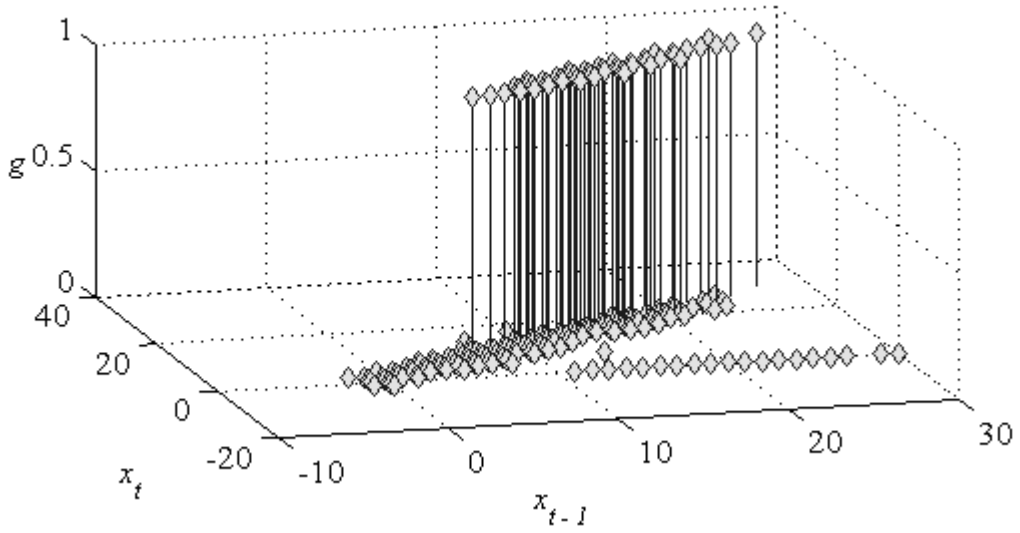

Figure 7.16 – Recalibrated Stickout and Adjusted Release Augmented Phase Space (Observed)

The TSDM goal, primary objective function, event characterization, and optimization formulation remain the same. An alternative secondary objective function,

$$f_3(P) = \begin{cases} -(f_n + t_n)^2 & \text{if } t_p = 0 \wedge f_p = 0 \\ t_p - (t_p + f_p + t_n + f_n) \cdot f_p & \text{otherwise} \end{cases}, \quad (7.3)$$

is introduced, which maximizes the number of true positives while penalizing any false positives.

The augmented phase space, illustrated by Figure 7.16, while still complex, is more orderly than the unadjusted release augmented phase space shown in Figure 7.8.

Five different sets of genetic algorithms parameters are used to find the temporal pattern clusters. For all sets, the elite count was one, the gene length was eight, and the tournament size was two. The other parameters are listed in Table 7.5.

	Random Search multiplier	Population size	Mutation rate	Convergence criteria	Secondary objective function
Set 1	10	30	0.2%	1	$f_3(P)$
Set 2	1	100	0.2%	1	$f_3(P)$
Set 3	1	100	0.02%	1	$f_3(P)$
Set 4	10	30	0%	0.5	$f_2(P)$
Set 5	10	30	0.05%	0.65	$f_2(P)$
Set 6	10	30	0.05%	0.5	$f_2(P)$

Table 7.5 – Genetic Algorithm Parameters for Recalibrated Stickout and Adjusted Release Time Series

The training stage results are shown in Table 7.6.

Result	Value
Temporal pattern cluster count, $c(\mathcal{C})$	67
Temporal pattern cluster dimensions	1~14
Clusters cardinality, $c(M)$	138

Result	Value
Clusters mean eventness, μ_M	0.81
Clusters standard deviation eventness, σ_M	0.39
Non-clusters cardinality, $c(\tilde{M})$	2,353
Non-clusters mean eventness, $\mu_{\tilde{M}}$	0.017
Non-clusters standard deviation eventness, $\sigma_{\tilde{M}}$	0.13
z_r	-49
α_r	0
z_m	24
α_m	$2.9\text{x}10^{-124}$
True positives, t_p	112
False positives, f_p	26
True negatives, t_n	2,313
False negatives, f_n	40
Accuracy, $f_1(\mathcal{C})$	97.35%
Positive accuracy, $f_2(\mathcal{C})$	81.16%

Table 7.6 – Recalibrated Stickout and Adjusted Release Results (Observed)

Sixty-seven temporal pattern clusters form the temporal pattern cluster collection used to identify the events. The statistical tests with $\alpha_r = 0$ and $\alpha_m = 2.9\text{x}10^{-124}$ show that the two sets, clustered and non-clustered, are statistically different. The accuracy statistic indicates that 97.35% (vs. 96.23% using the unadjusted release time series) of the release observations are correctly characterized. The positive accuracy indicates that 81.16% (vs.

71.13% using the unadjusted release time series) of the release observations categorized as events are events.

The testing stage time series is shown in Figure 7.17. The augmented phase space for the testing time series is illustrated in Figure 7.18.

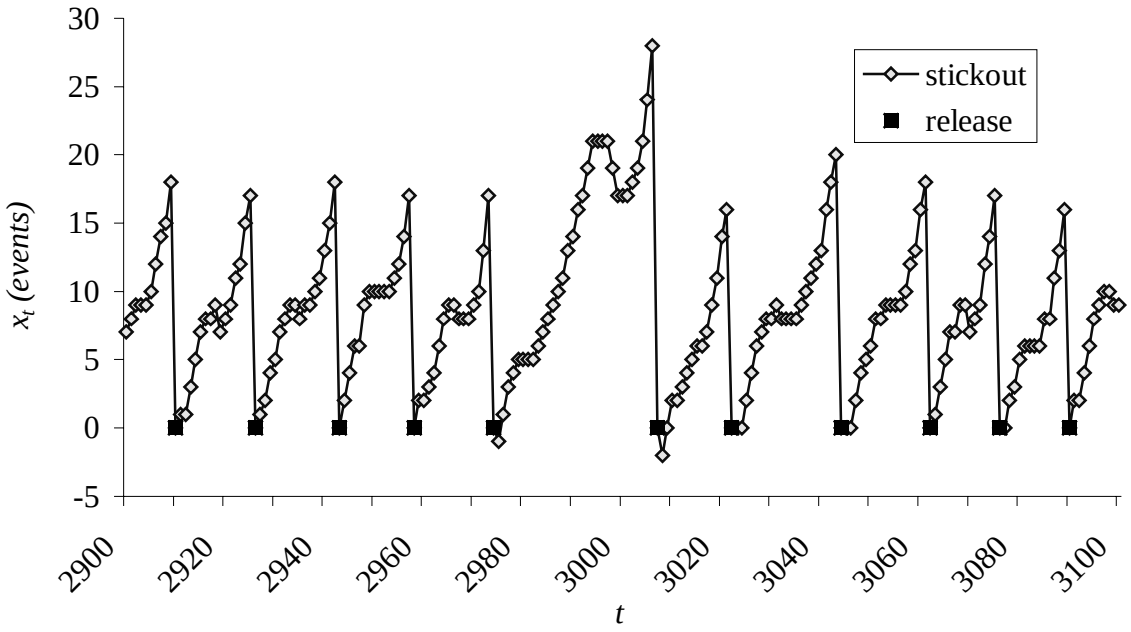

Figure 7.17 – Recalibrated Stickout and Adjusted Release Time Series (Testing)

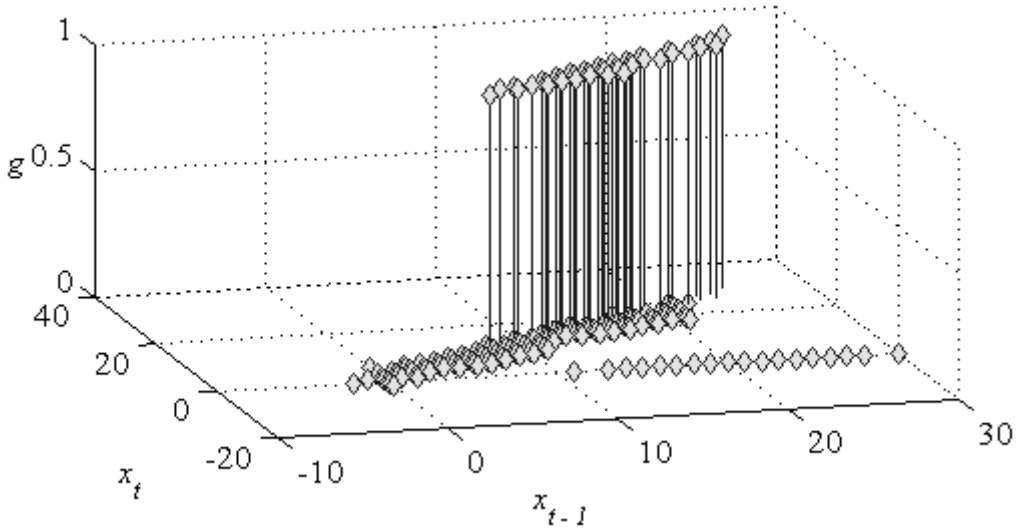

Figure 7.18 – Recalibrated Stickout and Adjusted Release Augmented Phase Space (Testing)

The testing stage results are presented in Table 7.7.

Result	Value
Clusters cardinality, $c(M)$	161
Clusters mean eventness, μ_M	0.70
Clusters standard deviation eventness, σ_M	0.46
Non-clusters cardinality, $c(\tilde{M})$	2,331
Non-clusters mean eventness, $\mu_{\tilde{M}}$	0.017
Non-clusters standard deviation eventness, $\sigma_{\tilde{M}}$	0.13
z_r	-49
α_r	0
z_m	19
α_m	$1.63 \text{x} 10^{-79}$
True positives, t_p	113
False positives, f_p	48
True negatives, t_n	2,291
False negatives, f_n	40
Accuracy, $f_1(\mathcal{C})$	96.47%
Positive accuracy, $f_2(\mathcal{C})$	70.19%

Table 7.7 – Recalibrated Stickout and Adjusted Stickout Results (Testing)

As with the training stage results, the testing stage results are statistically significant as seen by both the runs and *z* tests. The $\alpha_r = 0$, and the $\alpha_m = 1.63x10^{-79}$. The prediction accuracy is 96.47% (vs. 96.43% with the unadjusted release time series) and the positive accuracy is 70.19% (vs. 73.53% with the unadjusted release time series).

According to the total prediction accuracy, the recalibrated stickout and adjusted stickout results are better. Whereas according to the positive prediction accuracy, the unadjusted release time series results are better.

7.3 Stickout, Release, Current and Voltage Synchronization

The last two sections focused on using the stickout time series temporal patterns for characterization and prediction of droplet releases. The TSDM-S/M method has yielded excellent results. The next step is to use the current and voltage time series to help characterize and predict droplet releases. Unfortunately, the stickout and release time series are not synchronized with the current and voltage time series. This leaves two problems to be solved. The first is to synchronize the four time series. The second is to compensate for the different sampling rates.

The synchronization is done by matching the first and last voltage peaks with the first and last droplet releases. For the voltage time series, these observations are 973 and 25764. For the droplet release time series, these observations are 187 and 5151.

Recall that the stickout and release time series sampling rate was reported to be 1kHz and the current and voltage sampling-rate was reported to be 5kHz. If these sampling rates are perfectly calibrated, the 1kHz time series could be up-sample to the 5kHz rate by interpolating four additional points for each observation or down-sampling the 5kHz time series by averaging five observations into one observation. However, when this is done, the time series lose synchronization.

The initial synchronization was done using the first voltage spike and the first droplet release. Using the reported five-to-one sampling ratio and the last droplet release

observation of 5151, the last voltage spike should be observation 25,793. It is actually observation 25,764, which is determined by visualizing the data. The true sampling rates are not exactly in a 5:1 ratio.

The problem is solved using Matlab's *interp1* [57, pp5.9-5.11] function with the cubic spline option. This function allows conversion between arbitrary sampling rates by providing the initial time series with its sampling times and by specifying a vector with the desired sampling times. The function performs interpolation using a cubic spline. It may be used for either up-sampling or down-sampling. Both the up-sampling to 5kHz and down sampling to 1kHz time series were generated by appropriately mapping the first and last synchronization observations onto each other.

7.4 Adjusted Release Characterization and Prediction Using Stickout, Voltage, and Current

With the synchronization problem solved, the TSDM-M/M method is applied to the voltage, current, and stickout time series to characterize and predict droplet releases. The adjusted release time series is used as the indicator of events. The time series are normalized to the range [0,1], using the transformation

$$Z = \frac{X - \min(X)}{\max(X - \min(X))}. \tag{7.4}$$

A sample of the observed time series is shown in Figure 7.19.

The TSDM goal, primary objective function, event characterization, and optimization formulation remain the same. An alternative secondary objective function,

$$f_4(P) = \frac{\mu_M - \mu_{\tilde{M}}}{\sqrt{\frac{\sigma_M^2}{c(M)} + \frac{\sigma_{\tilde{M}}^2}{c(\tilde{M})}}} \tag{7.5}$$

also is used.

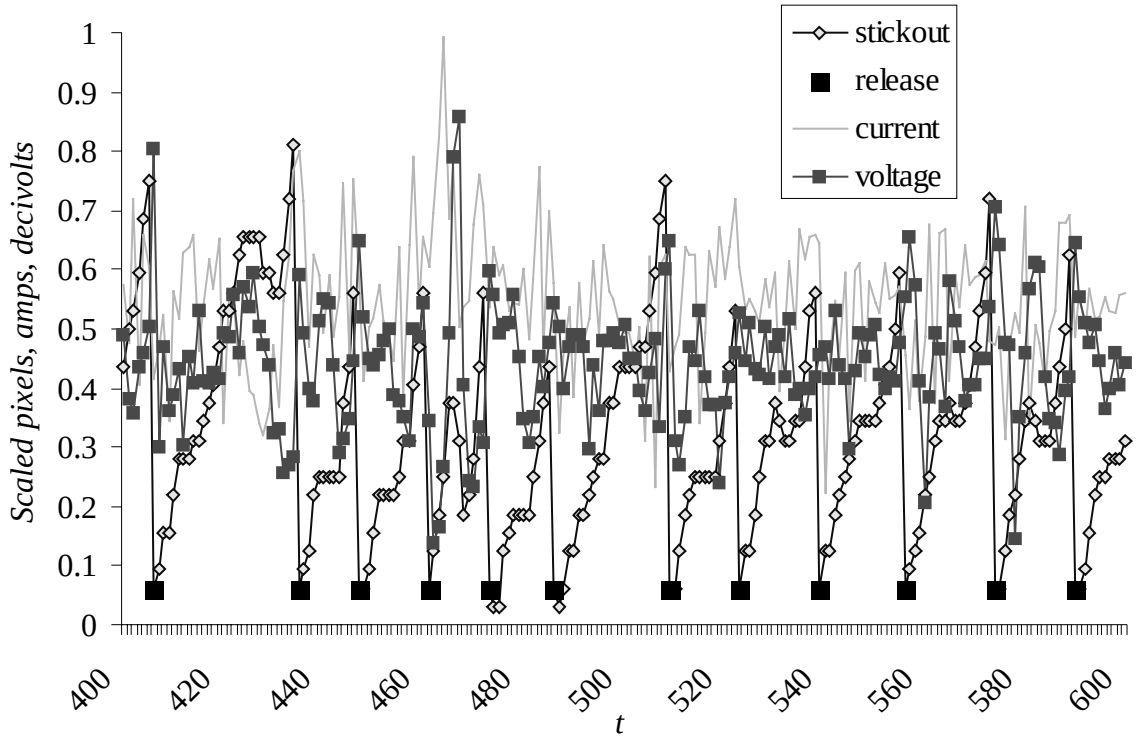

Figure 7.19 – Recalibrated Stickout, Current, Voltage, and Adjusted Release Time Series (Observed)

Because the smallest phase space that can be formed using all the time series is three-dimensional, and the corresponding augmented phase space is four-dimensional, graphical illustrations are not possible. Nonetheless, these spaces are formed and searched using a tournament genetic algorithm. The set of genetic algorithm search parameters is presented in Table 7.5. Three different sets of genetic algorithm parameters were used to find all the temporal pattern clusters. For all parameter sets, the elite count was one, the gene length was eight, the tournament size was two, and mutation rate was 0.2%. The other parameters by set are listed in Table 7.8.

