Identifying Temporal Patterns for Characterization and Prediction of Financial Time Series Events

Richard J. Povinelli

Department of Electrical and Computer Engineering, Marquette University,
P.O. Box 1881, Milwaukee, WI 53201-1881, USA
Richard.Povinelli@Marquette.edu
http://povinelli.eece.mu.edu

Abstract. The novel Time Series Data Mining (TSDM) framework is applied to analyzing financial time series. The TSDM 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. This contrasts with other time series analysis techniques, which typically characterize and predict all observations. The TSDM framework and concepts are reviewed, and the applicable TSDM method is discussed. Finally, the TSDM method is applied to time series generated by a basket of financial securities. The results show that statistically significant temporal patterns that are both characteristic and predictive of events in financial time series can be identified.

1 Introduction

The Time Series Data Mining (TSDM) framework [1-4] is applied to the prediction of financial time series. TSDM-based methods can 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.

A time series is "a sequence of observed data, usually ordered in time" [5, p. 1]. Fig. 1 shows an example time series $X = \{x_t, t = 1,\ldots,N\}$, where t is a time index, and $N = 126$ is the number of observations. Time series analysis is fundamental to engineering, scientific, and business endeavors, such as the prediction of welding droplet releases and stock market price fluctuations [1, 2, 4].

This paper, which is divided into four sections, presents the results of applying the TSDM framework to the problem of finding a trading-edge, i.e., a small, but significant, advantage that allows greater than expected returns to be realized. The first section presents the problem and reviews other time series analysis techniques. The second section introduces the key TSDM concepts and method. The third section presents the prediction results. The fourth section discusses the results and proposes future work.

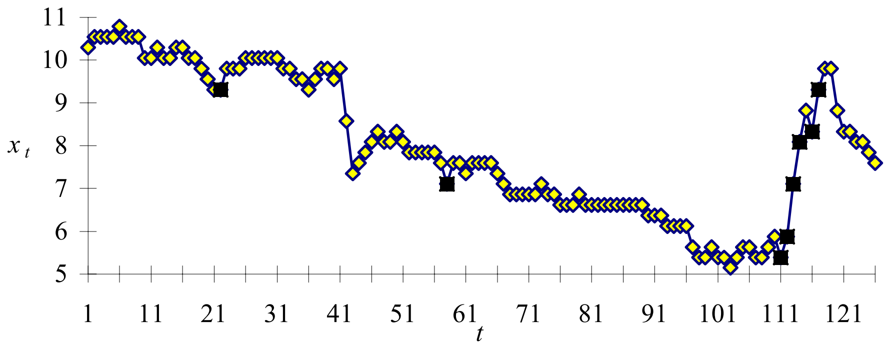

Fig. 1 – Stock Daily Open Price Time Series

1.1 Problem Statement

The predominant theory for describing the price behavior of a financial security is the efficient market hypothesis, which is explained using the expected return or fair game model [6, p. 210]. The expected value of a security is $E(P_{t+1}|\Phi_t) = (1 + E(r_{t+1}|\Phi_t))P_t$ [6, p. 210], 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.

The three forms of the efficient market hypothesis are weak, semistrong, and strong. The weak form, which is relevant to this work, assumes Φ_t is all security-market information, such as historical sequence of price, rates of return, and trading volume data [6, p. 211].

The weak form of the efficient market hypothesis has been supported in the literature [6, p. 213-215]. The efficient market hypothesis has been 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 [6, p. 213-215].

The problem is to find a trading-edge, which is a small advantage that allows greater than expected returns to be realized. If the weak form of the efficient market hypothesis holds, the TSDM method should not be able to find hidden patterns that can be exploited to achieve such a trading-edge.

Fig. 1 illustrates the problem, where the horizontal axis represents time, and the vertical axis observations. The diamonds show the open price of a stock. The results of a successful prediction technique are illustrated by the black squares, which indicate buying opportunities. If the stock were purchased on those days and sold the next day, a greater than 5% return would be realized for each buy-sell sequence.

To summarize, the problem is to find hidden patterns that are, on average, characteristic and predictive of a larger than normal increase in the price of a stock and to use these hidden patterns in a trading strategy.

1.2 Review of Time Series Analysis Techniques

The analysis of financial time series has a long history. This review will briefly touch on some of the many time series analysis techniques that may be applied to predicting stock prices, including ARIMA, machine learning, genetic programming, neural network, and various data mining methods.

