A New Temporal Pattern Identification Method for Characterization and Prediction of Complex Time Series Events

Richard J. Povinelli, *Senior Member*, *IEEE*, and Xin Feng, *Senior Member*, *IEEE*

Abstract—A new method for analyzing time series data is introduced in this paper. Inspired by data mining, the new method employs time-delayed embedding and identifies temporal patterns in the resulting phase spaces. An optimization method is applied to search the phase spaces for optimal heterogeneous temporal pattern clusters that reveal hidden temporal patterns, which are characteristic and predictive of time series events. The fudemantal concepts and framework of the method are explained in detail. The method is then applied to the characterization and prediction, with a high degree of accuracy, of the release of metal droplets from a welder. The results of the method are compared to those from a Time Delay Neural Network and the C4.5 decision tree algorithm.

Index Terms—Temporal pattern identification, time series analysis, data mining, time delay embedding, optimization clustering, and genetic algorithms.

1 Introduction

1.1 Problem Statement

Consider a time series

$$X = \{x_t,\ t = 1, \dots, N\},$$

where t is the time index and N is the total number of observations. Such a time series being examined often contains events of interest, and we wish to discover them in a timely manner. For instance, investors want to predict the future value of certain securities. One of the approaches is to identify time-ordered structures (called temporal patterns) in the time series that are characteristic and predictive of the events of interest.

Fig. 1 illustrates an example problem. The time series represents simulated seismic activity generated from a randomly occurring synthetic earthquake contaminated with a noise signal. The square markers indicate events of interest. The temporal patterns that are predictive of the events of interest are a prototype of the subsequences of the time series consisting of data points before the events.

However, in a variety of applications, many of the significant temporal patterns are unobvious, contaminated with noise, or hidden in the data and, hence, are difficult to identify using traditional time series analysis methods, such as the well-known Box-Jenkins method. The Box-Jenkins method is limited by the requirement of stationarity of the time series, and normality and independence of the residuals [1]. For real-world time series such as welding droplet releases and stock market prices, the stationarity and independence conditions are not met.

New research has been devoted to the finding of innovative methods for the identification of temporal patterns that characterize the events of interest in the time series. Among them, data mining is one of the promising technologies dealing with temporal pattern identification in the time series.

1.2 Review of Data Mining in Time Series

As an emerging discipline, data mining is the process of discovering useful patterns in data that are hidden and unknown in normal circumstances. It stems from several fields, including machine learning, statistics, and database design [2]. It uses techniques, such as, clustering, association rules, visualization, and probabilistic graphical dependency models to identify hidden, and useful structures in large databases [2], [3]. The patterns of the data being discovered are in a variety of formats, including spatial and temporal patterns.

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 [2]." 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 [4]."

Others who have applied data mining concepts to finding patterns in time series include Berndt and Clifford [5], Keogh and Smith [6], [7], [8], Rosenstein and Cohen [9], Guralnik et al. [10], Faloutsos et al. [11], Yi et al. [12], Agrawal et al. [13], and Faloutsos et al. [14]. 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 [9] also use a predefined set of templates to match a time series generated from robot sensors. Instead of using the dynamic programming

- *The authors are with the Department of Electrical and Computer Engineering, Marquette University, PO Box 1881, Milwaukee, Wisconsin 53201. E-mail: {Richard.Povinelli, Xin.Feng}@Marquette.edu.*

Manuscript received 31 May 2000; revised 17 Apr. 2001; accepted 4 Sept. 2001.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 112212.

1041-4347/03/$17.00 © 2003 IEEE Published by the IEEE Computer Society

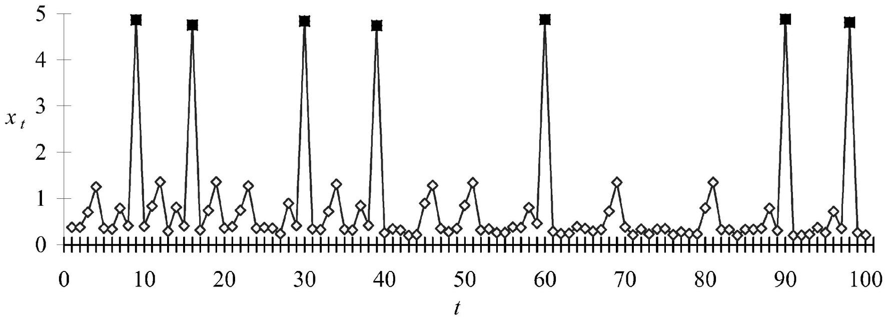

Fig. 1. Synthetic seismic time series with events.

methods as in [5], they employ the time-delay embedding process to match their predefined templates.

Similarly, Keogh and Smith represent 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 [6]." Keogh and Smith's approach uses a probabilistic method for matching the known templates to the time series data.

Guralnik et al. [10] 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.

It should be pointed out that the approaches proposed in [5], [6], [7], [8], [9], [10] require a priori knowledge of the structures or temporal patterns to be discovered and represent these temporal patterns as a set of templates.

Faloutsos et al. [11], [14], Yi et al. [12], and Agrawal et al. [13] have developed highly efficient, with respect to time and space, algorithms for extracting similar subsequences based on a given query from a sequence database. Their initial work [13] uses the Discrete Fourier Transform to generate a small set of Fourier coefficients for efficient indexing. Faloutsos et al. [11], [14] and Yi et al. [12] extend this work by integrating r*-tree (a spatial access method) and dynamic time warping techniques.

1.3 Outline of the Proposed Method

Inspired by concepts in data mining and dynamical systems, this paper introduces a new method for identifying temporal patterns in time series that are significant for characterizing and predicting events, i.e., the important occurrences. The new method is capable of characterizing temporal patterns of complex time series, which are often nonperiodic, irregular, and chaotic. This method identifies predictive temporal structures in reconstructed phase spaces [15], [16]. A genetic algorithm searches such phase spaces for optimal heterogeneous (varying dimension) clusters that are predictive of the desired events.

The framework, initially introduced in [17], differs fundamentally from the approaches mentioned above. Instead of predefining the temporal patterns subjectively, the method applies an optimization approach to search for the optimal temporal patterns, which match the specific goal of the problem. This is accomplished by defining a problem specific event characteristic function $g(\cdot)$ to formalize the concept of eventness. The search is performed in reconstructed phase spaces created using time-delayed embedding [15], [16]. In the optimization, a genetic algorithm is used to search for points in the phase space that are predictive of the events of interest.

There are several significant features of the proposed method. First, the method focuses on the identification of the temporal patterns that are characteristic of the events. Second, with the temporal patterns identified, the new method focuses on event prediction rather than complete time series prediction. This allows the prediction of complicated time series events such as the release of metal droplets from a welder. Third, the objective function in the optimization reflects the goal of the time series being examined, i.e., droplet releases, and is problem specific.