	Random Search multiplier	Population size	Convergence criteria	Secondary objective function
Set 1	10	30	0.75	$f_2(P)$
Set 2	1	30	1	$f_3(P)$
Set 3	1	10	1	$f_4(P)$

Table 7.8 – Genetic Algorithm Parameters for Recalibrated Stickout, Current, Voltage, and Adjusted Release Time Series

The training stage results are shown in Table 7.9.

Result	Value
Temporal pattern cluster count, $c(\mathcal{C})$	62
Temporal pattern cluster dimensions	3~15
Clusters cardinality, $c(M)$	117
Clusters mean eventness, μ_M	0.89
Clusters standard deviation eventness, σ_M	0.32
Non-clusters cardinality, $c(\tilde{M})$	2,374
Non-clusters mean eventness, $\mu_{\tilde{M}}$	0.020
Non-clusters standard deviation eventness, $\sigma_{\tilde{M}}$	0.14
z_r	-49
α_r	0
z_m	30
α_m	7.1×10^{-193}
True positives, t_p	104
False positives, f_p	13
True negatives, t_n	2,326

Result	Value
False negatives, f_n	48
Accuracy, $f_1(\mathcal{C})$	97.55%
Positive accuracy, $f_2(\mathcal{C})$	88.89%

Table 7.9 – Recalibrated Stickout, Current, Voltage, and Adjusted Release Results (Observed)

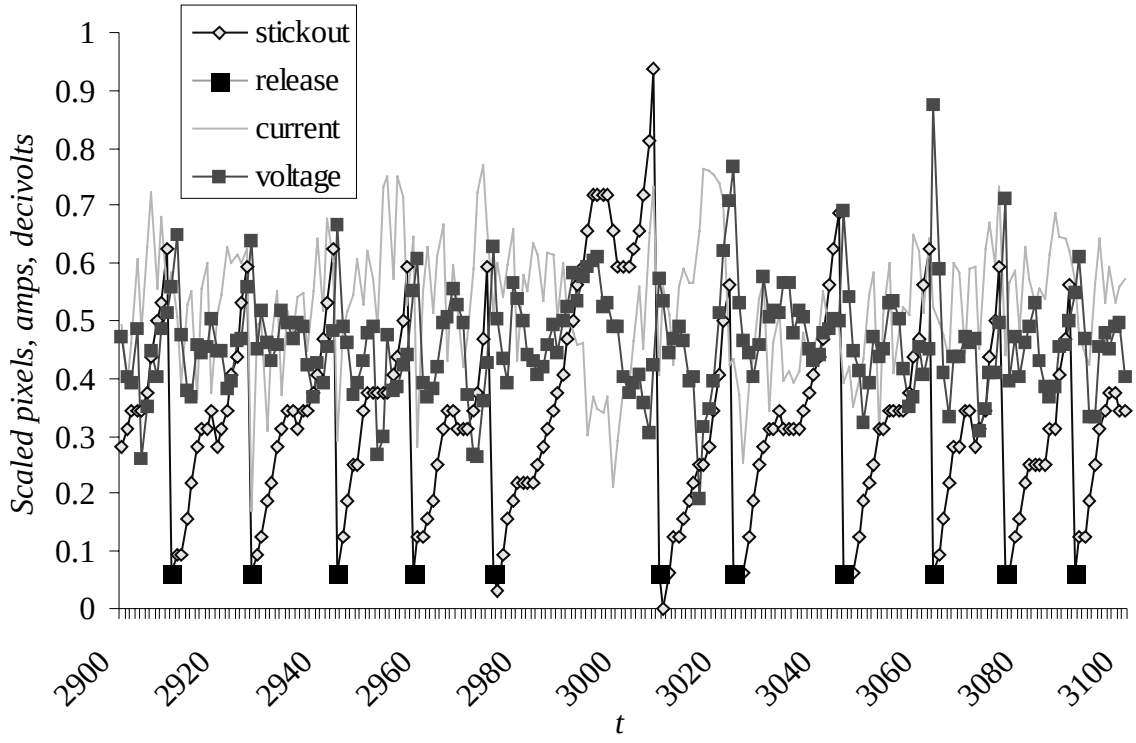

Figure 7.20 – Recalibrated Stickout, Current, Voltage, and Adjusted Release Time Series (Testing)

Sixty-two temporal pattern clusters form the collection of temporal pattern clusters used to identify the events. This collection contains temporal pattern clusters that vary in dimension from 3 to 15. The runs and *z* tests with $\alpha_r = 0$ and $\alpha_m = 7.1\text{x}10^{-193}$ show that the two sets, clustered and non-clustered, are statistically different. The accuracy statistic indicates that 97.55% (vs. 97.35% using just the stickout and the

adjusted release time series and vs. 96.23% using the stickout and unadjusted release time series) of the release observations are correctly characterized. The positive accuracy indicates that 88.89% (vs. 81.16% using just the stickout and the adjusted release time series and vs. 71.13% using the stickout and unadjusted release time series) of the release observations categorized as events are events.

The testing stage time series is illustrated in Figure 7.20 and results in Table 7.7.

Result	Value
Clusters cardinality, $c(M)$	117
Clusters mean eventness, μ_M	0.67
Clusters standard deviation eventness, σ_M	0.47
Non-clusters cardinality, $c(\tilde{M})$	2,375
Non-clusters mean eventness, $\mu_{\tilde{M}}$	0.032
Non-clusters standard deviation eventness, $\sigma_{\tilde{M}}$	0.17
z_r	-49
α_r	0
z_m	14
α_m	2.1×10^{-47}
True positives, t_p	78
False positives, f_p	39
True negatives, t_n	2,300

Result	Value
False negatives, f_n	75
Accuracy, $f_1(\mathcal{C})$	95.42%
Positive accuracy, $f_2(\mathcal{C})$	66.67%

Table 7.10 – Recalibrated Stickout, Current, Voltage, and Adjusted Release Results (Testing)

As with the training stage, the testing stage results are statistically significant as seen by both the runs and *z* tests. The $\alpha_r = 0$ and $\alpha_m = 2.1\text{x}10^{-41}$. More importantly, the prediction accuracy is 95.42% (vs. 96.47% using just the stickout and the adjusted release time series and vs. 96.43% using the stickout and unadjusted release time series) and the positive accuracy is 66.67% (vs. 70.19% using just the stickout and the adjusted release time series and vs. 73.53% using the stickout and unadjusted release time series).

The prediction results using the stickout, current, and voltage time series are not as good as using just the stickout time series. There are two possible explanations for this. Recall that the training stage results using all three time series were better than the training results using just the stickout time series. In addition, the search space is be higher dimensional and therefore sparser, because the multi-dimensional time series embeds to a higher dimensional phase space. This suggests that the training stage over-fit the temporal pattern clusters to the training stage observations, i.e., the temporal pattern clusters discovered in the training stage are too specific to the training stage time series. The second explanation is that the recalibration process has introduced noise causing the testing results to be worse.

7.5 Conclusion

Using from one to three time series generated from sensors on a welding station, the problem of predicting when a droplet of metal will release from the welder was solved with a high degree of accuracy – from 95.42% to 96.47% total prediction accuracy and from 66.67% to 73.53% positive prediction accuracy. These results show that the TSDM method could be used in a system to control and monitor the welding seam thereby improving the quality of the weld.

The next chapter applies the TSDM methods to the financial domain.

Chapter 8 Financial Applications of Time Series Data Mining

This chapter, organized into four sections, presents significant results found by applying the Time Series Data Mining (TSDM) method to financial time series. The first section discusses the definition of events for this application and the generation of the time series. The second and third sections present the results of applying the $TSDMe_1$-S/S and $TSDMe_1$-M/S methods to a financial time series. The final section applies the $TSDMe_2$-S/S method to a collection of time series.

In this chapter, the analyzed time series are neither synthetically generated as in Chapter 5, nor measured from a physical system as in Chapter 7. Instead, they are created by the dynamic interaction of millions of investors buying and selling securities through a secondary equity market such as the New York Stock Exchange (NYSE) or National Association of Securities Dealers Automated Quotation (NASDAQ) market [58]. The times series are measurements of the activity of a security, specifically a stock. The time series are the daily open price, which is the price of the first trade, and the daily volume, which is the total number of shares of the stock traded.

Before applying the TSDM framework to security price prediction, an explanation of the underlying structure of security price behavior is required, i.e., the efficient market hypothesis. The efficient market hypothesis is described using the expected return or fair game model, which puts the efficient market hypothesis on firmer theoretical grounds than using the random walk hypothesis [58, p. 210]. The expected value of a security is

$$E(P_{t+1}|\Phi_t) = [1 + E(r_{t+1}|\Phi_t)]P_t \quad \text{[58, p. 210]}, \tag{8.1}$$

where P_t is the price of a security at time *t*, r_{t+1} is the one-period percent rate of return for the security during period *t*+1, and Φ_t is the information assumed to be fully reflected in the security price at time *t*.

There are three forms of the efficient market hypothesis. The weak form assumes Φ_t is all security-market information, such as historical sequence of price, rates of return, and trading volume data [58, p. 211]. The semistrong form assumes Φ_t is all public information, which is a super set of all security-market information, including earnings and dividend announcements, price-to-earning ratios, and economic and political news [58, p. 211]. The strong form assumes Φ_t is all public and private information, also including restricted data such as company insider information [58, p. 212].

The weak form of the efficient market hypothesis, which has been supported in the literature, applies to the current chapter. The efficient market hypothesis is verified by showing that security price time series show no autocorrelation and are random according to the runs test. In addition, tests of trading rules have generally shown that the weak form of the efficient market hypothesis holds [58, p. 213-215].

The TSDM goal is to find a trading-edge, a small advantage that allows greater than expected returns to be realized. If the weak form of the efficient market hypothesis holds, the TSDM methods should not be able to find any temporal patterns that can be exploited to achieve such a trading-edge. The TSDM goal is to find temporal pattern clusters that are, on average, characteristic and predictive of a larger than normal increase in the price of a stock.

8.1 ICN Time Series Using Open Price

This section presents the results of applying the $TSDMe_1$-S/M method to characterizing and predicting the change in the open price of ICN, a NASDAQ traded stock. ICN is an international pharmaceutical company. Two periods, 1990 and 1991, are analyzed. The first half of 1990 will be used as the observed time series and the second half as the testing time series. The 1991 time series will be similarly divided.

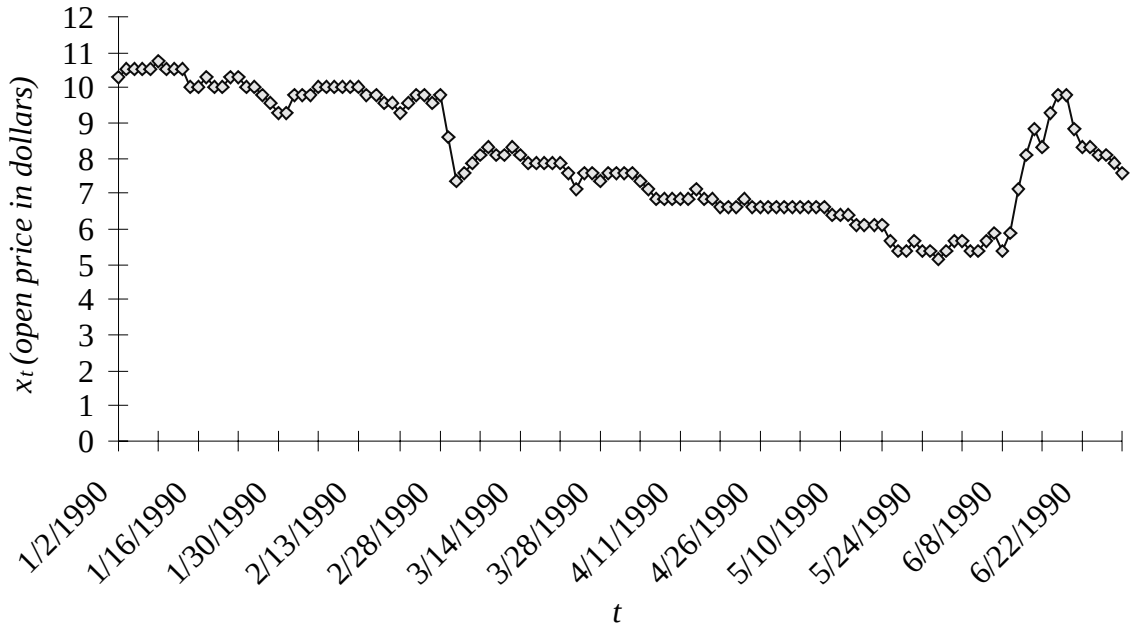

Figure 8.1 – ICN 1990H1 Daily Open Price Time Series (Observed)

8.1.1 ICN 1990 Time Series Using Open Price

The Figure 8.1 illustrates the observed time series *X*, which is the ICN open price for the first half of 1990 (1990H1). To identify temporal patterns that are both characteristic and predictive of events, a filter is needed. The $\Delta^{\%}$ filter converts the time series into a percentage change open price time series. The filtered time series has a more consistent range, as seen in Figure 8.2, facilitating the discovery of temporal pattern clusters.