Some of the first applications of the traditional Box-Jenkins or Autoregressive Integrated Moving Average (ARIMA) method was to the analysis of the IBM stock time series [5]. The ARIMA techniques provide a comprehensive approach for analyzing stationary time series whose residuals are normal and independent [5]. For real-world time series such as stock market prices, the conditions of time series stationarity and residual normality and independence are not met. Another drawback of the ARIMA approach is its inability to identify complex hidden characteristics. This limitation occurs because of the goal of characterizing all time series observations.

For stock time series, the typical AR(1) model is $x_t = x_{t-1} + a_t$ [5, pp. 30-31], i.e., the expected next value in the time series is the current value. This model does not help in making trading decisions.

An example of applying machine learning techniques is provided by the work of Zemke, who uses a bagging approach to combine predictions made by an artificial neural network, a nearest neighbor method, and an evolved logic program to predict various stock indices [7]. Zemke is able to achieve an average daily excess return of 0.15% more than a random trading strategy.

Kaboudan uses a genetic programming approach to learn the nonlinear generating function to predict stock time series [8]. He develops a trading strategy that is tested against six stocks. Kaboudan is able to achieve an average daily excess return of 0.45% more than a naïve trading approach.

Berndt and Clifford [9], Keogh [10-12], Rosenstein and Cohen [13], and Guralnik et al. [14] are among those who have applied data mining concepts to finding patterns in time series. Data Mining [15, 16] is the analysis of data with the goal of uncovering hidden patterns. It 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. Data mining evolved from several fields, including machine learning, statistics, and database design [16]. 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 [15, 16].

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 [13] 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 [9], 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" [10]. Keogh's approach uses a probabilistic method for matching the known templates to the time series data.

Guralnik et al. [14] have developed a language for describing temporal patterns (episodes) in sequence data. They have developed an efficient sequential pattern tree for identifying frequent episodes. Their work, like that of others discussed here, focuses on quickly finding patterns that match predefined templates.

The novel TSDM framework, initially introduced by Povinelli and Feng in [1], differs fundamentally from both data mining and other time series approaches. The TSDM framework differs from most time series analysis techniques by focusing on discovering hidden temporal patterns that are predictive of events, which are *important* occurrences, rather than trying to predict all observations. 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.

The data mining approaches advanced in [9-14] require *a priori* knowledge of the types of structures or temporal patterns to be discovered. These approaches represent temporal patterns as a set of templates. The use of predefined templates in [9-14] prevents the achievement of the basic data mining goal of discovering useful, novel, and hidden temporal patterns. The TSDM framework is not restricted by the use of predefined templates.

The novel TSDM framework creates a new structure for analyzing time series by adapting concepts from data mining [15, 16]; time series analysis [5, 17, 18]; genetic algorithms [19-21]; and chaos, nonlinear dynamics, and dynamical systems [22-25]. 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 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 TSDM methods, specifically Takens' Theorem [26] and Sauer's extension [27].

2 Some Time Series Data Mining Concepts

Previous work [1, 2, 4] presented the TSDM framework. In this section, the fundamental TSDM concepts such as events, temporal patterns, event characterization function, temporal pattern cluster, time-delay embedding, phase space, augmented phase space, objective function, and optimization are defined and explained as is the TSDM method for identifying temporal pattern clusters.

The TSDM method discussed here discovers hidden temporal patterns (vectors of length Q) characteristic of events (important occurrences) by time-delay embedding [22, 25] an observed time series X into a reconstructed phase space, here simply called *phase space*. An event characterization function g is used to represent the eventness of a temporal pattern. An augmented phase space is formed by extending the phase space with g. The augmented phase space is searched for a temporal pattern cluster P that best characterizes the desired events. The temporal pattern clusters are then used to predict events in a testing time series.

2.1 Events, Temporal Pattern, and Temporal Pattern Cluster

In a time series, an event is an important occurrence. The definition of an event is dependent on the TSDM goal. For example, an event may be defined as the sharp rise or fall of a stock 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 Fig. 1. The events, highlighted as squares in Fig. 1, are those days when the price increases more than 5%.

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 is represented as a point in a Q dimensional real metric space, i.e., $\mathbf{p} \in \mathbb{R}^Q$.

Because a temporal pattern may not perfectly match the time series observations that precede events, a temporal pattern cluster is defined as the set of all points within δ of the temporal pattern. The temporal pattern cluster $P = \left\{ a \in \mathbb{R}^Q : d(\mathbf{p}, a) \le \delta \right\}$, where d is the distance or metric defined on the space. This defines a hypersphere of dimension Q, radius δ, and center **p**.