A brief outline of the method is given here, with a detailed description presented in the following sections. Given a training time series $X = \{x_t,\ t = 1, \ldots, N\}$, the method is as follows:

Step A. The time series X is unfolded [15] into $\mathbb{R}^Q$—a reconstructed phase space, called simply phase space here—using time-delayed embedding [15], [18], [19]. The unfolding mechanism maps X into $\mathbb{R}^Q$. Specifically, a set of Q time series observations $\{x_{t-(Q-1)\tau}, \ldots, x_{t-2\tau}, x_{t-\tau}, x_t\}$ taken from X map to

$$\mathbf{x}_t = \left(x_{t-(Q-1)\tau}, \ldots, x_{t-2\tau}, x_{t-\tau}, x_t\right)^T,$$

where $\mathbf{x}_t$ is a column vector or point in the phase space, τ is the time delay, and t is an integer in the interval $[(Q-1)\tau + 1, N]$.

Step B. A real valued function $g(\mathbf{x}_t)$, the so-called "event characterization function," is defined and associated with each phase space point $\mathbf{x}_t$. The event characterization function represents the value of future "eventness" for the phase space point $\mathbf{x}_t$.

Step C. Construct a heterogeneous (in the sense that Q may take multiple values) collection of temporal pattern clusters $\mathcal{C}^*$, such that, $\mathcal{C}^*$ is the optimizer of the objective function f, where a temporal pattern cluster P is defined as a ball consisting of all points within a certain distance δ of a temporal pattern $\mathbf{p}$ in the aforementioned $\mathbb{R}^Q$ phase space and the temporal pattern $\mathbf{p}$ is a $Q \times 1$ vector in the same $\mathbb{R}^Q$ phase space. The objective function f maps a collection of temporal pattern clusters $\mathcal{C}$ onto the real line, thereby providing an ordering to collections of temporal pattern clusters according to their ability to

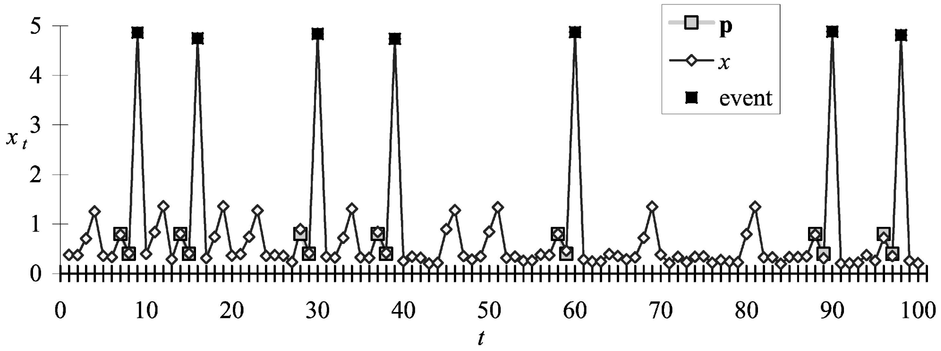

Fig. 2. Synthetic seismic time series with temporal pattern and events.

characterize events. The objective function f is constructed in such a manner that its optimizer $\mathcal{C}^*$ is predictive of the events of interest. An event is then predicted whenever a phase space point $\mathbf{x}_t$ formed from a set of Q time series observations $\{x_{t-(Q-1)\tau}, \ldots, x_{t-2\tau}, x_{t-\tau}, x_t\}$ is within one of the temporal pattern clusters P that comprise $\mathcal{C}^*$.

In the next section, the important concepts of the method are presented in detail. In Section 3, the framework is presented and a clarifying example is discussed. Section 4 presents the application of our method to the prediction of welding droplet releases[1] and these results are compared with those obtained using the C4.5 decision tree algorithm [20] and a time delayed neural network (TDNN) [21]. The final section summarizes the results and discusses future directions.

2 Fundamental Concepts

2.1 Event, Temporal Pattern, and Temporal Pattern Cluster

In a time series, an event is an important occurrence. Importance is application dependent. In a seismic time series, for instance, an earthquake is defined as an event, as illustrated in Fig. 1. Other examples of events include sharp rises or falls of a stock price or the release of a droplet of metal from a welder.

Fig. 2 shows the synthetic seismic time series with contaminating noise. Recall that a temporal pattern $\mathbf{p}$ is a $Q \times 1$ vector in $\mathbb{R}^Q$. The temporal pattern that is hidden in the synthetic seismic time series is highlighted. 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 employed to capture the variability of a temporal pattern. A temporal pattern cluster P is defined as a neighborhood of $\mathbf{p}$, consisting of all points within a certain distance δ of $\mathbf{p}$. Hence, $P = \{a \in \mathbb{R}^Q : d(\mathbf{p}, a) \leq \delta\}$, where d is the distance metric defined on the space.

The observations

$$\{x_{t-(Q-1)\tau}, \ldots, x_{t-2\tau}, x_{t-\tau}, x_t\}$$

form a subsequence of the time series X that can be compared to a temporal pattern, where x_t represents the present 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 these 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 subsequences of time series to be easily compared to temporal patterns.

2.2 Phase Space and Time-Delay Embedding

A reconstructed phase space [15], [18], [19], is a Q-dimensional metric space into which a time series is unfolded. Takens showed that if Q is large enough, the phase space is homeomorphic to the state space that generated the time series [22]. Takens' Theorem provides the theoretical justification for reconstructing state spaces using time-delay embedding. Takens proved that the state space of an unknown system can be reconstructed [16], [22].

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 [23]. 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 [15], [16].

It is important to note that the time-delay embedding process does not necessarily create an embedding. A sufficient condition for the phase space to be an embedding of the state space from which X was sampled is that Q is greater than twice the dimension of the original state space [22]. If the phase space is not an embedding of the original state space, it is an immersion.

The difficulty in the time-delay embedding process is in estimating Q, the original state space dimension. Estimating Q is more difficult when the original time series contains both stochastic and deterministic signals since the stochastic component may require that Q be infinite. However, as shown in [15], [17], [23], [24], useful information can be extracted from the reconstructed state space even if its dimension is less than $2m + 1$, where m is the dimension of the state space of the original system.

[1] Drs. C. Tolle, E. Larsen, D. Pace, and D. Iosty of INEEL gathered the data used in this paper. Their work was supported by the US Department of Energy, Office of Science, Office of Basic Energy Sciences, Division of Materials and Materials Engineering, under the US Department of Energy, Idaho Operations Office Contract DE-AC07-94ID13223.