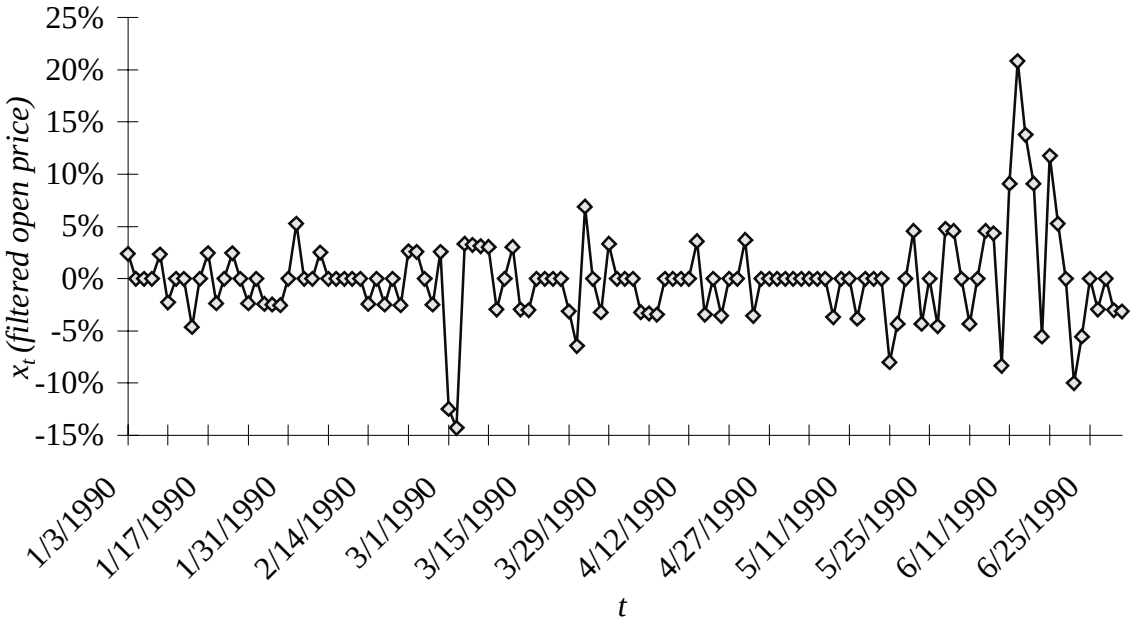

Figure 8.2 – Filtered ICN 1990H1 Daily Open Price Time Series (Observed)

The TSDM goal of finding a trading-edge is restated in terms of TSDM concepts. The objective function is

$$f(P) = \begin{cases} \mu_M & \text{if } c(M)/c(\Lambda) \geq \beta \\ (\mu_M - g_0)\dfrac{c(M)}{\beta c(\Lambda)} + g_0 & \text{otherwise} \end{cases}, \qquad (8.2)$$

where $\beta = 0.05$. The event characterization function is $g(t) = x_{t+1}$, which allows for one-step-ahead characterization and prediction. The optimization formulation is $\max f(P)$ subject to $\min b(\delta)$.

Figure 8.3 presents an illustrative phase space for the filtered ICN 1990H1 daily open price time series with a Euclidean distance metric. Figure 8.4 shows the augmented phase space.

The complexity of the embedding as illustrated in Figure 8.4. Clearly, the identification of a temporal pattern cluster that separates events from non-events is not possible. This will not prevent the TSDM goal of finding a trading-edge, though. The

goal is to find temporal pattern clusters that have higher objective function values and are statistically different from the phase space points outside the temporal pattern clusters.

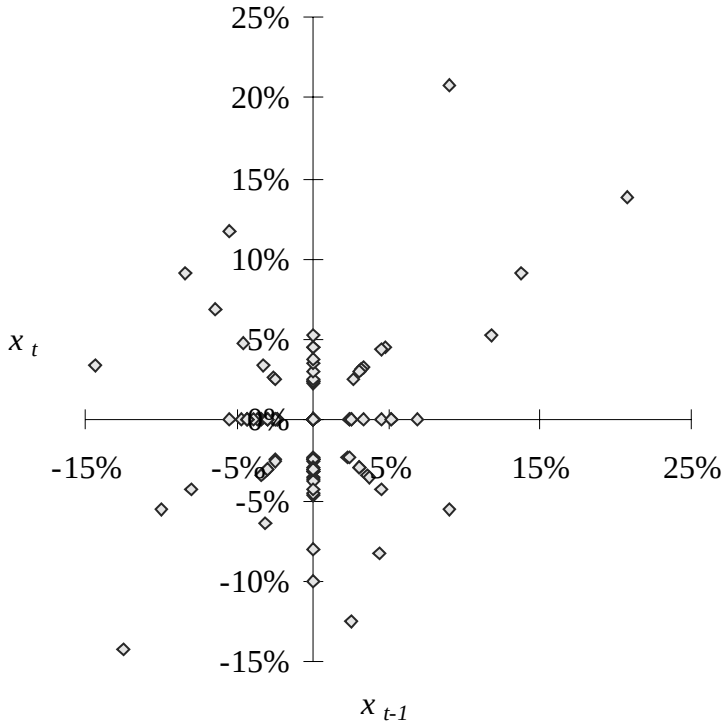

Figure 8.3 – Filtered ICN 1990H1 Daily Open Price Phase Space (Observed)

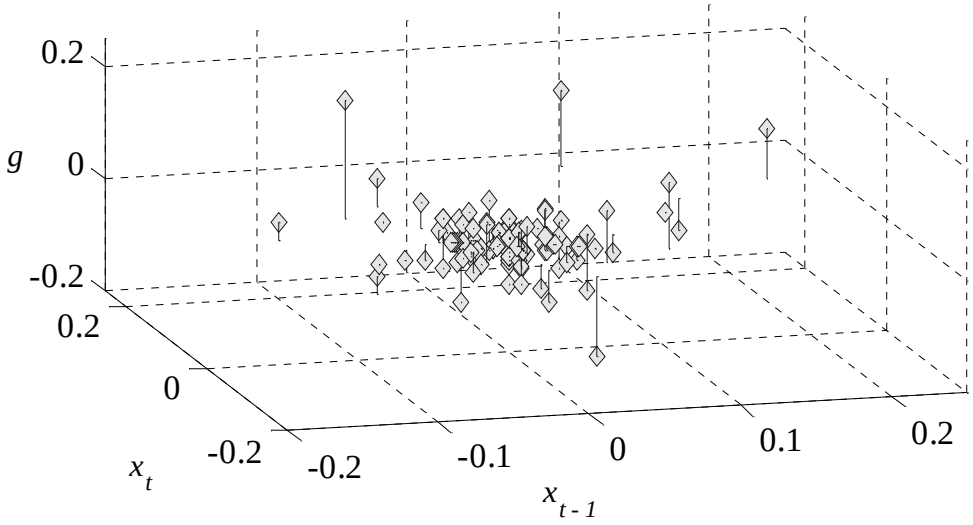

Figure 8.4 – Augmented Phase Space of Filtered ICN 1990H1 Daily Open Price (Observed)

The genetic algorithm search parameters are presented in Table 8.1.

Parameter	Values
Random search multiplier	10
Population size	30
Elite count	1
Gene length	6
Tournament size	2
Mutation rate	0 %
Convergence criteria	1

Table 8.1 – Genetic Algorithm Parameters for Filtered ICN 1990H1 Daily Open Price Time Series

The training stage results are shown in Table 8.2.

Result	Set 1	Set 2	Set 3	Combined Set
Temporal pattern cluster count, $c(\mathcal{C})$	1	1	1	3
Temporal pattern cluster dimensions	1	3	5	1,3,5
Clusters cardinality, $c(M)$	8	10	7	19
Clusters mean eventness, μ_M	5.43%	3.50%	6.49%	3.37%
Clusters standard deviation eventness, σ_M	8.70%	6.95%	7.47%	6.60%
Non-clusters cardinality, $c(\tilde{M})$	116	112	113	105
Non-clusters mean eventness, $\mu_{\tilde{M}}$	-0.56%	-0.50%	-0.61%	-0.81
Non-clusters standard deviation eventness, $\sigma_{\tilde{M}}$	3.60%	3.92%	3.80%	3.43%
z_r	-4.58	-2.07	-1.88	-4.61

Result	Set 1	Set 2	Set 3	Combined Set
α_r	4.71×10^{-6}	3.84×10^{-2}	6.02×10^{-2}	3.95×10^{-6}
z_m	1.94	1.79	2.50	2.70
α_m	5.30×10^{-2}	7.28×10^{-2}	1.26×10^{-2}	6.93×10^{-3}

Table 8.2 – Filtered ICN 1990H1 Daily Open Price Results (Observed)

In each case, the cluster mean eventness is greater than the non-cluster mean eventness. However, because of the limited training set size, the probability of a Type I error – incorrectly rejecting the null hypothesis that the two sets are the same – is higher than in the previous chapters. By combining the sets, the statistical significance is increased. This type of financial time series is nonstationary on all of the levels defined in this dissertation: stochastic, deterministic, and chaotic. The patterns persist for a short time period. This causes problems in achieving the desired 0.05 significance level.

The testing time series and the filtered testing time series are shown in Figure 8.5 and Figure 8.6, respectively. Figure 8.7 illustrates the testing phase space. The augmented phase space is seen in Figure 8.8.

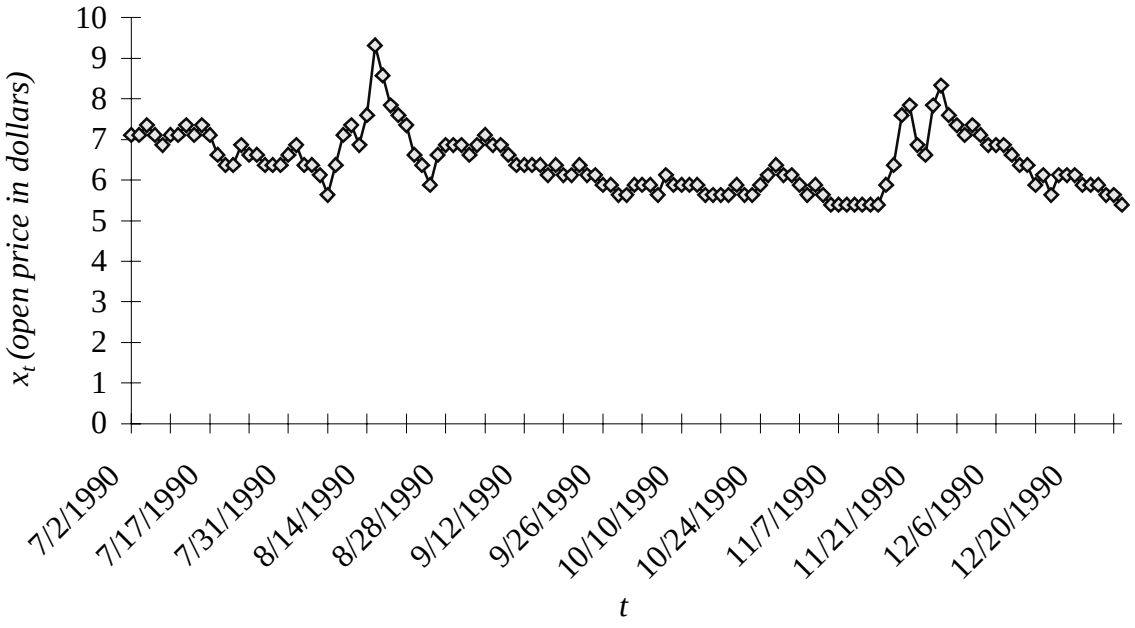

Figure 8.5 – ICN 1990H2 Daily Open Price Time Series (Testing)

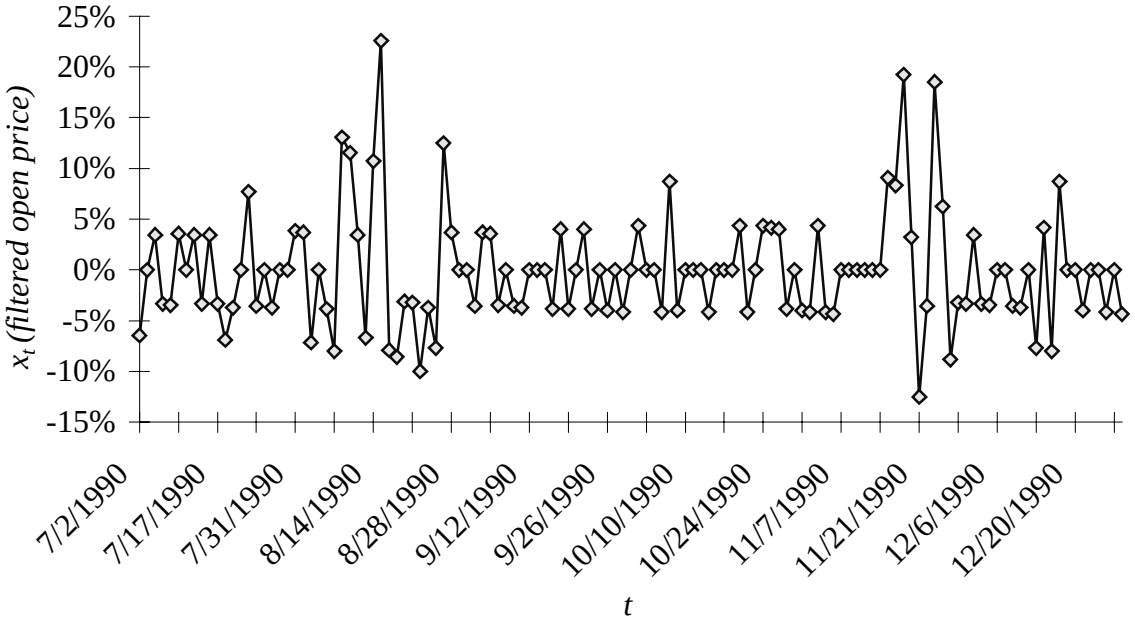

Figure 8.6 – Filtered ICN 1990H2 Daily Open Price Time Series (Testing)

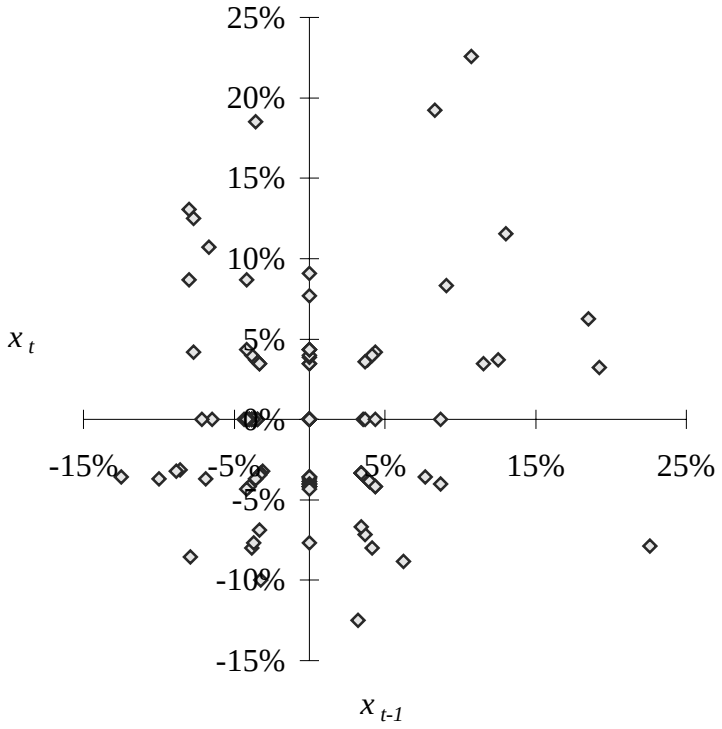

Figure 8.7 – Filtered ICN 1990H2 Daily Open Price Phase Space (Testing)