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.

2.2 Phase Space and Time-Delay Embedding

A reconstructed phase space [22] is a Q-dimensional metric space into which a time series is embedded. Takens showed that if Q is large enough, the phase space is homeomorphic to the state space that generated the time series [26]. 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, $\mathbf{x}_t = \left(x_{t-(Q-1)\tau}, \ldots, x_{t-2\tau}, x_{t-\tau}, x_t \right)^T$.

2.3 Event Characterization Function

To link a temporal pattern (past and present) with an event (future) the 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.

In Fig. 1, the goal is to decide if the stock should be purchased today and sold tomorrow. The event characterization function that achieves this goal is $g(t) = (x_{t+1} - x_t)/x_t$, which assigns the percentage change in the stock price for the next day to the current time index. Alternatively the time series maybe filtered, thereby simplifying the event characterization function.

2.4 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 $< \mathbf{x}_t, g(t) > \in \mathbb{R}^{Q+1}$.

Fig. 2, a stem-and-leaf plot, shows the augmented phase space for the daily return time series generated from the open price time series illustrated in Fig. 1. The height of the leaf represents the significance of $g(\cdot)$ for that time index.

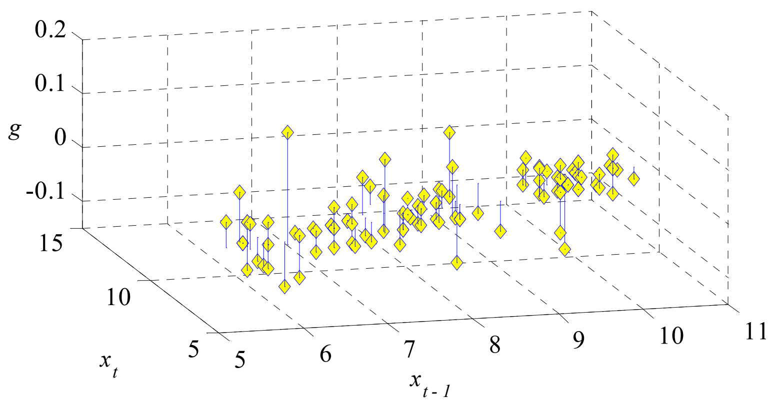

Fig. 2 – Stock Daily Return Augmented Phase Space

2.5 Objective Function

The TSDM objective function 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.

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

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

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

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

where $c(M)$ is the cardinality of M.

The following 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

$$f(P) = \begin{cases} \mu_M & \text{if } c(M)/c(\Lambda) \geq \beta \\ (\mu_M - g_0)\dfrac{c(M)}{\beta c(\Lambda)} + g_0 & \text{otherwise} \end{cases}, \tag{1}$$

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

The parameter β in the linear barrier function in (1) is chosen so that $c\left(M^*\right)$ is non-trivial, i.e., the neighborhood around **p** includes some percentage of the total phase space points. If $\beta = 0$, then $c\left(M^*\right) = 1$ 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 $\beta = 0$, (1) is still defined, because $c(M)/c(\Lambda) \geq 0$ is always true.

2.6 Optimization

The key to the TSDM framework is finding optimal temporal pattern clusters that characterize and predict events. Thus, an optimization algorithm to maximize $f(P)$ over **p** and δ is necessary. A modified simple GA (sGA) [19] composed of a Monte Carlo initialization, roulette selection, and random locus crossover is used for finding P^*. The Monte Carlo search generates the initial population for the sGA. Although a mutation operator is typically incorporated into an sGA, it is not used for discovering the results presented in this paper. The sGA uses a binary chromosome with gene lengths of six and single individual elitism. The stopping criterion for the GA is convergence of all fitness values. The population size is 30 and the Monte Carlo search size is 300. A hashing technique is employed to improve computational performance [28].

2.7 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 $X = \{x_t, t = 1, \ldots, N\}$, the goal is to find hidden temporal patterns that are characteristic of events in *X*, where events are specified in the context of the problem. Likewise, given a testing time series $Y = \{x_t, t = R, \ldots, S\} \quad N < R < S$, the goal is to use the hidden temporal patterns discovered in *X* to predict events in *Y*.