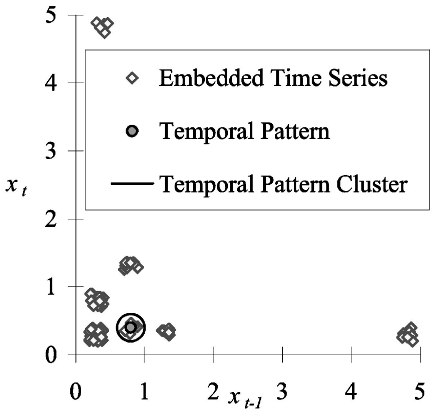

Fig. 3. Synthetic seismic phase space.

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 spaces can be as valid as those from the original state space.

Fig. 3 illustrates the two-dimensional phase space generated by time-delay embedding the time series X from Fig. 2. Also, presented are the temporal pattern and temporal pattern cluster. Although our algorithm places no restriction on Q, for illustrative purposes Q is set to two in Fig. 3.

Although a single temporal pattern cluster is illustrated in Fig. 3, the proposed method can easily model irregularly shaped regions with a collection of temporal pattern clusters. Such a region may be required for patterns that depend on some (but not all Q) points in the past. The proposed method can overcome this problem by discovering collections of temporal pattern clusters that approximate hyperplanes parallel to the phase space axes that are not predictive of the event. If the subset of the previous Q points, which are predictive of an event, is variable, the search problem is harder, but still possible. The proposed method simply discovers each possible predictive structure.

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

2.3 Event Characterization Function

In order to correlate a temporal pattern (past and present) with an event (future), the event characterization function $g(\cdot)$ is introduced. The event characterization function represents the value of future "eventness" for the present time index. The event characterization function is defined a priori and is created to address the specific goal of the time series being considered. The event characterization function is defined such that its value at time t correlates highly with the occurrence of an event at some specified time in the future.

For the time series illustrated in Fig. 1, the goal is to predict occurrences of synthetic earthquakes. One possible event characterization function to address this goal is $g(\mathbf{x}_t) = x_{t+1}$, 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 $g(\mathbf{x}_t) = x_{t+3}$. A more complex event characterization function that would predict an event occurring one, two, or three time-steps ahead is $g(\mathbf{x}_t) = \max\{x_{t+1}, x_{t+2}, x_{t+3}\}$.

The choice of g can significantly affect the results of the method. For example, the selection of the event characterization function $g(\mathbf{x}_t) = x_{t+3}$, for the synthetic earthquake problem presented in Fig. 1 will yield poor prediction results, as no predictive structure exists three time steps before the event. However, the converse problem of using $g(\mathbf{x}_t) = x_{t+1}$ when the predictive structures exist three time steps before the events is not as difficult to resolve. The method will find a collection of temporal pattern clusters that ignores the first two time steps, while focusing on the third time step.

In practice, the selection of g has required little attention, as good solutions to the prediction problems we have studied have been achieved without adjusting g. We have found that the event characterization function $g(\mathbf{x}_t) = x_{t+1}$ is a good starting point.

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

Fig. 4, a stem-and-leaf plot, shows the augmented phase space for the synthetic seismic time series. The height of the leaf represents the value of $g(\cdot)$ for $\mathbf{x}_t$. From this plot, the required temporal pattern and temporal pattern cluster are easily identified.

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.

2.5 Objective Function

The objective function represents the efficacy of a collection of temporal pattern clusters to characterize events. The objective function f maps a collection of temporal pattern clusters $\mathcal{C}$ onto the real line, thereby providing an ordering to collections of temporal pattern clusters according to their ability to characterize events. The objective function f is constructed in such a manner that its optimizer $\mathcal{C}^*$ is predictive of the events of interest.

Fig. 4 illustrates the requirement of an objective function. The collection of temporal pattern clusters $\mathcal{C}_1$ that contains the single temporal pattern cluster P_1 is obviously the best collection for identifying events, while the collection of temporal pattern clusters $\mathcal{C}_2$ that contains the single temporal pattern cluster P_2 is not. The objective function must map the collection of temporal pattern clusters such that $f(\mathcal{C}_1) > f(\mathcal{C}_2)$.

The form of the objective functions is application dependent, and several different objective functions may achieve the same goal. We next present three example objective functions. The first is based on the familiar t-test for the difference between two independent means [25] and is useful for identifying a single temporal pattern cluster

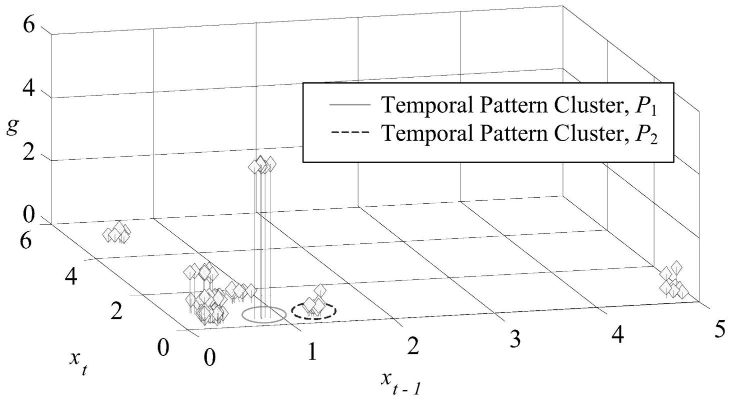

Fig. 4. Synthetic seismic augmented phase space with highlighted temporal pattern clusters.

that is statistically significant. The second objective function is useful for minimizing false positive predictions. The third example objective function is useful for maximizing classification/prediction accuracy.

2.5.1 Objective Function Based on the t-Test

The first objective function is the *t*-test for the difference between two independent means [25] and is useful for identifying a single temporal pattern. In other words, the specific format of the objective function is the statistical value, which is useful for identifying statistically significant temporal pattern clusters with a high average eventness.