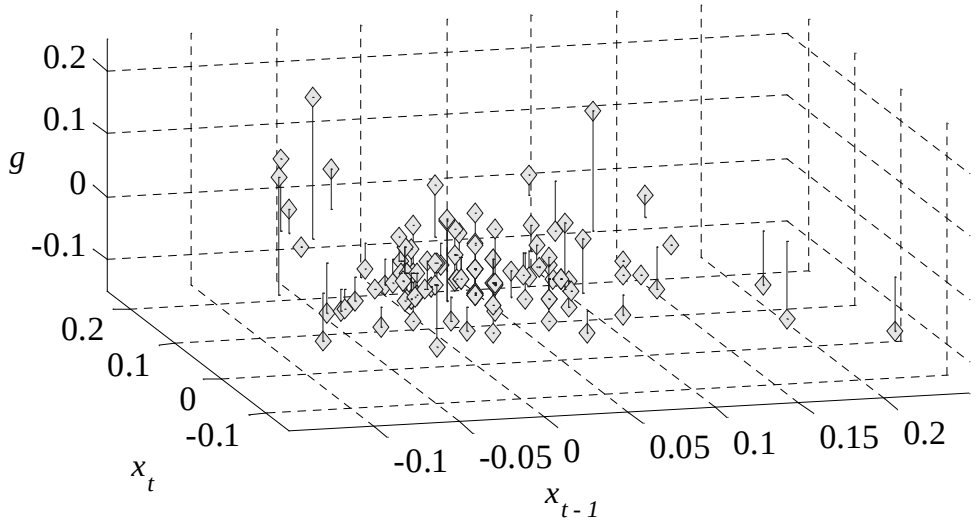

Figure 8.8 – Augmented Phase Space of Filtered ICN 1990H2 Daily Open Price (Testing)

The testing stage results are seen in Table 8.3.

Result	Set 1	Set 2	Set 3	Combined Set
Temporal pattern cluster count, $c(\mathcal{C})$	1	1	1	3
Temporal pattern cluster dimensions	1	3	5	1,3,5
Clusters cardinality, $c(M)$	13	16	12	32
Clusters mean eventness, μ_M	4.16%	0.96%	1.95%	1.48%
Clusters standard deviation eventness, σ_M	9.58%	8.41%	9.64%	7.97%
Non-clusters cardinality, $c(\tilde{M})$	112	107	109	93
Non-clusters mean eventness, $\mu_{\tilde{M}}$	-0.56%	-0.23%	-0.30%	-0.60%
Non-clusters standard deviation eventness, $\sigma_{\tilde{M}}$	4.80%	5.15	5.09%	4.48%
z_r	-1.12	-1.95	-2.40	-3.45
α_r	$2.62\text{x}10^{-1}$	$5.06\text{x}10^{-2}$	$1.65\text{x}10^{-2}$	$5.5\text{x}10^{-4}$
z_m	1.75	0.55	0.78	1.40
α_m	$8.02\text{x}10^{-2}$	$5.82\text{x}10^{-1}$	$4.25\text{x}10^{-1}$	$1.61\text{x}10^{-1}$

Table 8.3 – Filtered ICN 1990H2 Daily Open Price Results (Testing)

As with the training stage results, the average eventness values of time series observations inside the temporal pattern clusters are greater than the average eventness of the observations outside the temporal pattern clusters. However, for the same reasons discussed previously – sample size and temporal pattern stationarity – the statistical significance as shown by α is never less than 0.01. The TSDM goal is met in that a trading-edge is identified, but it is not statistically significant.

8.1.2 ICN 1991 Time Series Using Open Price

The same TSDM goal, objective function, event characterization function and optimization formulation are applied to the 1991 open price time series. The observed time series *X*, the open price for first half of 1991 (1991H1), is illustrated in Figure 8.9.

Figure 8.10 shows the filtered observed time series observations. Figure 8.11 presents an illustrative phase space, and Figure 8.12 an illustrative augmented phase space. The tournament genetic algorithm search parameters are presented in Table 8.1. The training stage results are shown in Table 8.4.

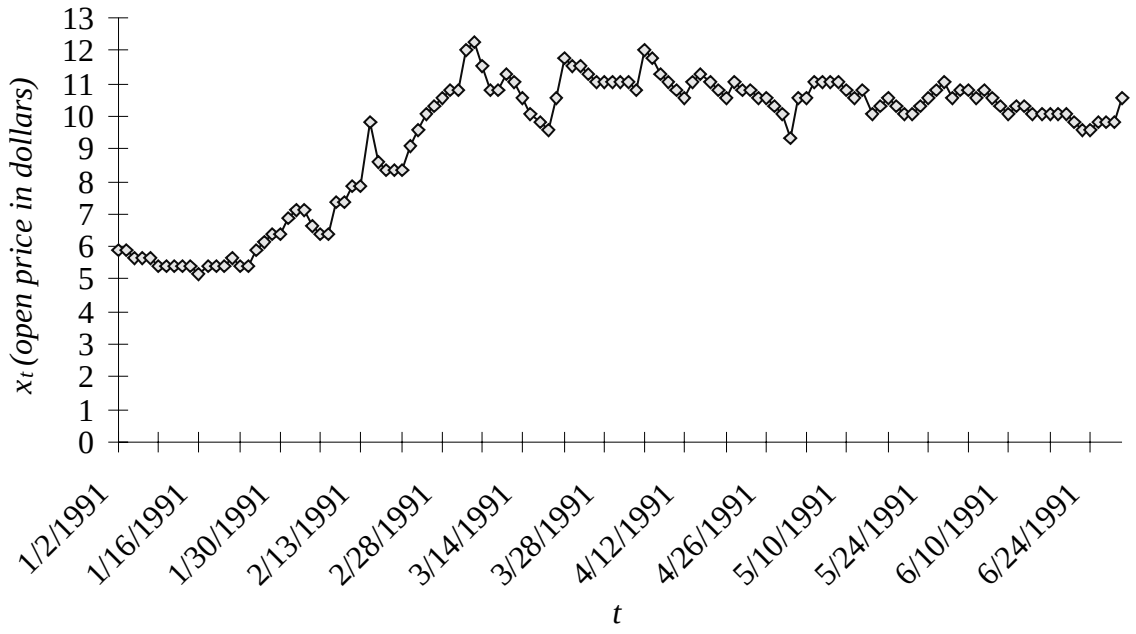

Figure 8.9 – ICN 1991H1 Daily Open Price Time Series (Observed)

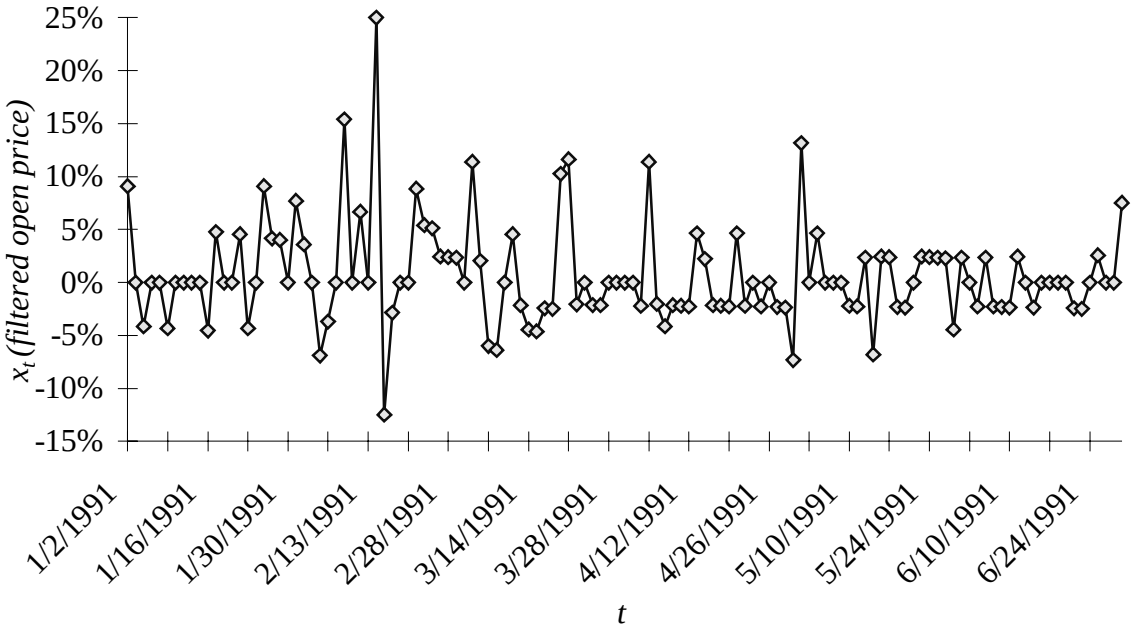

Figure 8.10 – Filtered ICN 1991H1 Daily Open Price Time Series (Observed)

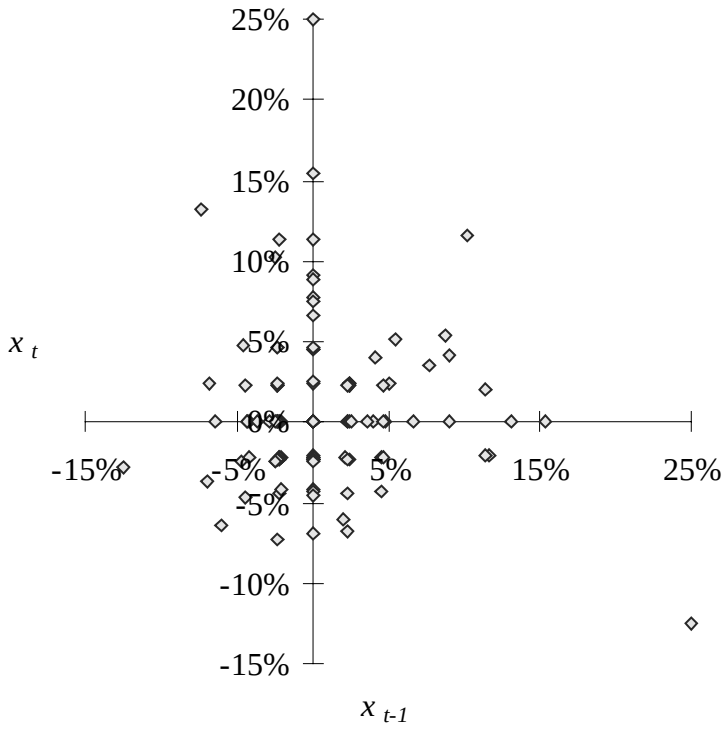

Figure 8.11 – Filtered ICN 1991H1 Daily Open Price Phase Space (Observed)

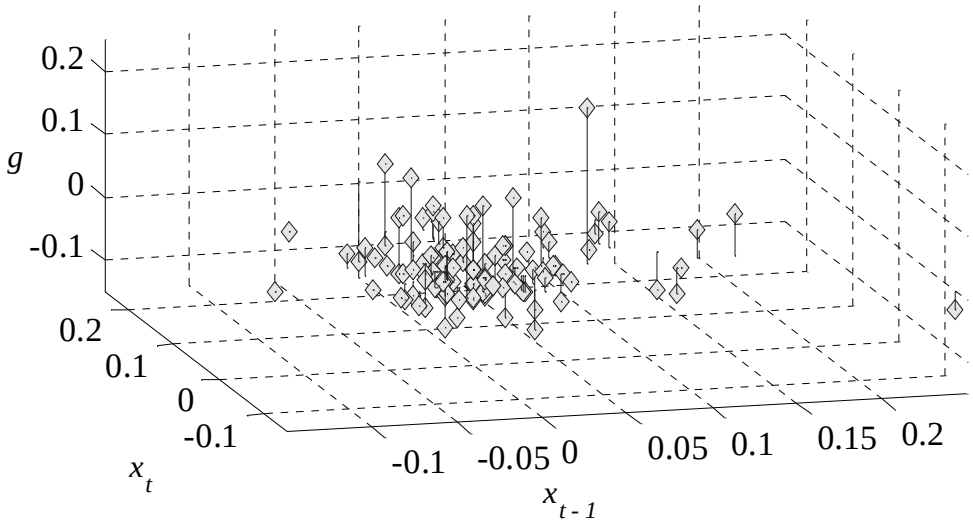

Figure 8.12 – Augmented Phase Space of Filtered ICN 1991H1 Daily Open Price (Observed)

Result	Set 1	Set 2	Set 3	Combined Set
Temporal pattern cluster count, $c(\mathcal{C})$	1	1	1	3
Temporal pattern cluster dimensions	1	3	5	1,3,5
Clusters cardinality, $c(M)$	7	8	6	19
Clusters mean eventness, μ_M	4.62%	4.41%	5.49%	3.71%
Clusters standard deviation eventness, σ_M	3.59%	9.50	10.13%	6.65%
Non-clusters cardinality, $c(\tilde{M})$	116	113	113	104
Non-clusters mean eventness, $\mu_{\tilde{M}}$	0.34%	0.36%	0.42%	0.01%
Non-clusters standard deviation eventness, $\sigma_{\tilde{M}}$	4.79%	4.29%	4.36%	4.21%
z_r	-1.05	0.04	-2.39	-3.19
α_r	$2.95\text{x}10^{-1}$	$9.65\text{x}10^{-1}$	$1.68\text{x}10^{-2}$	$1.43\text{x}10^{-3}$

Result	Set 1	Set 2	Set 3	Combined Set
z_m	2.99	1.197270	1.219608	2.336999
α_m	2.75×10^{-3}	2.31×10^{-1}	2.23×10^{-1}	1.94×10^{-2}

Table 8.4 – Filtered ICN 1991H1 Daily Open Price Results (Observed)

The training results show that a trading-edge can be found from the observed time series. However, because of the small sample size, statistical significance is more difficult to achieve. The testing stage time series is illustrated by Figure 8.13.