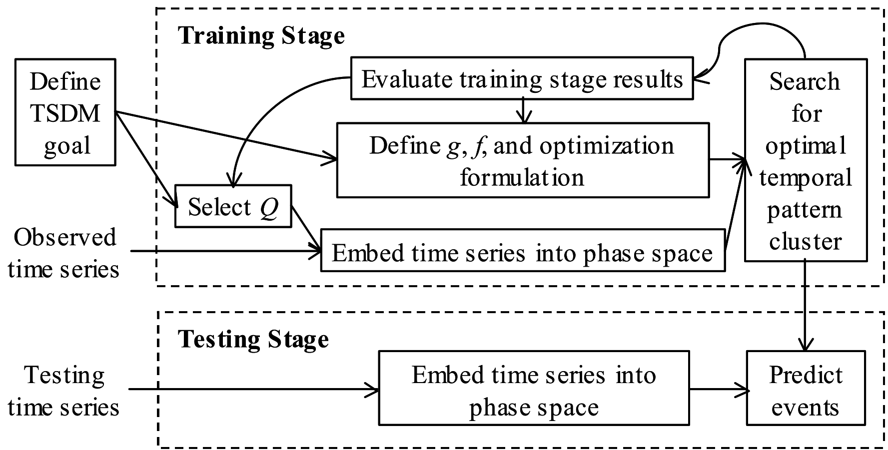

Fig. 3 – Block Diagram of TSDM Method

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:

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.

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.

3 Financial Applications of Time Series Data Mining

This section presents significant results found by applying the Time Series Data Mining (TSDM) method to a basket of financial time series. The time series 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 [6]. The times series are measurements of the activity of a security, specifically a stock.

The 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 method should not be able to find 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.

Two sets of time series are analyzed. The first set of time series are the inter-day returns for the 30 Dow Jones Industrial Average (DJIA) components from January 2, 1990 through March 8, 1991. This time period allows for approximately 200 testing stages. The inter-day return $r_t = (o_{t+1} - o_t)/o_t$, where o_t is the daily open price, which is the price of the first trade. Detailed results for this set of time series are provided.

The second set of time series are the intra-day returns for the 30 DJIA components from October 16, 1998 through December 22, 1999. Again, this time period allows for approximately 200 testing stages. The intra-day return $r_t = (c_t - o_t)/o_t$, where o_t is the daily open price and c_t is the daily closing price, which is the price of the last trade. Summary results are provided for this set of time series.

Fig. 4 illustrates the DJIA during the first time period.

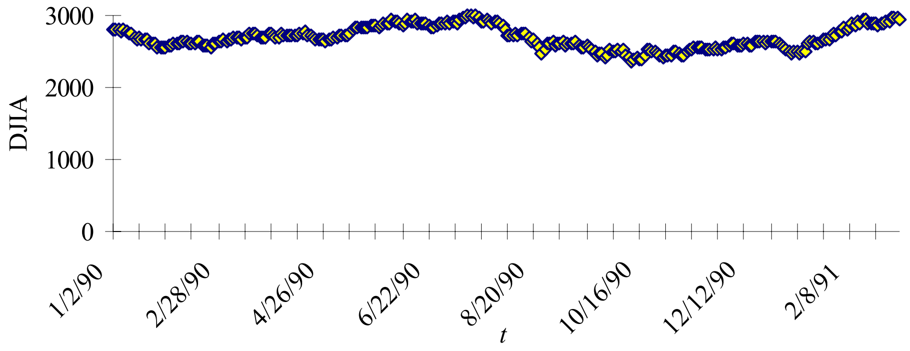

Fig. 4 – DJIA Daily Open Price Time Series

3.1 Training Stage

The TSDM method, illustrated in Fig. 4, is applied 198 times to each of the DJIA component time series for a total of 5,940 training stages for each set of time series. The 198 observed time series are formed from a moving window of length 100. The testing time series is a single observation. The parameters of the method are:

- The event characterization function $g(t) = x_{t+1}$, which allows for one-step-ahead characterization and prediction.
- The objective function (1) with a $\beta = 0.05$ is used.
- The optimization formulation is $\max f(P)$.
- The dimension of the phase space $Q = 2$.

The statistical training results for each DJIA component are presented in Table 1. Of the 5,940 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 (α) was less than 5% based on the independent means statistical test.