Let the cardinality of P be defined as

$$c(P) = \sum_{\mathbf{x}_t \in P} 1,$$

and the cardinality of $\tilde{P}$ be defined as

$$c(\tilde{P}) = \sum_{\mathbf{x}_t \notin P} 1,$$

where $\tilde{P} = \{\mathbf{x}_t : \mathbf{x}_t \notin P\}$. Let the average value of g, also called the average eventness, of the phase space points in the temporal pattern cluster P be

$$\mu_P = \frac{1}{c(P)} \sum_{\mathbf{x}_t \in P} g(\mathbf{x}_t).$$

Let the average eventness of the phase space points not in P be

$$\mu_{\tilde{P}} = \frac{1}{c(\tilde{P})} \sum_{\mathbf{x}_t \notin P} g(\mathbf{x}_t).$$

Let the average eventness of all phase space points be given by

$$\mu_{\mathbf{x}_t} = \frac{1}{N-(Q-1)\tau} \sum g(\mathbf{x}_t),$$

where $(Q-1)\tau$ is the largest embedding time-delay, and N is the number of observations in the time series.
Let the corresponding variances be

$$\sigma_P^2 = \frac{1}{c(P)} \sum_{\mathbf{x}_t \in P} \left(g(\mathbf{x}_t) - \mu_P\right)^2,$$

$$\sigma_{\tilde{P}}^2 = \frac{1}{c(\tilde{P})} \sum_{\mathbf{x}_t \notin P} \left(g(\mathbf{x}_t) - \mu_{\tilde{P}}\right)^2,$$

and

$$\sigma_{\mathbf{x}_t}^2 = \frac{1}{N-(Q-1)\tau} \sum \left(g(\mathbf{x}_t) - \mu_{\mathbf{x}_t}\right)^2.$$

We define the objective function as

$$f(P) = \frac{\mu_P - \mu_{\tilde{P}}}{\sqrt{\frac{\sigma_P^2}{c(P)} + \frac{\sigma_{\tilde{P}}^2}{c(\tilde{P})}}}.$$

This objective function is useful for identifying the temporal patterns present in the synthetic seismic time series illustrated in Fig. 2. It is also useful for identifying temporal pattern clusters hidden in financial time series.

2.5.2 Objective Function to Minimize False Positives

Whereas the first objective function presented above is useful for finding a single statistically significant temporal pattern cluster, the next objective function is useful for finding a single temporal pattern cluster that minimizes the incorrect positive predictions. It is useful in prediction problems where the events belong to various classes. This objective function will be used in the prediction of welding droplet releases where a particular time index of the time series either is or is not a released droplet. It is used in combination with the third objective function presented below.

The second objective function is

$$f(P) = \frac{tp}{tp + fp},$$

where the values *tp*, *tn*, *fp*, and *fn* are described in Table 1 for the case of binary classes. It is the positive accuracy of a temporal pattern cluster in predicting events.

This objective function can be used in combination with the next objective function as an intermediate step in finding a collection of temporal pattern clusters that maximize prediction accuracy.

2.5.3 Objective Function to Maximize Characterization/Prediction Accuracy

The following objective function is useful for classification and prediction problems where accuracy of prediction is of

TABLE 1
Event Categorization

	Actually an event	Actually a nonevent
Categorized as an event	True positive, *tp*	False positive, *fp*
Categorized as a nonevent	False negative, *fn*	True negative, *tn*

primary importance and the events of interest belong to various classes. It is used to determine the efficacy of a collection of temporal pattern clusters in total prediction accuracy.

The third objective function is

$$f(\mathcal{C}) = \frac{tp + tn}{tp + tn + fp + fn},$$

where $\mathcal{C}$ is a collection of temporal pattern clusters. It has an optimal value when the class of every event is correctly predicted.

3 Framework of the Method

Now, with the fundamental concepts explained, this section shows how the proposed method, outlined previously in Section 1.3—Outline of the Proposed Method, is expanded to provide a framework for identifying temporal patterns predictive of the events of interest. The framework is then applied to the example problem of predicting synthetic earthquakes in the time series illustrated in Fig. 1.

3.1 The Method

The basic concept of applying the new 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 temporal pattern clusters for the important events. This allows for the prediction of complex real-world time series using phase spaces with minimal dimensions.

The first step in applying the method is to define the 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 pattern clusters that are characteristic of events in X, where events are specified in the context of the goal. Likewise, given a testing time series $Y = \{x_t, t = R, \ldots, S\}$, the goal is to use the hidden temporal pattern clusters discovered in X to predict events in Y.

Fig. 5 presents a block diagram of the method. Given a goal, a training portion of the time series to be characterized, and the testing portion of the time series to be examined, the steps in the method are:

1. Training Stage (Batch Process).

    a. Model the goal in terms of the event characterization function $g(\cdot)$ and objective function f, and formulate it into the a constrained optimization problem.

    b. Determine the range of Q, i.e., the dimensions of the phase spaces and the lengths of the temporal patterns. Assign the value of τ.

    c. (Step A in Section 1.3 —Outline of the Proposed Method). The time series X is unfolded, which is defined in [15], into $\mathbb{R}^Q$—a reconstructed phase space—using time-delayed embedding [15], [18], [19].The unfolding mechanism maps X into $\mathbb{R}^Q$. Specifically, a set of Q time series observations

    $$\{x_{t-(Q-1)\tau}, \ldots, x_{t-2\tau}, x_{t-\tau}, x_t\}$$

    taken from X map to

    $$\mathbf{x}_t = \left(x_{t-(Q-1)\tau}, \ldots, x_{t-2\tau}, x_{t-\tau}, x_t\right)^T,$$

    where $\mathbf{x}_t$ is a column vector or point in the phase space, τ is time delay, and t is an integer in the interval $[(Q-1)\tau + 1, N]$.

    d. (Step B in Section 1.3—Outline of the Proposed Method). Form the augmented phase spaces. A real valued function $g(\mathbf{x}_t)$, the so-called "event characterization function," is defined and associated with each phase space point $\mathbf{x}_t$. The event characterization function represents the value of future "eventness" for the phase space point $\mathbf{x}_t$.

    e. (Step C in Section 1.3—Outline of the Proposed Method) Search for the optimal collection of temporal pattern clusters in the augmented phase space, where the collection is optimal in the sense that it is the optimizer of the objective

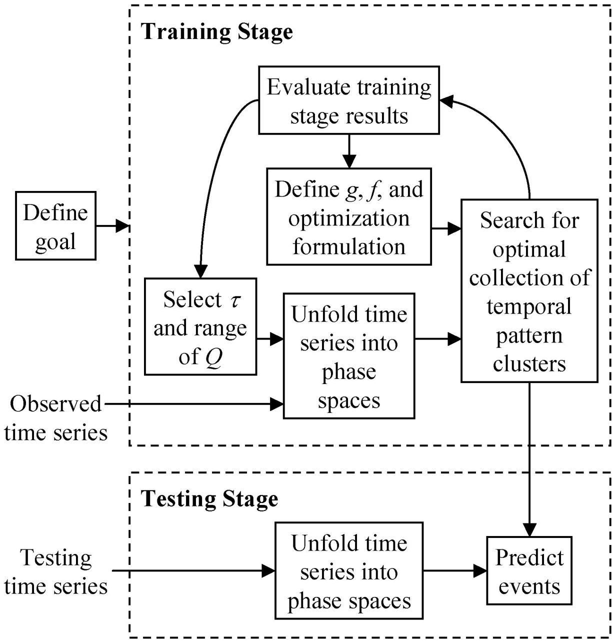
Training Stage
Evaluate training stage results
Define goal
Define g, f, and optimization formulation
Search for optimal collection of temporal pattern clusters
Select τ and range of Q
Unfold time series into phase spaces
Observed time series
Testing Stage
Testing time series
Unfold time series into phase spaces
Predict events

Fig. 5. Block diagram of algorithm.