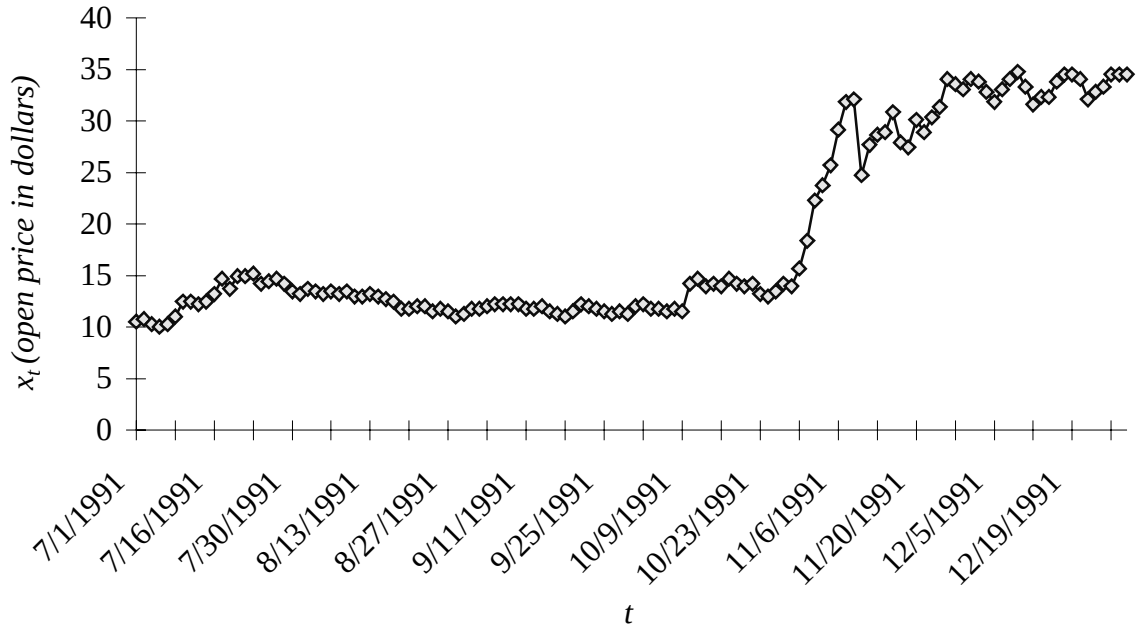

Figure 8.13 – ICN 1991H2 Daily Open Price Time Series (Testing)

The filtered version of the testing time series is shown in Figure 8.14. Illustrative phase and augmented phase spaces are shown in Figure 8.15 and Figure 8.16, respectively. The training stage results are seen in Table 8.5.

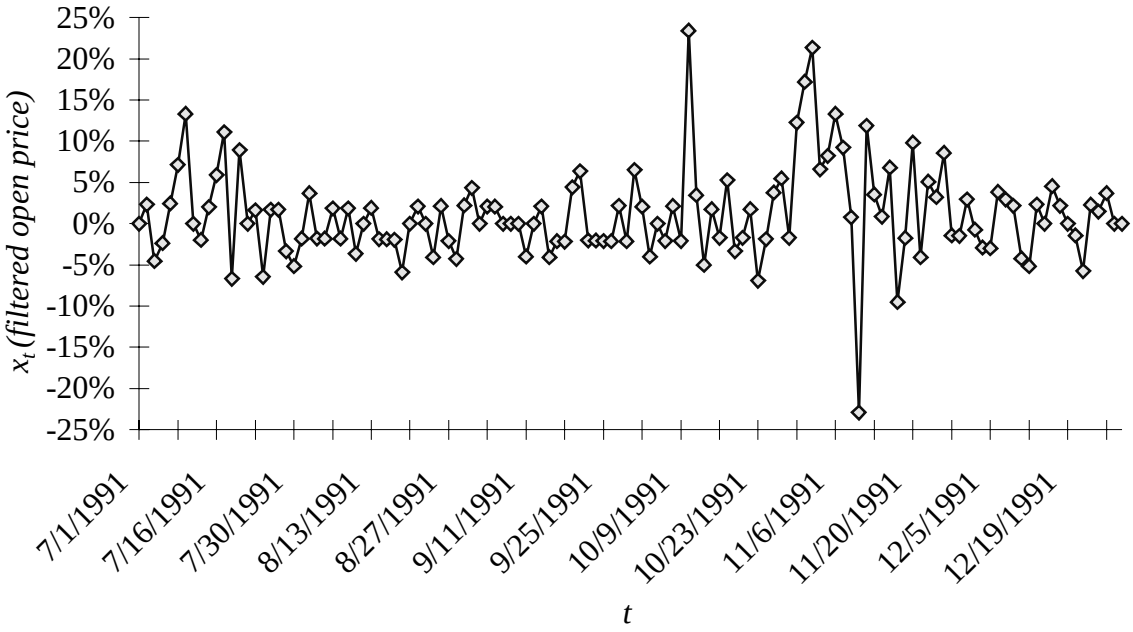

Figure 8.14 – Filtered ICN 1991H2 Daily Open Price Time Series (Testing)

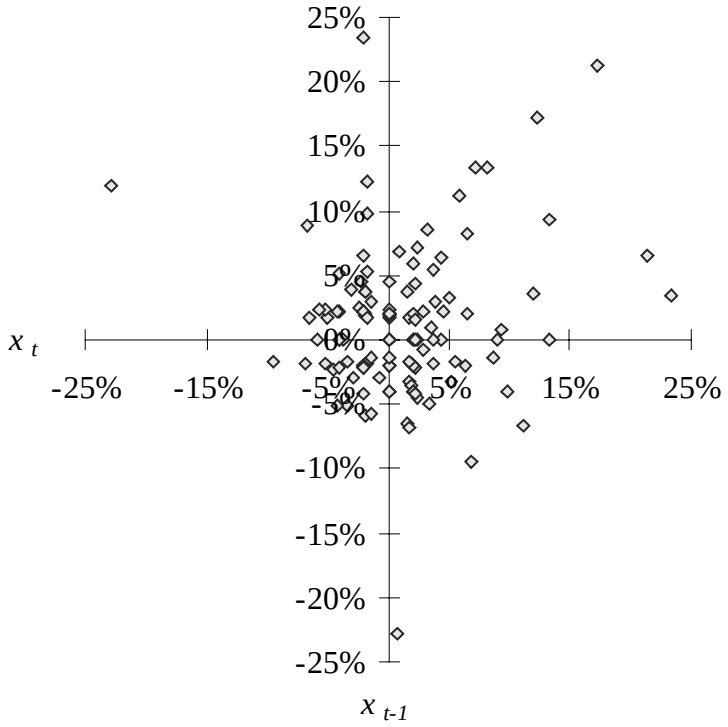

Figure 8.15 – Filtered ICN 1991H2 Daily Open Price Phase Space (Testing)

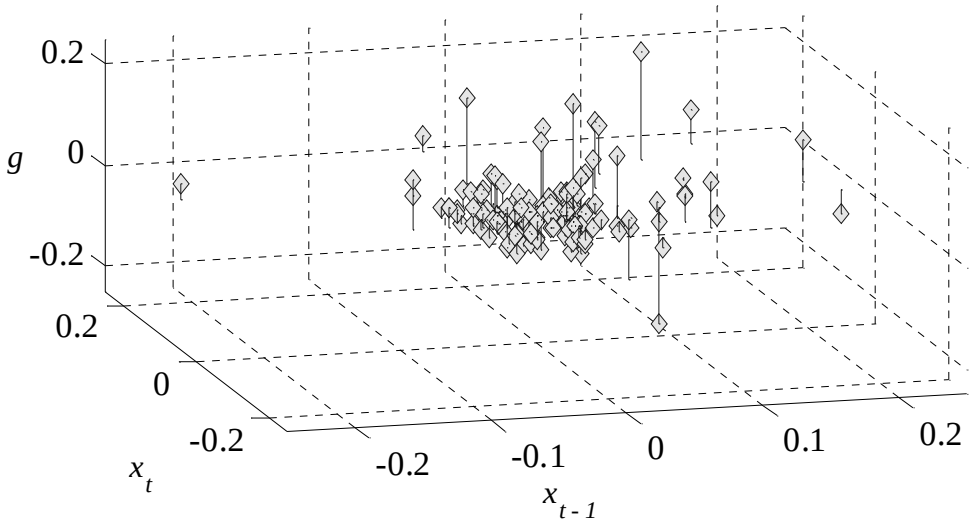

Figure 8.16 – Augmented Phase Space of Filtered ICN 1991H2 Daily Open Price (Testing)

Result	Set 1	Set 2	Set 3	Combined Set
Temporal pattern cluster count, $c(\mathcal{C})$	1	1	1	3
Temporal pattern cluster dimensions	1	3	5	1,3,5
Clusters cardinality, $c(M)$	13	7	7	22
Clusters mean eventness, μ_M	2.06%	0.46%	0.88%	0.41%
Clusters standard deviation eventness, σ_M	7.21%	5.06%	11.93%	8.04%
Non-clusters cardinality, $c(\tilde{M})$	113	117	115	104
Non-clusters mean eventness, $\mu_{\tilde{M}}$	0.98%	1.2%	1.12%	1.23%
Non-clusters standard deviation eventness, $\sigma_{\tilde{M}}$	5.57%	5.81%	5.28%	5.16%
z_r	-0.65	0.69	-0.17	-1.66
α_r	5.18×10^{-1}	4.90×10^{-1}	8.65×10^{-1}	9.69×10^{-2}

Result	Set 1	Set 2	Set 3	Combined Set
z_m	0.52	-0.37	-0.05	-0.46
α_m	6.01×10^{-1}	7.09×10^{-1}	9.59×10^{-1}	6.47×10^{-1}

Table 8.5 – Filtered ICN 1991H2 Daily Open Price Results (Testing)

For this collection of testing stage results, Set 1 has a higher cluster mean eventness than non-cluster mean eventness. Sets 2, 3, and combined do not. These results are presented so they may be contrasted with those in the next section, which incorporates the volume time series in predicting events. The next section demonstrates that, for the same set of possible events, including the volume time series yields better and more statistically significant temporal pattern clusters.

8.2 ICN Time Series Using Open Price and Volume

This section extends the results of applying the TSDM method to predicting the change in the open price of ICN by including the volume time series in the analysis. As with the previous section, this one is broken into two subsections each addressing 1990 and 1991 periods, respectively. Adding information in the form of a second time series enables better characterization and prediction results.

8.2.1 ICN 1990 Time Series Using Open Price and Volume

Figure 8.17 illustrates the observed time series *X*, the first half of 1990 (1990H1) open price and volume time series. The TSDM goal remains the same, as does the representation in TSDM concepts. The search parameters are described in Table 8.1, and the training stage results are shown in Table 8.6.

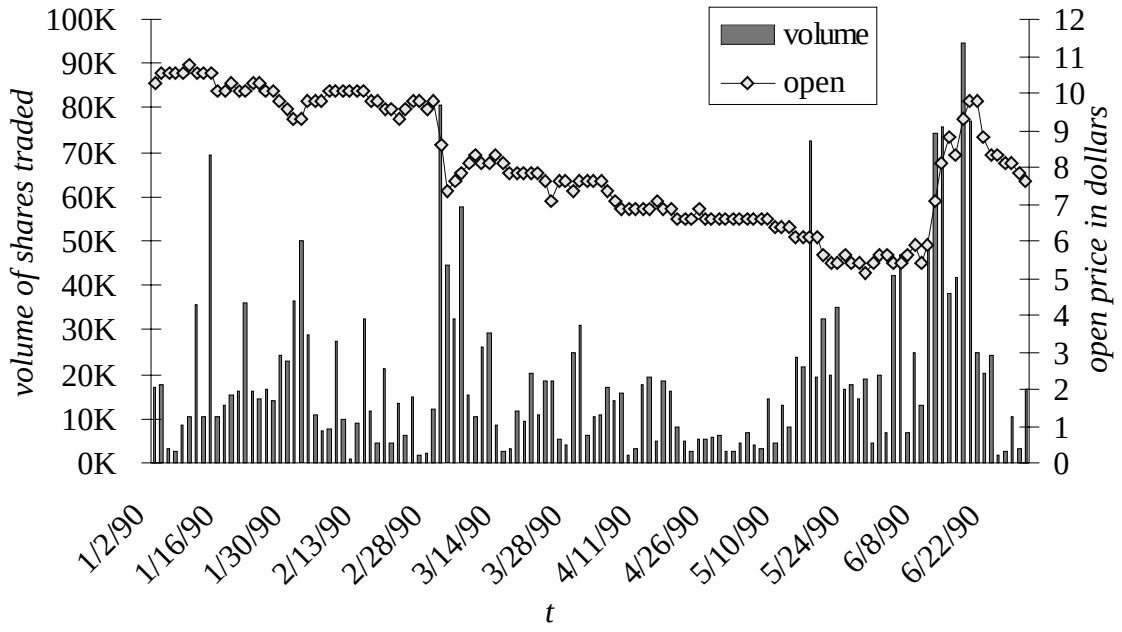

Figure 8.17 -ICN 1990H1 Daily Open Price and Volume Time Series (Observed)

Result	Set 1	Set 2	Set 3	Combined Set
Temporal pattern cluster count, $c(\mathcal{C})$	1	1	1	3
Temporal pattern cluster dimensions	2	6	10	2,6,10
Clusters cardinality, $c(M)$	6	7	6	13
Clusters mean eventness, μ_M	7.24%	4.85%	7.95%	5.09%
Clusters standard deviation eventness, σ_M	9.50%	7.68%	7.15%	7.27%
Non-clusters cardinality, $c(\tilde{M})$	118	115	114	111
Non-clusters mean eventness, $\mu_{\tilde{M}}$	-0.55%	-0.48%	-0.63%	-0.79%
Non-clusters standard deviation eventness, $\sigma_{\tilde{M}}$	3.57%	3.92%	3.78%	3.38%
z_r	-2.46	-0.17	-0.40	-2.08
α_r	$1.39\text{x}10^{-2}$	$8.65\text{x}10^{-1}$	$6.89\text{x}10^{-1}$	$3.73\text{x}10^{-2}$

Result	Set 1	Set 2	Set 3	Combined Set
z_m	2.00	1.82	2.92	2.88
α_m	4.54×10^{-2}	6.83×10^{-2}	3.53×10^{-3}	4.02×10^{-3}

Table 8.6 – ICN 1990H1 Daily Open Price and Volume Results (Observed)

In each case, the cluster mean eventness is greater than the non-cluster mean eventness. A comparison to the same time period results from Table 8.2 shows that these results are better for both the cluster mean eventness and the statistical measures. Four of the statistical tests are significant to the 0.05 α level.

The testing stage time series is shown in Figure 8.18. The testing stage results are seen in Table 8.7.