Table 1 – DJIA Component Results, January 2, 1990, through March 8, 1991 (Observed)

Ticker	$\mu_M > \mu_X$	$\alpha \leq 0.05$
AA	100%	82%
ALD	100%	72%
AXP	100%	71%
BA	100%	70%
CAT	100%	79%
CHV	100%	54%
DD	100%	42%
DIS	100%	83%
EK	100%	55%
GE	100%	66%

Ticker	$\mu_M > \mu_X$	$\alpha \leq 0.05$
GM	100%	73%
GT	100%	62%
HWP	100%	55%
IBM	100%	67%
IP	100%	80%
JNJ	100%	89%
JPM	100%	90%
KO	100%	67%
MCD	100%	62%
MMM	100%	57%
MO	100%	65%
MRK	100%	59%
PG	100%	76%
S	100%	59%
T	100%	66%
TRV	100%	78%
UK	100%	36%
UTX	100%	94%
WMT	100%	73%
XON	100%	75%
Combined	100%	69%

3.2 Testing Stage

Using the 5,940 training processes, 471 events are predicted for the January 2, 1990 through March 8, 1991 time series. The statistical prediction results for each DJIA component are presented in Table 2. 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. These low rates of statistical significance at the 5% α level are typical for predictions of financial time series [1, 2].

Table 2 – DJIA Component Results, January 2, 1990, through March 8, 1991 (Testing)

Ticker	$c(M)$	μ_M	σ_M	$c(\tilde{M})$	$\mu_{\tilde{M}}$	$\sigma_{M'}$	α_m
AA	16	0.569%	1.652%	182	-0.013%	1.620%	$1.78\text{x}10^{-1}$
ALD	14	0.438%	1.428%	184	-0.102%	1.851%	$1.83\text{x}10^{-1}$
AXP	14	0.027%	2.058%	184	-0.023%	2.610%	$9.32\text{x}10^{-1}$
BA	13	0.080%	2.044%	185	-0.030%	2.181%	$8.52\text{x}10^{-1}$
CAT	26	-0.003%	1.817%	172	-0.098%	2.127%	$8.08\text{x}10^{-1}$
CHV	16	0.057%	1.572%	182	0.061%	1.200%	$9.92\text{x}10^{-1}$
DD	16	0.526%	1.946%	182	-0.045%	1.635%	$2.55\text{x}10^{-1}$
DIS	20	-0.024%	1.488%	178	0.069%	2.069%	$8.00\text{x}10^{-1}$

Ticker	$c(M)$	μ_M	σ_M	$c(\tilde{M})$	$\mu_{\tilde{M}}$	$\sigma_{M'}$	α_m
EK	14	-0.045%	1.879%	184	0.074%	1.998%	$8.20x10^{-1}$
GE	16	0.094%	1.410%	182	0.000%	1.881%	$8.04x10^{-1}$
GM	16	0.671%	2.090%	182	-0.149%	1.863%	$1.29x10^{-1}$
GT	20	-0.962%	2.034%	178	-0.066%	2.549%	$6.93x10^{-2}$
HWP	13	-0.779%	1.881%	185	0.116%	2.664%	$1.08x10^{-1}$
IBM	16	-1.079%	1.785%	182	0.175%	1.460%	$6.32x10^{-3}$
IP	16	1.197%	2.525%	182	0.025%	1.587%	$6.80x10^{-2}$
JNJ	13	0.665%	1.444%	185	0.160%	1.551%	$2.25x10^{-1}$
JPM	11	1.420%	1.878%	187	0.040%	1.985%	$1.82x10^{-2}$
KO	11	1.794%	3.396%	187	0.008%	1.807%	$8.36x10^{-2}$
MCD	13	0.367%	1.753%	185	-0.013%	1.977%	$4.54x10^{-1}$
MMM	16	0.238%	1.044%	182	0.043%	1.258%	$4.82x10^{-1}$
MO	17	0.038%	1.820%	181	0.251%	1.641%	$6.42x10^{-1}$
MRK	19	0.669%	1.163%	179	0.073%	1.580%	$4.11x10^{-2}$
PG	13	0.174%	1.615%	185	0.047%	1.707%	$7.85x10^{-1}$
S	14	1.449%	2.677%	184	-0.157%	1.938%	$2.77x10^{-2}$
T	11	1.307%	1.797%	187	-0.193%	1.645%	$6.88x10^{-3}$
TRV	21	1.531%	2.449%	177	-0.147%	2.617%	$3.21x10^{-3}$
UK	14	-0.449%	2.263%	184	0.041%	1.900%	$4.30x10^{-1}$
UTX	14	-0.289%	1.979%	184	-0.028%	1.828%	$6.33x10^{-1}$
WMT	18	0.658%	1.950%	180	0.120%	2.458%	$2.77x10^{-1}$
XON	20	0.077%	1.398%	178	0.090%	1.263%	$9.68x10^{-1}$
All	471	0.313%	1.970%	5,469	0.011%	1.919%	$1.38x10^{-3}$

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.