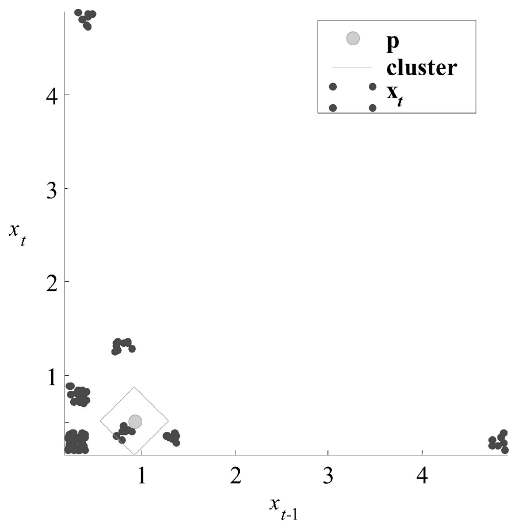

Fig. 6. Synthetic seismic phase space with temporal pattern cluster (training).

function. Construct a heterogeneous (in the sense that Q may take multiple values) collection of temporal pattern clusters $\mathcal{C}^*$, such that $\mathcal{C}^*$ is the optimizer of the objective function f.

f. Evaluate training stage results. Repeat training if necessary.

2. Testing Stage (Real Time or Batch Process).

   a. Embed the testing time series into the phase space.
   b. Use the optimal collection of temporal pattern clusters for identifying and predicting events. An event is predicted whenever a phase space point $\mathbf{x}_t$ formed from a set of Q time series observations $\{x_{t-(Q-1)\tau}, \ldots, x_{t-2\tau}, x_{t-\tau}, x_t\}$ is within one of the temporal pattern clusters P that comprise $\mathcal{C}^*$.
   c. Evaluate testing stage results.

3.2 Search Process

There are two forms of the search. The first is for a single optimal temporal pattern cluster. This is done using a genetic algorithm. The second is for an optimal collection of temporal pattern clusters. This is done using a greedy method built upon the first form of the search.

3.2.1 Genetic Algorithm Search for a Single Optimal Temporal Pattern Cluster

A variant of the well-known simple Genetic Algorithm (sGA) [26] is employed here to search for a single optimal temporal pattern cluster P^*. The sGA is augmented with the tabu search characteristic of storing previously evaluated fitness values. Additionally, the initial population is created though the elitist mechanism of generating a population of size $s \cdot T$, where T is the size of the desired population, and taking the top $1/s\mathrm{th}$ of the large generated population as the initial population. This sGA variant has been previously presented in [27], [28]. It uses a hash table to store previously calculated fitness values, thereby achieving a 2-4x computational speedup without sacrificing accuracy.

The objective/fitness function used by the genetic algorithm is designed to achieve the overall goal of the framework. Applicable objective/fitness functions include

$$f(P) = \frac{\mu_P - \mu_{\tilde{P}}}{\sqrt{\frac{\sigma_P^2}{c(P)} + \frac{\sigma_{\tilde{P}}^2}{c(\tilde{P})}}}$$

and

$$f(P) = \frac{tp}{tp + fp},$$

both of which were presented in the previous section.

The phenotype for the GA, $P = [\mathbf{p} \quad \delta]$, is encoded as a binary string. The decoding of the genotype is defined as

$$p_i = \frac{p_{\max} - p_{\min}}{2^l - 1} \sum_{j=0}^{l-1} 2^j p_{i,j} + p_{\min},$$

where l is the length of the allele (subsequence of the chromosome) used to encode p_i, $p_{\max} = \max X$,

$$p_{\min} = \min X,$$

and X is the training time series. The radius is defined as

$$\delta = \frac{\delta_{\max}}{2^l - 1} \sum_{j=0}^{l-1} 2^j \delta_j,$$

where $\delta_{\max} = Q(p_{\max} - p_{\min})$ and Q is the dimension of $\mathbf{p}$.

A tournament of size two is used as the selection mechanism. Mutation in the range of 0-0.05 percent is used. The stopping criterion is convergence of the fitness values. Elitism of one is employed.

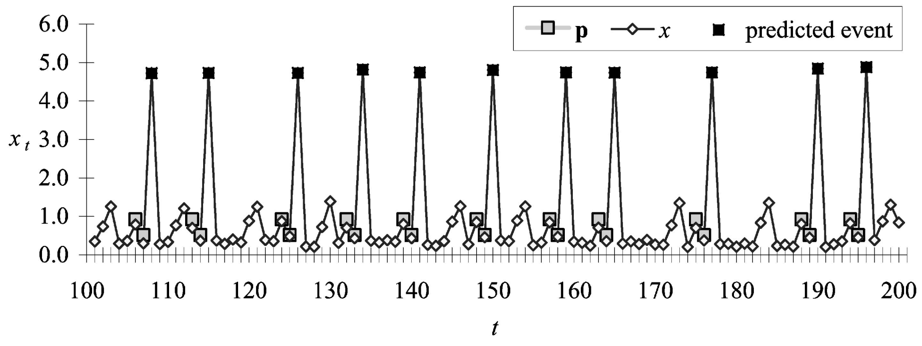

Fig. 7. Synthetic seismic time series with temporal patterns and predicted events highlighted (testing).

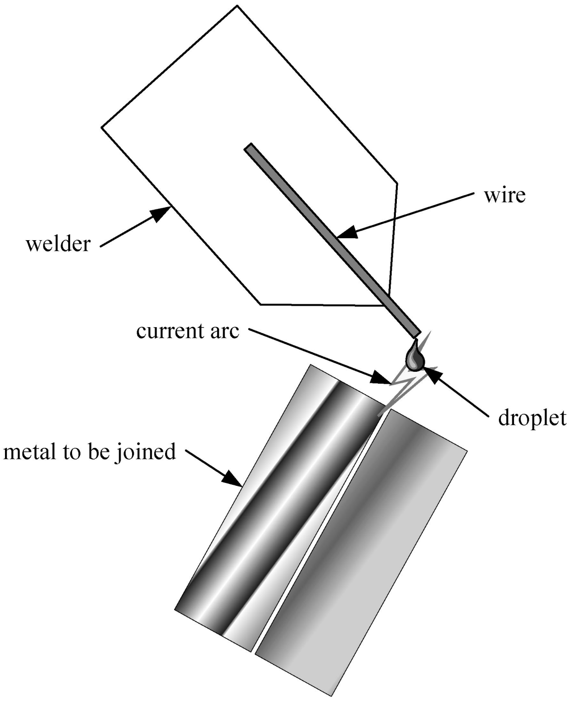

Fig. 8. Cartoon of a welder.