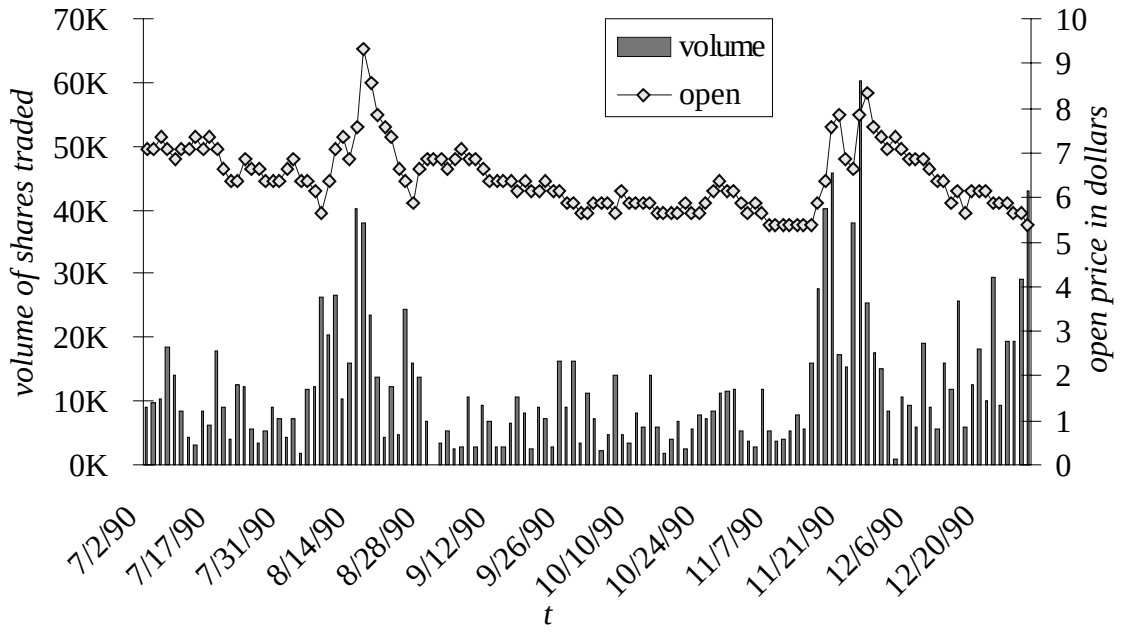

Figure 8.18 – ICN 1990H2 Daily Open Price and Volume Time Series (Testing)

Result	Set 1	Set 2	Set 3	Combined Set
Temporal pattern cluster count, $c(\mathcal{C})$	1	1	1	3
Temporal pattern cluster dimensions	2	6	10	2,6,10

Result	Set 1	Set 2	Set 3	Combined Set
Clusters cardinality, $c(M)$	12	7	6	18
Clusters mean eventness, μ_M	5.24%	3.14%	4.41%	3.27%
Clusters standard deviation eventness, σ_M	9.14%	10.67%	12.57%	9.44%
Non-clusters cardinality, $c(\tilde{M})$	113	116	115	107
Non-clusters mean eventness, $\mu_{\tilde{M}}$	-0.63%	-0.27%	-0.31%	-0.63%
Non-clusters standard deviation eventness, $\sigma_{\tilde{M}}$	4.84%	5.22%	5.09%	4.52%
z_r	-1.42	-0.18	-4.43	-2.87
α_r	$1.57\text{x}10^{-1}$	$8.60\text{x}10^{-1}$	$9.44\text{x}10^{-6}$	$4.09\text{x}10^{-3}$
z_m	2.19	0.84	0.91	1.72
α_m	$2.84\text{x}10^{-2}$	$4.02\text{x}10^{-1}$	$3.61\text{x}10^{-1}$	$8.54\text{x}10^{-2}$

Table 8.7 – ICN 1990H2 Daily Open Price and Volume Results (Testing)

As with the training stage, the testing stage results achieve the goal of finding a trading-edge. The cluster mean eventness is greater than the non-cluster mean eventness. A comparison to the same time period results from Table 8.3 reveals that these results are better in both the cluster mean eventness and the statistical measures. Three of the statistical tests are significant to the 0.05 α level.

8.2.2 ICN 1991 Time Series Using Open Price and Volume

Figure 8.19 illustrates the observed time series *X*, the first half of 1990 (1990H1) open price and volume time series. The training stage results are shown in Table 8.8

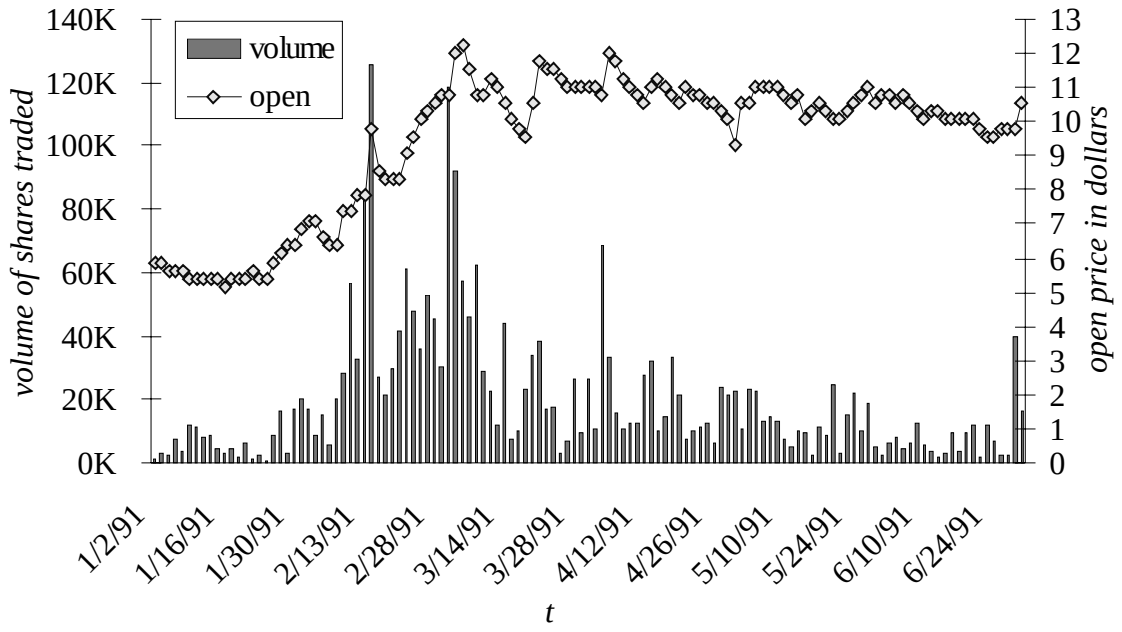

Figure 8.19 -ICN 1991H1 Daily Open Price and Volume Time Series (Observed)

Result	Set 1	Set 2	Set 3	Combined Set
Temporal pattern cluster count, $c(\mathcal{C})$	1	1	1	3
Temporal pattern cluster dimensions	2	6	10	2,6,10
Clusters cardinality, $c(M)$	7	7	6	12
Clusters mean eventness, μ_M	5.76%	10.54%	9.88%	7.87%
Clusters standard deviation eventness, σ_M	4.98%	6.87%	7.92%	6.78%
Non-clusters cardinality, $c(\tilde{M})$	116	114	113	111
Non-clusters mean eventness, $\mu_{\tilde{M}}$	0.27%	0.02%	0.19%	-0.20%
Non-clusters standard deviation eventness, $\sigma_{\tilde{M}}$	4.65%	3.99%	4.16%	3.85%
z_r	-1.05	-5.35	-2.39	-4.52

Result	Set 1	Set 2	Set 3	Combined Set
α_r	2.95×10^{-1}	8.91×10^{-8}	1.68×10^{-2}	6.15×10^{-6}
z_m	2.84	4.01	2.98	4.05
α_m	4.53×10^{-3}	6.16×10^{-5}	2.92×10^{-3}	5.07×10^{-5}

Table 8.8 – ICN 1991H1 Daily Open Price and Volume Results (Observed)

Again, the cluster mean eventness is greater than the non-cluster mean eventness for each set, and the results are better than the same time period results from Table 8.4, which used only the open price time series. All but one of the statistical tests are significant to the 0.05 α level, and all but two are significant to the 0.005 α level. The testing stage time series is shown in Figure 8.20, and the results are seen in Table 8.9.

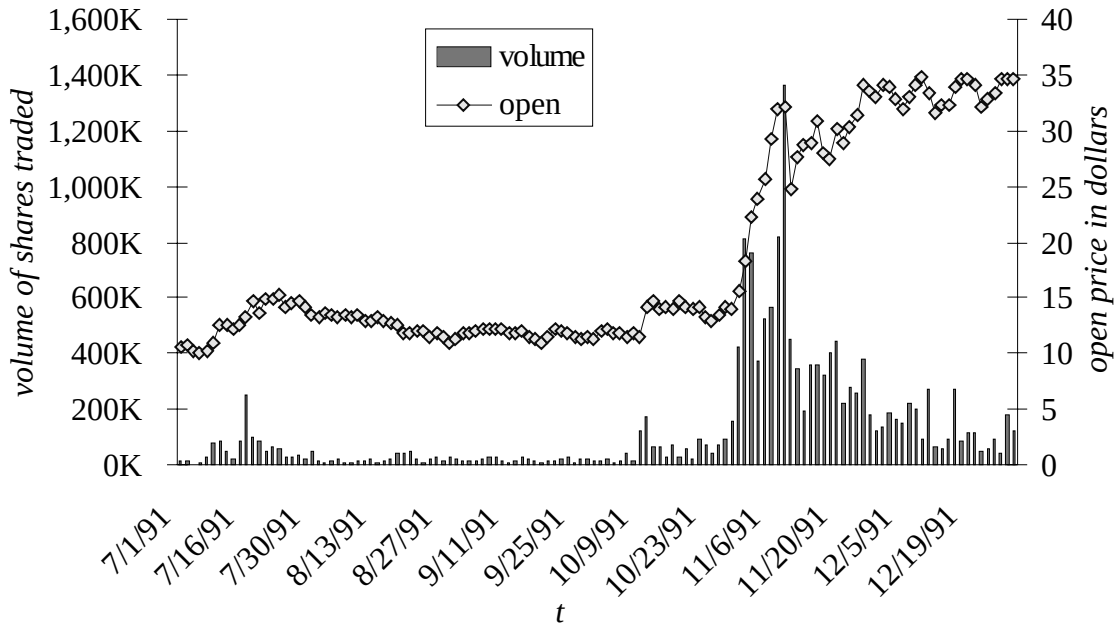

Figure 8.20 – ICN 1991H2 Daily Open Price and Volume Time Series (Testing)

Result	Set 1	Set 2	Set 3	Combined Set
Temporal pattern cluster count, $c(\mathcal{C})$	1	1	1	3
Temporal pattern cluster dimensions	2	6	10	2,6,10

Result	Set 1	Set 2	Set 3	Combined Set
Clusters cardinality, $c(M)$	9	6	4	15
Clusters mean eventness, μ_M	5.14%	1.26%	6.40%	3.48%
Clusters standard deviation eventness, σ_M	7.98%	15.07%	11.91%	11.07%
Non-clusters cardinality, $c(\tilde{M})$	117	118	118	111
Non-clusters mean eventness, $\mu_{\tilde{M}}$	0.78%	1.16%	0.92%	0.77%
Non-clusters standard deviation eventness, $\sigma_{\tilde{M}}$	5.45%	5.01%	5.46%	4.58%
z_r	0.89	-3.48	-1.12	-1.05
α_r	3.75×10^{-1}	5.08×10^{-4}	2.61×10^{-1}	2.95×10^{-1}
z_m	1.61	0.02	0.92	0.94
α_m	1.07×10^{-1}	9.87×10^{-1}	3.59×10^{-1}	3.48×10^{-1}

Table 8.9 – ICN 1991H2 Daily Open Price and Volume Results (Testing)

As with the characterization, the cluster mean eventness for each set is greater than the non-cluster mean eventness. A comparison to the same time period results (from Table 8.5) shows that these results are better in both the cluster mean eventness and the statistical measures. Recall that, in Table 8.5, only one of the sets had a cluster mean eventness that was greater than the non-cluster mean eventness. Here, all of the cluster mean eventnesses are greater. However, as seen before, the statistical significances are hampered by the limited sample size and temporal pattern stationarity.

In the next section, the ideas gained from analyzing the ICN time series are applied. For the ICN time series, the section applied a temporal pattern discovered in a

half-year's worth of data to the next half-year's worth of data. The next section will apply a half-year's worth of training to the next day's prediction. The training stage is repeated at each time-step.

8.3 DJIA Component Time Series

This section presents the results of applying the TSDMe$_2$-S/M method to the 30 open daily price time series of the Dow Jones Industrial Average (DJIA) components from January 2, 1990, through March 8, 1991, which allows approximately 200 testing stages. The following stocks in Table 8.10 make up the DJIA during this period.

Ticker	Company Name	Ticker	Company Name
AA	Aluminum of America	JNJ	Johnson & Johnson
ALD	AlliedSignal Inc.	JPM	J.P. Morgan
AXP	American Express	KO	Coca-Cola
BA	Boeing	MCD	McDonald's
CAT	Caterpillar	MMM	Minnesota Mining & Manufacturing
CHV	Chevron	MO	Philip Morris
DD	DuPont	MRK	Merck
DIS	Walt Disney	PG	Procter & Gamble
EK	Eastman Kodak	S	Sears, Roebuck
GE	General Electric	T	AT & T Corp.
GM	General Motors	TRV	Travelers (Now part of Citigroup Inc.)
GT	Goodyear Tire & Rubber	UK	Union Carbide
HWP	Hewlett-Packard	UTX	United Technologies
IBM	International Business Machines	WMT	Wal-Mart Stores
IP	International Paper	XON	Exxon

Table 8.10 – Dow Jones Industrial Average Components (1/2/1990 – 3/8/1991)

Rather than graphically present each of the 30 DJIA component stocks, Figure 8.21 illustrates the DJIA. As with the ICN time series, a percentage filter is applied to each DJIA component time series to facilitate finding temporal pattern clusters.

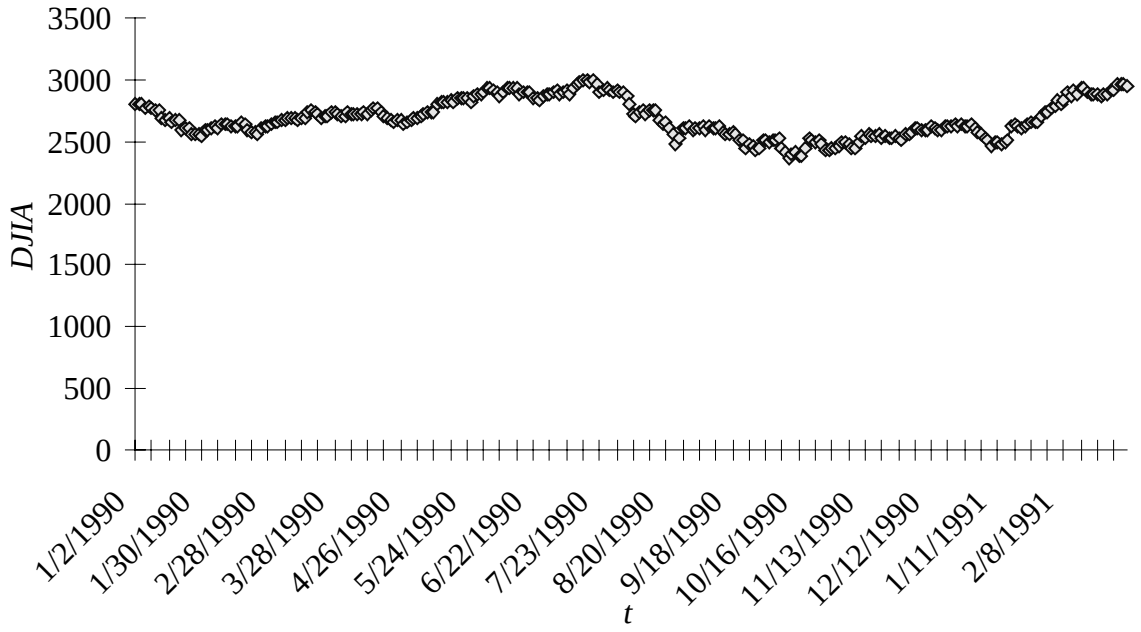

Figure 8.21 – DJIA Daily Open Price Time Series

The TSDM goal is to find a trading-edge. The next section shows how this goal is captured through TSDM concepts.