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 3. This period is shorter 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.

Table 3 – Trading Results, May 29, 1990 through March 8, 1991

Portfolio	Investment Method	Return	Annualized Return
All DJIA components	Temporal Pattern Cluster	30.98%	41.18%
DJIA	Buy and Hold	2.95%	3.79%
All DJIA components	Not in Temporal Pattern Cluster	0.35%	0.45%
All DJIA components	Buy and Hold	3.34%	4.29%

The results for the investment period of March 15, 1999 through December 22, 1999 are shown in Table 4. Again, this period is shorter than the total time frame (October 16, 1998 through December 22, 1999) because the first part of the time series is used only for training. The return of the DJIA varies significantly from the buy and hold strategy for all DJIA components not only because the DJIA has a non-equal weighting among its components, but more importantly because intra-day return time series are used. The results for this set of time series is less significant than the previous with an $\alpha = 0.12$.

Table 4 – Trading Results, March 15, 1999 through December 22, 1999

Portfolio	Investment Method	Return	Annualized Return
All DJIA components	Temporal Pattern Cluster	22.70%	29.88%
DJIA	Buy and Hold	13.39%	17.42%
All DJIA components	Not in Temporal Pattern Cluster	-10.92%	-13.74%
All DJIA components	Buy and Hold	-8.26%	-10.43%

An initial investment of $10,000 made on May 29, 1990 in the 30 DJIA component stocks using the TSDM method would have grown to $13,098 at the end of March 8, 1991. The maximum draw down, the largest loss during the investment period, is 9.65%. An initial investment of $10,000 made on March 15, 1999 using the TSDM method would have grown to $12,700 at the end of December 22, 1999 with a maximum draw down of 10.2%. One caveat to this result is that it ignores trading costs [29]. 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, where the bid is the offer price for buying and the ask is the offer price for selling. The trading cost in percentage terms would need to be kept in the 0.02% range. 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.

4 Conclusions and Future Work

Through the novel Time Series Data Mining (TSDM) framework and its associated method, this paper has made an original 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.

The TSDM method was successfully applied to characterizing and predicting complex, nonstationary time series events from the financial domain. In the financial domain, it was able to generate a trading-edge.

Future research efforts will involve the direct comparison over the same time periods of the TSDM method present here with the techniques proposed by Zemke [7] and Kaboudan [8]. Additional comparisons with Hidden Markov Model techniques also will be investigated. A detailed study of the risk-return characteristics of these various methods will be undertaken.

Additionally, new time series predictability metrics will be created that specifically address the event nature of the TSDM framework. This research direction will study the characteristics of time series that allow for the successful application of the TSDM framework.

References

[1] 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.

[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, "Using Genetic Algorithms to Find Temporal Patterns Indicative of Time Series Events," proceedings of Genetic and Evolutionary Computation Conference (GECCO-2000) Workshop Program, Las Vegas, Nevada, 2000, pp. 80-84.

[4] R. J. Povinelli, *Time Series Data Mining: Identifying Temporal Patterns for Characterization and Prediction of Time Series Events*, Ph.D. Dissertation, Marquette University, 1999.

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

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

[7] S. Zemke, "Bagging imperfect predictors," proceedings of Artificial Neural Networks in Engineering, St. Louis, Missouri, 1999, pp. 1067-1072.

[8] M. Kaboudan, "Genetic Programming Prediction of Stock Prices," *Computational Economics*, to appear.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] V. Guralnik, D. Wijesekera, and J. Srivastava, "Pattern Directed Mining of Sequence Data," proceedings of International Conference on Knowledge Discovery and Data Mining, New York, NY, 1998, pp. 51-57.

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

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

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

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

[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. 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.

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

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

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

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

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

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

[28] R. J. Povinelli, "Improving Computational Performance of Genetic Algorithms: A Comparison of Techniques," proceedings of Genetic and Evolutionary Computation Conference (GECCO-2000) Late Breaking Papers, Las Vegas, Nevada, 2000, pp. 297-302.

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