3.2.2 Greedy Search for an Optimal Collection of Temporal Pattern Clusters

We search for a collection of optimal temporal pattern clusters using a greedy approach. This is done to reduce the dimensionality of the search. Let $Q_{\min}$ and $Q_{\max}$ be the minimum and maximum time-delay embedding dimension, respectively. Let $f(\mathcal{C})$ be the objective function for the collection of temporal pattern clusters. An example of such an objective function is

$$f(\mathcal{C}) = \frac{tp + tn}{tp + tn + fp + fn},$$

which was presented in the previous section. The greedy search is performed as follows:

```
set $i = Q_{\min}$
while $i \leq Q_{\max}$
        search for an optimal temporal pattern
            cluster $P$ in the $\mathbb{R}^i$ phase space using the
            genetic algorithm described above.
    if $f(\mathcal{C} \cup \{P\}) > f(\mathcal{C})$ then
        set $\mathcal{C} = (\mathcal{C} \cup \{P\})$
        remove all $\mathbf{x}_t \in P$ from the phase space
    else
        $i = i + 1$;
```

A problem with multiple classes of events can be solved using the proposed method by searching for a family of temporal pattern clusters, where each element of the family is a collection of temporal pattern clusters predictive of a particular event class.

3.3 An Example

This example illustrates how the framework is applied to the synthetic seismic time series illustrated in Fig. 1. We define the goal to be to identify a single temporal pattern cluster predictive of the large spikes in the time series, e.g., the "earthquakes."

3.3.1 Training Stages

Step 1—Model the Goal.

Since the goal is to identify a single temporal pattern cluster predictive of the synthetic earthquakes, the event characterization function is $g(\mathbf{x}_t) = x_{t+1}$, which allows one-step prediction. Since the temporal pattern cluster that characterizes the events is to be statistically significant from other temporal patterns, the objective function used is

$$f(P) = \frac{\mu_P - \mu_{\tilde{P}}}{\sqrt{\frac{\sigma_P^2}{c(P)} + \frac{\sigma_{\tilde{P}}^2}{c(\tilde{P})}}},$$

which orders temporal pattern clusters according to their ability to statistically differentiate between events and nonevents. The optimization formulation is: $\max f(P)$.

Step 2—Determine Temporal Pattern Length.

The value of Q, i.e., the length of the temporal pattern and the dimension of the phase space, is selected based on Takens' Theorem, which states 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. Since Takens' Theorem provides no mechanism for determining m, using the principle of parsimony, temporal patterns with small Q are examined

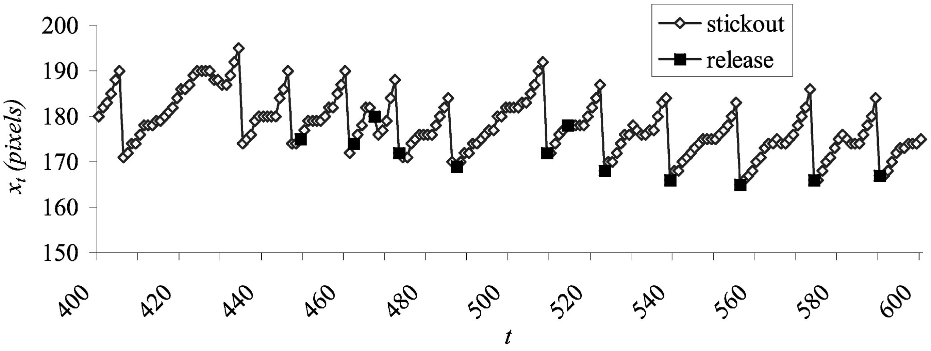

Fig. 9. Stickout and release time series.

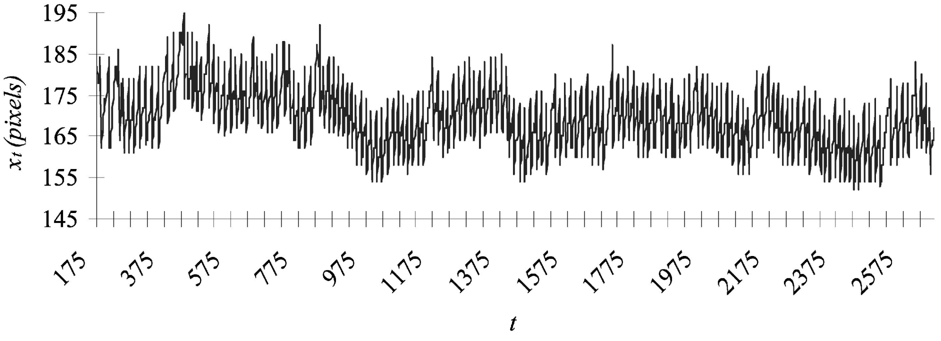

Fig. 10. Stickout time series (training).

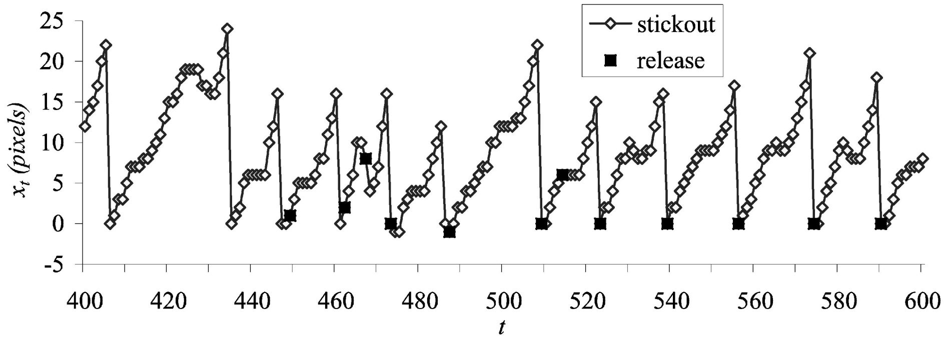

Fig. 11. Recalbrated stickout and release time series (training).

first. For this example, $Q = 2$, which allows a graphical presentation of the phase space.

Step 3—Unfold the Training Time Series into the Phase Space.

For this example, 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, as illustrated in Fig. 3. The value of τ is set to one. The Manhattan or l_1 distance [29] is chosen as the metric for this phase space. Given two points $\mathbf{y}$ and $\mathbf{z}$ in $\mathbb{R}^Q$, the l_1 distance between the two points is

$$d(\mathbf{y}, \mathbf{z}) = \sum_{i=1}^{Q} |y_i - z_i|.$$

Step 4—Form Augmented Phase Space.