8.3.1 Training Stage

The objective function is

$$f(P) = \begin{cases} \mu_M & \text{if } c(M)/c(\Lambda) \geq \beta \\ (\mu_M - g_0)\dfrac{c(M)}{\beta c(\Lambda)} + g_0 & \text{otherwise} \end{cases}, \qquad (8.3)$$

where $\beta = 0.05$. The event characterization function to is $g(t) = x_{t+1}$, which allows for one-step-ahead characterization and prediction. The optimization formulation is $\max f(P)$.

Because of the large number of training processes – 5,970 – a graphical presentation of each step is not made. Recall that the $TSDMe_2$ method uses a moving training window and a single observation testing window. The training window is 100 observations.

The search parameters are presented in Table 8.11. The roulette selection genetic algorithm was used.

Parameter	Values
Random search multiplier	10
Population size	30
Elite count	1
Gene length	6
Mutation rate	0%
Convergence criteria	1

Table 8.11 – Genetic Algorithm Parameters for DJIA Component Time Series

Because of the large number of training and testing sets and because of the trading-edge goal, the results presented are of a summary nature. The statistical training results for each DJIA component are presented in Table 8.12. Of the 5,970 training processes, the cluster mean eventness (μ_M) was greater than total mean eventness (μ_X) every time. For 69% of the temporal pattern clusters, the probability of a Type I error was less than 5% based on the independent means statistical test. For 49% of the temporal pattern clusters, the probability of a Type I error was less than 5% based on the runs statistical test.

Ticker	$\mu_M > \mu_X$	$\alpha_m \leq 0.05$	$\alpha_r \leq 0.05$
AA	100%	82%	55%
ALD	100%	72%	52%
AXP	100%	71%	48%
BA	100%	70%	42%
CAT	100%	79%	48%
CHV	100%	54%	34%
DD	100%	42%	35%
DIS	100%	83%	25%
EK	100%	55%	18%
GE	100%	66%	81%
GM	100%	73%	49%
GT	100%	62%	44%
HWP	100%	55%	34%
IBM	100%	67%	24%
IP	100%	80%	78%
JNJ	100%	89%	37%
JPM	100%	90%	14%
KO	100%	67%	87%
MCD	100%	62%	62%
MMM	100%	57%	75%
MO	100%	65%	29%
MRK	100%	59%	70%
PG	100%	76%	38%
S	100%	59%	86%
T	100%	66%	40%
TRV	100%	78%	63%
UK	100%	36%	66%
UTX	100%	94%	46%
WMT	100%	73%	37%
XON	100%	75%	61%
Combined	100%	69%	49%

Table 8.12 – DJIA Component Results (Observed)

8.3.2 Testing Stage Results

Using the 5,970 training processes, 471 events are predicted. The statistical prediction results for each DJIA component are presented in Table 8.13. The cluster mean eventness (μ_M) was greater than the non-cluster mean eventness ($\mu_{\tilde{M}}$) 20 out of 30 times or 67% of the time. For 16.7% of the temporal pattern clusters, the probability of a Type I error was less than 5% based on the independent means statistical test. For 3.3% of the temporal pattern clusters, the probability of a Type I error was less than 5% based on the runs statistical test. These low rates of statistical significance at the 5% α level are typical for predictions of financial time series as seen from the previously presented ICN results.

Ticker	$c(M)$	μ_M	σ_M	$c(\tilde{M})$	$\mu_{\tilde{M}}$	$\sigma_{M'}$	α_m	α_r
AA	16	0.569%	1.652%	182	-0.013%	1.620%	$1.78x10^{-1}$	$7.76x10^{-1}$
ALD	14	0.438%	1.428%	184	-0.102%	1.851%	$1.83x10^{-1}$	$9.91x10^{-1}$
AXP	14	0.027%	2.058%	184	-0.023%	2.610%	$9.32x10^{-1}$	$9.91x10^{-1}$
BA	13	0.080%	2.044%	185	-0.030%	2.181%	$8.52x10^{-1}$	$1.76x10^{-1}$
CAT	26	-0.003%	1.817%	172	-0.098%	2.127%	$8.08x10^{-1}$	$3.19x10^{-1}$
CHV	16	0.057%	1.572%	182	0.061%	1.200%	$9.92x10^{-1}$	$8.40x10^{-1}$
DD	16	0.526%	1.946%	182	-0.045%	1.635%	$2.55x10^{-1}$	$7.76x10^{-1}$
DIS	20	-0.024%	1.488%	178	0.069%	2.069%	$8.00x10^{-1}$	$9.87x10^{-1}$
EK	14	-0.045%	1.879%	184	0.074%	1.998%	$8.20x10^{-1}$	$2.66x10^{-1}$
GE	16	0.094%	1.410%	182	0.000%	1.881%	$8.04x10^{-1}$	$4.92x10^{-1}$
GM	16	0.671%	2.090%	182	-0.149%	1.863%	$1.29x10^{-1}$	$4.92x10^{-1}$
GT	20	-0.962%	2.034%	178	-0.066%	2.549%	$6.93x10^{-2}$	$9.87x10^{-1}$

Ticker	$c(M)$	μ_M	σ_M	$c(\tilde{M})$	$\mu_{\tilde{M}}$	$\sigma_{M'}$	α_m	α_r
HWP	13	-0.779%	1.881%	185	0.116%	2.664%	$1.08\text{x}10^{-1}$	$1.76\text{x}10^{-1}$
IBM	16	-1.079%	1.785%	182	0.175%	1.460%	$6.32\text{x}10^{-3}$	$8.41\text{x}10^{-1}$
IP	16	1.197%	2.525%	182	0.025%	1.587%	$6.80\text{x}10^{-2}$	$2.09\text{x}10^{-1}$
JNJ	13	0.665%	1.444%	185	0.160%	1.551%	$2.25\text{x}10^{-1}$	$8.63\text{x}10^{-1}$
JPM	11	1.420%	1.878%	187	0.040%	1.985%	$1.82\text{x}10^{-2}$	$5.90\text{x}10^{-1}$
KO	11	1.794%	3.396%	187	0.008%	1.807%	$8.36\text{x}10^{-2}$	$2.18\text{x}10^{-1}$
MCD	13	0.367%	1.753%	185	-0.013%	1.977%	$4.54\text{x}10^{-1}$	$3.14\text{x}10^{-1}$
MMM	16	0.238%	1.044%	182	0.043%	1.258%	$4.82\text{x}10^{-1}$	$4.92\text{x}10^{-1}$
MO	17	0.038%	1.820%	181	0.251%	1.641%	$6.42\text{x}10^{-1}$	$1.80\text{x}10^{-1}$
MRK	19	0.669%	1.163%	179	0.073%	1.580%	$4.11\text{x}10^{-2}$	$7.10\text{x}10^{-2}$
PG	13	0.174%	1.615%	185	0.047%	1.707%	$7.85\text{x}10^{-1}$	$3.14\text{x}10^{-1}$
S	14	1.449%	2.677%	184	-0.157%	1.938%	$2.77\text{x}10^{-2}$	$9.28\text{x}10^{-4}$
T	11	1.307%	1.797%	187	-0.193%	1.645%	$6.88\text{x}10^{-3}$	$5.44\text{x}10^{-2}$
TRV	21	1.531%	2.449%	177	-0.147%	2.617%	$3.21\text{x}10^{-3}$	$5.58\text{x}10^{-1}$
UK	14	-0.449%	2.263%	184	0.041%	1.900%	$4.30\text{x}10^{-1}$	$5.75\text{x}10^{-1}$
UTX	14	-0.289%	1.979%	184	-0.028%	1.828%	$6.33\text{x}10^{-1}$	$2.66\text{x}10^{-1}$
WMT	18	0.658%	1.950%	180	0.120%	2.458%	$2.77\text{x}10^{-1}$	$5.79\text{x}10^{-1}$
XON	20	0.077%	1.398%	178	0.090%	1.263%	$9.68\text{x}10^{-1}$	$4.19\text{x}10^{-1}$
All	471	0.313%	1.970%	5,469	0.011%	1.919%	$1.38\text{x}10^{-3}$	$6.76\text{x}10^{-1}$
Top 15	245	0.596%	1.966%	2,725	-0.020%	1.809%	$2.27\text{x}10^{-6}$	$8.84\text{x}10^{-3}$

Table 8.13 – DJIA Component Results (Testing)

For the combined results – using all predictions – the mean cluster eventness is greater than the non-cluster mean eventness. It also is statistically significant to the 0.005α level according to the independent means test. However, better results can be achieved by predicting which temporal pattern clusters are more likely to yield accurate predictions. This is done by defining

$$\alpha_\mu = \frac{(\alpha_m \le 0.05) + (\alpha_r \le 0.05)}{2}. \tag{8.4}$$

The α_μ is the average of the $\alpha_m \le 0.05$ and $\alpha_r \le 0.05$ from Table 8.12. The excess return,

$$\mu_e = \mu_M - \mu_{\tilde{M}}, \tag{8.5}$$

is the difference in the returns achieved by using the temporal pattern clusters and the complement of the temporal pattern clusters. The α_μ has a 0.50 correlation with the excess return. Figure 8.22 illustrates this.

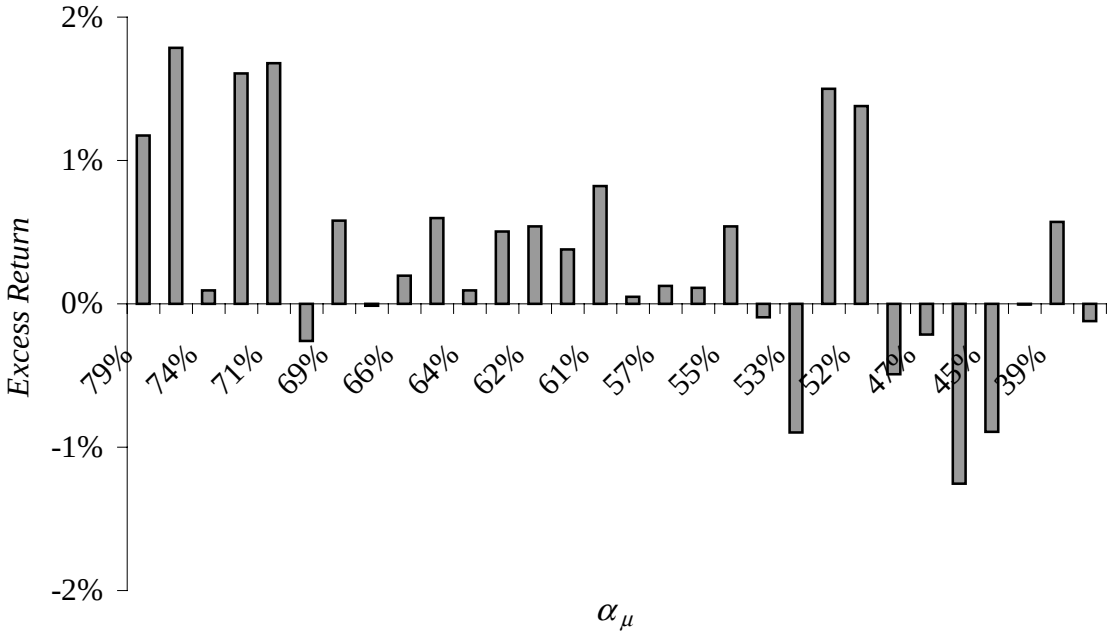

Figure 8.22 – α_μ vs. Excess Return

The top 15 stocks are selected based on their α_μ. The prediction results using the portfolio formed from these top 15 DJIA components yields exceptional results. Using the temporal pattern clusters for the top 15 stocks, 245 predictions are made. The cluster mean eventness (μ_M) was greater than the non-cluster mean eventness ($\mu_{\tilde{M}}$) 13 out of 15 times or 87% of the time. The average predicted event had a 0.596% increase in open price. The average of the not predicted events was -0.020%. According to both statistical tests, the results are statistically significant. Using the means test, there is only a 0.000227% chance of making a Type I error in rejecting the null hypothesis that the predicted events are the same as the not predicted observations. Using the runs test, there is a 0.884% chance of making a Type I error.

The best way to understand the effectiveness of the TSDM method when applied to financial time series is to show the trading results that can be achieved by applying the temporal pattern clusters discovered above. An initial investment is made as follows: If a temporal pattern cluster from any of the stocks in the portfolio predicts a high eventness, the initial investment is made in that stock for one day. If there are temporal pattern clusters for several stocks that indicate high eventness, the initial investment is split equally among the stocks. If there are no temporal pattern clusters indicating high eventness, then the initial investment is invested in a money market account with an assumed 5% annual rate of return. The training process is rerun using the new 100 most recent observation window. The following day, the initial investment principal plus return is invested according to the same rules. The process is repeated for the remaining investment period.

The results for the investment period of May 29, 1990 through March 8, 1991 are shown in Table 8.14. This period is less than the total time frame (January 1, 1990, through March 8, 1991) because the first part of the time series is used only for training. The return of the DJIA also is given, which is slightly different from the buy and hold strategy for all DJIA components because the DJIA has a non-equal weighting among its components.

Portfolio	Investment Method	Return	Annualized Return
All DJIA components	Temporal Pattern Cluster	30.98%	41.18%
Top 15 DJIA components	Temporal Pattern Cluster	67.77%	93.70%
DJIA	Buy and hold	2.95%	3.79%
All DJIA components	Not in Temporal Pattern Cluster	0.35%	0.45%
Top 15 DJIA components	Not in Temporal Pattern Cluster	-2.94%	-3.74%
All DJIA components	Buy and hold	3.34%	4.29%
Top 15 DJIA components	Buy and hold	2.81%	3.60%

Table 8.14 – Trading Results

An initial investment of $10,000 made on May 29, 1990, in the top 15 DJIA component stocks using the TSDM method would have grown to $16,777 at the end of March 8, 1991. One caveat to this result is that it ignores trading costs [59]. The trading cost is a percentage of the amount invested and includes both the buying and selling transaction costs along with the spread between the bid and ask. The return of the top 15 DJIA component portfolio using the temporal pattern cluster investment method is reduced to 63.73% or 87.76% annualized when a trading cost rate of 0.01% applied. This

level of trading cost would require investments in the $500,000 to $1,000,000 range and access to trading systems that execute in between the bid and ask prices or have spreads of 1/16th or less. A 0.2% trading cost applied to the same portfolio results would reduce the return to 3.54% or 4.55% annualized.