The next step is to form the augmented phase space by augmenting the phase space with the extra dimension $g(\cdot)$, as illustrated by Fig. 4. The vertical lines represent the dimension $g(\cdot)$ associated with the pairs of (x_{t-1}, x_t).

Step 5—Search for Optimal Temporal Pattern Cluster.

The next step is to find an optimal temporal pattern cluster. In Fig. 6, the temporal pattern cluster found by our GA is highlighted in the phase space. The "circle" P (recall the phase space distance is Manhattan) in Fig. 6 has its center at $\mathbf{p}$ with radius δ.

3.3.2 Testing Stages

Step 1—Unfold the Testing Time Series into the Phase Space.

The time series Y shown in Fig. 7 is unfolded into a two-dimensional phase space.

Step 2—Predict Events.

The last step in our method is to predict events by applying the discovered temporal pattern cluster to the testing phase space. For this example, Fig. 7 clearly illustrates the accuracy of the temporal pattern in predicting events. The pair of connected gray squares that match subsequences of time series observations before events is the temporal pattern. The black squares indicate predicted events.

4 Application—Welding Droplet Releases

4.1 Problem Statement

In this section, our method of temporal pattern identification is applied to the prediction of metal droplet release events in the welding process [30]. Simply put, the welding process joins two pieces of metal by forming a joint between them. As illustrated in Fig. 8, 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.

Predicting when a droplet of metal will release from a welder allows the quality of the metal joint to be monitored and controlled. The problem is to predict the releases using a stickout time series. Samples of these time series are presented in Fig. 9.

Sensors on the welding station generate 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 indicates the release of the metal droplets. The release time series indicates the events

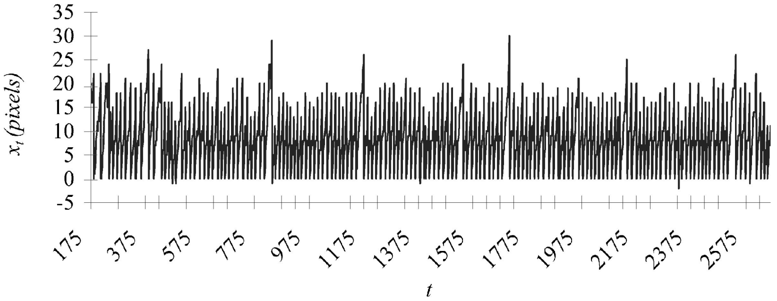

Fig. 12. Recalibrated stickout time series (training).

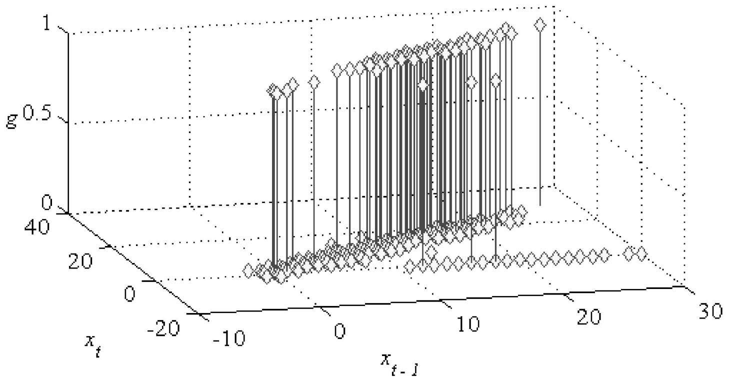

Fig. 13. Sickout and release augmented phase space (training).

with a one indicating an event and a zero indicating a nonevent.

4.2 Training Results

The training stickout time series X consists of the 2,492 equally sampled observations, at t = 175 through 2,666. Fig. 10 illustrates all observations, while Fig. 9 provides a detailed view of a sample of the time series.

The stickout time series is preprocessed to remove the large-scale artifact. Removing the trend transforms structurally different temporal patterns into similar temporal patterns. Without this transformation, testing accuracy is lower and the method predicts few of the droplet releases. This is done using a simple recalibration rule. When there is a 10-pixel drop between two consecutive observations, the second observation is recalibrated to zero. Figs. 11 and 12 illustrate that the trend in the stickout time series has been removed in the postprocessed time series. It is important to note that this preprocessing occurs online and incorporates no future data into current prediction calculations.

The events are captured in the release time series Y, as illustrated in Fig. 11. The release time series is defined as a binary sequence, where the ones indicate a release (event) and the zeros a nonrelease (nonevent).

Now that the training time series have been presented, the goal to predict the release of a droplet is restated in terms of the objective and event characterization functions. The event characterization function is $g(\mathbf{x}_t) = y_{t+1}$, which allows one-step prediction. We search for an optimal collection of temporal pattern clusters. Our method requires one objective function for the greedy search for the collection of temporal pattern clusters. The objective function for the collection of temporal pattern clusters is

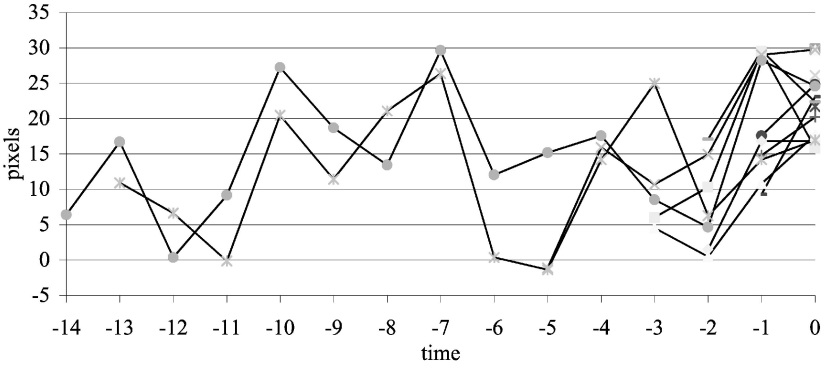

Fig. 14. Temporal patterns.

TABLE 2
Recalibrated Stickout and Release Results (Training)

	Actually an event	Actually a nonevent
Categorized as an event	*tp* = 101	*fp* = 41
Categorized as a nonevent	*fn* = 53	*tn* = 2296

$$f(\mathcal{C}) = \frac{tp + tn}{tp + tn + fp + fn}.$$

An objective function is also required for the identification of individual temporal pattern clusters. This is the fitness function used by our genetic algorithm. It is

$$f(P) = \frac{tp}{tp + fp}.$$

Two parameter sets are used for the genetic algorithm. For both sets, the initial population size multiplier s is 10, the population size is 30, the elite count is one, the gene length l is eight, maximum phenotype value $p_{\max}$ is 30, minimum phenotype value $p_{\min}$ is -2, and the tournament size is two. The first parameter set has a mutation rate of 0 percent and convergence criterion of 0.65. The second set has mutation rate of 0.05 percent and a convergence criterion of 0.5.