In this chapter, the TSDM method was applied to financial time series. Using temporal pattern clusters from single and multiple time series as a trading tool has yielded significant results. Even with a complex, nonstationary time series like stock price and volume, the TSDM method uncovers temporal patterns that are both characteristic and predictive.

Chapter 9 Conclusions and Future Efforts

Through the novel Time Series Data Mining (TSDM) framework and its associated methods, this dissertation has made an original and fundamental contribution to the fields of time series analysis and data mining. The key TSDM concepts of event, event characterization function, temporal pattern, temporal pattern cluster, time-delay embedding, phase space, augmented phase space, objective function, and optimization were reviewed, setting up the framework from which to develop TSDM methods.

Chapters 4 and 6 developed TSDM methods to find optimal temporal pattern clusters that both characterize and predict time series events. TSDM methods were created for discovering both single and multiple temporal pattern clusters in single and multi-dimensional time series. Additionally, a set of filtering and time series windowing techniques was adapted to allow prediction of nonstationary events.

This dissertation has demonstrated that methods based on the TSDM framework successfully characterize and predict complex, nonperiodic, irregular, and chaotic time series. This was done, first, through a set of explanatory and basic examples that demonstrated the TSDM process. TSDM methods were then successfully applied to characterizing and predicting complex, nonstationary, chaotic time series events from both the engineering and financial domains. Given a multi-dimensional time series generated by sensors on a welding station, the TSDM framework was able to, with a high degree of accuracy, characterize and predict metal droplet releases. In the financial domain, the TSDM framework was able to generate a trading-edge by characterizing and predicting stock price events.

Future efforts will fall into three categories: theoretical, application, and performance. Theoretical research will be conducted to determine the required dimension of the reconstructed phase space given an arbitrary number of observable states. There are many research applications for TSDM, including: high frequency financial event prediction, incipient fault prediction in induction motor-drive systems, and characterization of heart fibrillation. As the time series data sets grow larger, the computational effort required to find hidden temporal patterns grows, requiring higher performance implementations of the TSDM methods.

As discussed in Chapter 2, Takens proved that a $2Q+1$ dimensional phase space formed using time-delay embedding is guaranteed to be an embedding of, i.e., topologically equivalent to, an original Q-dimensional state space. This theorem is based on using one observable state to reconstruct the state space. Povinelli and Feng showed experimentally in [2] that using multiple observable states can yield better results. The unanswered theoretical question is: What phase space dimension is required for an arbitrary number of observable states so that the phase space is topologically equivalent to the original state space? It is obvious that when all Q states are observable, then the reconstructed phase space need only be Q-dimensional. Future research efforts will investigate the relationship between the number of observable states n and the required phase space dimensionality when $1 < n < Q$.

One of the future application efforts will be to create a synergy between the research of Demerdash and Bangura, which demonstrated the powerful abilities of the Time-Stepping Coupled Finite Element-State Space (TSCFE-SS) method in predicting *a priori* characteristic waveforms of healthy and faulty motor performance characteristics

[60-65], and the Time Series Data Mining (TSDM) framework presented in this dissertation to characterizing and predicting incipient motor faults.

Improving computational performance will be addressed through two research directions. One direction is to investigate alternative global optimization methods such as interval branch and bound. A second parallel direction is to investigate distributed and parallel implementations of the TSDM methods.

Through the creation of the novel TSDM framework and methods, which have been validated on complex real-world time series, this dissertation has made a significant contribution to the state of the art in the fields of time series analysis and data mining.

References

[1] S. M. Pandit and S.-M. Wu, *Time series and system analysis, with applications*. New York: Wiley, 1983.

[2] R. J. Povinelli and X. Feng, "Data Mining of Multiple Nonstationary Time Series," proceedings of Artificial Neural Networks in Engineering, St. Louis, Missouri, 1999, pp. 511-516.

[3] R. J. Povinelli and X. Feng, "Temporal Pattern Identification of Time Series Data using Pattern Wavelets and Genetic Algorithms," proceedings of Artificial Neural Networks in Engineering, St. Louis, Missouri, 1998, pp. 691-696.

[4] G. E. P. Box and G. M. Jenkins, *Time series analysis: forecasting and control*, Rev. ed. San Francisco: Holden-Day, 1976.

[5] B. L. Bowerman and R. T. O'Connell, *Forecasting and time series: an applied approach*, 3rd ed. Belmont, California: Duxbury Press, 1993.

[6] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthursamy, *Advances in knowledge discovery and data mining*. Menlo Park, California: AAAI Press, 1996.

[7] S. M. Weiss and N. Indurkhya, *Predictive data mining: a practical guide*. San Francisco: Morgan Kaufmann, 1998.

[8] R. A. Gabel and R. A. Roberts, *Signals and linear systems*, 2nd ed. New York: Wiley, 1980.

[9] S. Haykin, *Adaptive filter theory*, 3rd ed. Upper Saddle River, New Jersey: Prentice Hall, 1996.

[10] C. K. Chui, *An introduction to wavelets*. Boston: Academic Press, 1992.

[11] C. K. Chui, *Wavelets: a tutorial in theory and applications*. Boston: Academic Press, 1992.

[12] I. Daubechies, *Ten lectures on wavelets*. Philadelphia: Society for Industrial and Applied Mathematics, 1992.

[13] E. Hernandez and G. L. Weiss, *A first course on wavelets*. Boca Raton, Florida: CRC Press, 1996.

[14] P. R. Massopust, *Fractal functions, fractal surfaces, and wavelets*. San Diego: Academic Press, 1994.

[15] T. H. Koornwinder, *Wavelets: an elementary treatment of theory and applications*. River Edge, New Jersey: World Scientific, 1993.

[16] G. Kaiser, *A friendly guide to wavelets*. Boston: Birkhäuser, 1994.

[17] G. Strang and T. Nguyen, *Wavelets and filter banks*. Wellesley, Massachusetts: Wellesley-Cambridge Press, 1996.

[18] R. Polikar, "The Engineer's Ultimate Guide To Wavelet Analysis - The Wavelet Tutorial," 2nd ed. available at http://www.public.iastate.edu/~rpolikar/WAVELETS/WTtutorial.html, 1996, cited 1 Aug 1997.

[19] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Reading, Massachusetts: Addison-Wesley, 1989.

[20] R. J. Povinelli and X. Feng, "Improving Genetic Algorithms Performance By Hashing Fitness Values," proceedings of Artificial Neural Networks in Engineering, St. Louis, Missouri, 1999, pp. 399-404.

[21] J. Heitkötter and D. Beasley, "The Hitch-Hiker's Guide to Evolutionary Computation (FAQ for comp.ai.genetic)," 5.2 ed. available at http://www.cs.purdue.edu/coast/archive/clife/FAQ/www/, 1997, cited 1 Aug 1997.

[22] R. L. Haupt and S. E. Haupt, *Practical genetic algorithms*. New York: Wiley, 1998.

[23] Z. Michalewicz, *Genetic algorithms + data structures = evolution programs*, 3rd rev. and extended ed. Berlin: Springer, 1996.

[24] E. Walters, *Design of efficient FIR digital filters using genetic algorithms*, Masters Thesis, Marquette University, 1998.

[25] G. Deboeck, *Trading on the edge: neural, genetic, and fuzzy systems for chaotic financial markets*. New York: Wiley, 1994.

[26] G. R. Harik, E. Cantú-Paz, D. E. Goldberg, and B. L. Miller, "The gambler's ruin problem, genetic algorithms, and the sizing of populations," proceedings of IEEE Conference on Evolutionary Computation, 1997, pp. 7-12.

[27] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, 1st MIT Press ed. Cambridge, Massachusetts: MIT Press, 1992.

[28] H. D. I. Abarbanel, *Analysis of observed chaotic data*. New York: Springer, 1996.

[29] A. J. Crilly, R. A. Earnshaw, and H. Jones, *Applications of fractals and chaos*. Berlin: Springer, 1993.

[30] N. B. Tufillaro, T. Abbott, and J. Reilly, *An experimental approach to nonlinear dynamics and chaos*. Redwood City, California: Addison-Wesley, 1992.

[31] E. E. Peters, *Chaos and order in the capital markets: a new view of cycles, prices, and market volatility*, 2nd ed. New York: Wiley, 1996.

[32] E. E. Peters, *Fractal market analysis: applying chaos theory to investment and economics*. New York: Wiley, 1994.

[33] R. Cawley and G.-H. Hsu, "Chaotic Noise Reduction by Local-Geometric-Projection with a Reference Time Series," proceedings of The Chaos Paradigm: Developments and Applications in Engineering and Science, Mystic, Connecticut, 1993, pp. 193-204.

[34] R. Cawley, G.-H. Hsu, and L. W. Salvino, "Detection and Diagnosis of Dynamics in Time Series Data: Theory of Noise Reduction," proceedings of The Chaos Paradigm: Developments and Applications in Engineering and Science, Mystic, Connecticut, 1993, pp. 182-192.

[35] J. Iwanski and E. Bradley, "Recurrence plot analysis: To embed or not to embed?," *Chaos*, vol. 8, pp. 861-871, 1998.

[36] F. Takens, "Detecting strange attractors in turbulence," proceedings of Dynamical Systems and Turbulence, Warwick, 1980, pp. 366-381.

[37] T. Sauer, J. A. Yorke, and M. Casdagli, "Embedology," *Journal of Statistical Physics*, vol. 65, pp. 579-616, 1991.

[38] "Aussie Gold History," available at http://www.uq.net.au/~zzdvande/history.html, cited 13 Sep 1998.

[39] "Newmont - Core Gold Values," available at http://www.newmont.com/aboutthe1.htm, cited 10 Sep 1998.

[40] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From Data Mining to Knowledge Discovery: An Overview," in *Advances in knowledge discovery and data mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthursamy, Eds. Menlo Park, California: AAAI Press, 1996.

[41] A. A. Freitas and S. H. Lavington, *Mining very large databases with parallel processing*. Boston: Kluwer Academic Publishers, 1998.

[42] H. Liu and H. Motoda, *Feature selection for knowledge discovery and data mining*. Boston: Kluwer Academic Publishers, 1998.

[43] P. Cabena and International Business Machines Corporation., *Discovering data mining : from concept to implementation*. Upper Saddle River, New Jersey: Prentice Hall, 1998.

[44] P. Gray and H. J. Watson, *Decision support in the data warehouse*. Upper Saddle River, New Jersey: Prentice Hall, 1998.

[45] S. Iyanaga and Y. Kawada, *Encyclopedic dictionary of mathematics by the Mathematical Society of Japan*. Cambridge, Massachusetts: MIT Press, 1977.

[46] E. Bradley, "Analysis of Time Series," in *An introduction to intelligent data analysis*, M. Berthold and D. Hand, Eds. New York: Springer, 1999, pp. 167-194.

[47] D. J. Berndt and J. Clifford, "Finding Patterns in Time Series: A Dynamic Programming Approach," in *Advances in knowledge discovery and data mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthursamy, Eds. Menlo Park, California: AAAI Press, 1996, pp. 229-248.

[48] E. Keogh and P. Smyth, "A Probabilistic Approach to Fast Pattern Matching in Time Series Databases," proceedings of Third International Conference on Knowledge Discovery and Data Mining, Newport Beach, California, 1997.

[49] E. Keogh, "A Fast and Robust Method for Pattern Matching in Time Series Databases," proceedings of 9th International Conference on Tools with Artificial Intelligence (TAI '97), 1997.

[50] E. J. Keogh and M. J. Pazzani, "An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback," proceedings of AAAI Workshop on Predicting the Future: AI Approaches to Time-Series Analysis, Madison, Wisconsin, 1998.

[51] M. T. Rosenstein and P. R. Cohen, "Continuous Categories For a Mobile Robot," proceedings of Sixteenth National Conference on Artificial Intelligence, 1999.

[52] H. D. I. Abarbanel, R. Brown, J. J. Sidorowich, and L. S. Tsimring, "The analysis of observed chaotic data in physical systems," *Reviews of Modern Physics*, vol. 65, pp. 1331-1392, 1993.

[53] E. W. Minium, *Statistical reasoning in psychology and education*, 2nd ed. New York: Wiley, 1978.

[54] D. J. Sheskin, *Handbook of parametric and nonparametric statistical procedures*. Boca Raton, Florida: CRC Press, 1997.

[55] A. Papoulis, *Probability, random variables, and stochastic processes*, 3rd ed. New York: McGraw-Hill, 1991.

[56] D. G. Luenberger, *Optimization by vector space methods*. New York: Wiley, 1969.

[57] *Using matlab: version 5*. Natick, Massachusetts: The MathWorks, Inc., 1998.

[58] F. K. Reilly and K. C. Brown, *Investment analysis and portfolio management*, 5th ed. Fort Worth, Texas: Dryden Press, 1997.

[59] J. D. Freeman, "Behind the smoke and mirrors: Gauging the integrity of investment simulations," *Financial Analysts Journal*, vol. 48, pp. 26-31, 1992.

[60] J. F. Bangura and N. A. Demerdash, "Simulation of Inverter-Fed Induction Motor Drives with Pulse-Width Modulation by a Time-Stepping Coupled Finite Element-Flux Linkage-Based State Space Model," *IEEE Transactions on Energy Conversion*, vol. 14, pp. 518-525, 1999.

[61] J. F. Bangura and N. A. Demerdash, "Comparison Between Characterization and Diagnosis of Broken Bars/End-Ring Connectors and Airgap Eccentricities of Induction motors in ASDs Using a Coupled Finite Element-State Space Method," *IEEE Transactions on Energy Conversion*, Paper No. PE313ECa (04-99).

[62] N. A. O. Demerdash and J. F. Bangura, "Characterization of Induction Motors in Adjustable-Speed Drives Using a Time-Stepping Coupled Finite-Element State-Space Method Including Experimental Validation," *IEEE Transactions on Industry Applications*, vol. 35, pp. 790-802, 1999.

[63] J. F. Bangura and N. A. O. Demerdash, "Effects of Broken Bars/End-Ring Connectors and Airgap Eccentricities on Ohmic and Core Losses of Induction Motors in ASDs Using a Coupled Finite Element-State Space Method," *IEEE Transactions on Energy Conversion*, Paper No. PE312EC (04-99).

[64] J. F. Bangura, *A Time-Stepping Coupled Finite Element-State Space Modeling for On-Line Diagnosis of Squirrel-Cage Induction Motor Faults*, Ph.D. Dissertation, Marquette University, June 1999.

[65] N. A. Demerdash and J. F. Bangura, "A Time-Stepping Coupled Finite Element-State Space Modeling for Analysis and Performance Quality Assessment of Induction Motors in Adjustable Speed Drives Applications," proceedings of Naval Symposium on Electric Machines, Newport, Rhode Island, 1997, pp. 235-242.