The range of phase space dimensions Q is $[1, 20]$ and τ is set to 1. Fig. 13 presents an illustrative two-dimensional augmented phase space. The events are not separable from the nonevents using a two-dimensional phase space.

The new method was run twice on the training time series. The first run uses the first GA parameter set and results in six temporal pattern clusters that vary in dimension from one to 15. The second run uses the second GA parameter set and results in eight temporal pattern clusters that vary in dimension from one to six. Recall that number and dimension of the temporal pattern clusters is not predetermined, but rather a result of the method. The temporal patterns are illustrated in Fig. 14.

Although Fig. 14 shows only the temporal patterns and not the temporal pattern clusters, it provides insight into the patterns in the stickout time series that precede the sharp drop in stickout and consequent release. We see that the peaks before the droplet release are of varying heights. We also can see that similar peaks may be reached without a droplet release as is illustrated by the longer temporal patterns. This illustrates the complicated nature of predicting welding droplet releases as was first observed in Fig. 13.

The result from the union of the two collections of temporal pattern clusters is shown in Table 2. The table contains 2,491 data points, which is one less than the number of time series observations in the training time series. This is because no data is used before the first observation to predict it; hence, the first observation cannot be categorized.

The statistics for accuracy indicate that 96.23 percent of the observations in the training time series are correctly predicted as events or nonevents. The positive accuracy indicates that 71.13 percent of the release observations predicted as events are events.

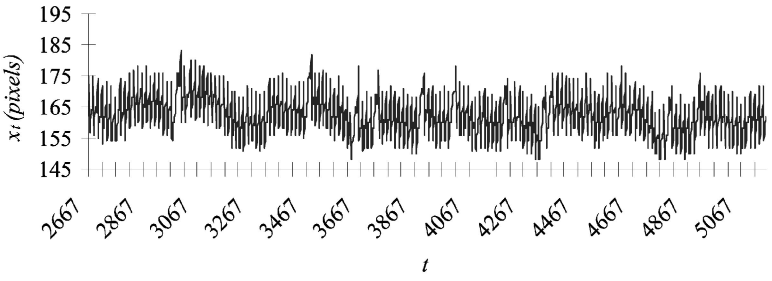

Fig. 15. Stickout time series (testing).

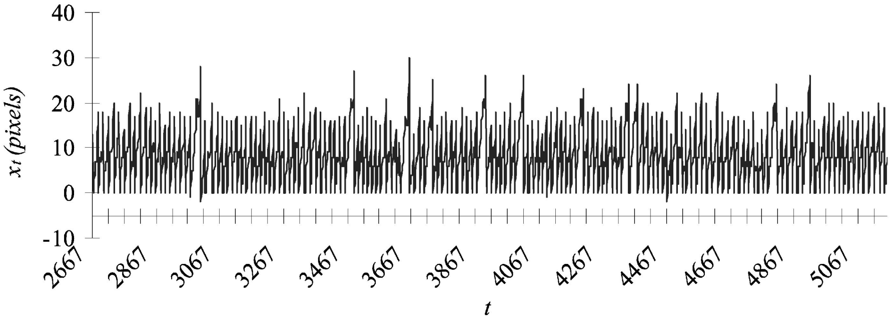

Fig. 16. Recalibrated stickout time series (testing).

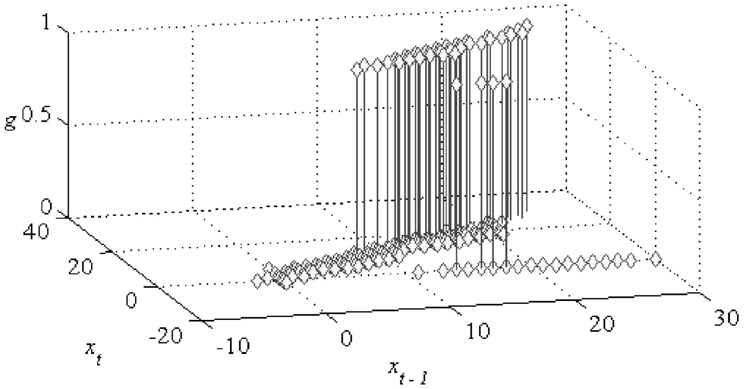

Fig. 17. Recalibrated stickout and release augmented phase space (testing).

4.3 Testing Results

The testing stickout time series, which is shown in Fig. 15, consists of the 2,493 equally sampled observations, at $t = 2,667$ through $5,159$. The recalibrated stickout time series is shown in Fig. 16.

The illustrative two-dimensional augmented phase space for the testing time series is seen in Fig. 17.

The results of applying the collection of temporal pattern clusters discovered during the training phase to the testing time series are seen in Table 3.

The prediction accuracy is 96.43 percent, and the positive accuracy is 73.53 percent. These results are better than those found in the training phase. They also indicate that overtraining has not occurred.

4.4 Analysis and Comparison of Results with TDNN and C4.5

In this section, we compare our method with a time delay neural network (TDNN) [21] and the C4.5 decision tree algorithm [20]. We also analyze the results for our method along with the results from the TDNN and C4.5. Our method's results are much better than both the TDNN and the C4.5 algorithm, especially for the test results for positive prediction, e.g., correctly predicting droplet releases.

Both algorithms were provided with the same data set used to train our method, that is the previous 15 values of the stickout time series to predict the droplet release in the next time step. Recall that our method as part of its optimization process discovered a collection of temporal pattern clusters. The maximum dimension of any of the temporal pattern clusters contained in the collection was 15. This indicates the number of previous values used in the prediction of an event.

The TDNN has four layers with 15 neurons in the input layer, 20 neurons in the first hidden layer, 100 neurons in the second hidden layer, and one neuron in the output layer. Sigmoid style activations functions are used in the first three layers, and a threshold style is used in the output layer. The TDNN was trained for 800 epochs.

The C4.5 tree has 53 nodes and 27 leaves. Table 4 shows the results for the TDNN, C4.5, and the new method.

TABLE 3
Recalibrated Stickout and Release Results (Testing)

	Actually an event	Actually a nonevent
Categorized as an event	*tp* = 100	*fp* = 36
Categorized as a nonevent	*fn* = 53	*tn* = 2303

TABLE 4
Summary of Prediction Results for Various Methods

	Training		Testing	
Method	Prediction Accuracy	Positive Prediction Accuracy	Prediction Accuracy	Positive Prediction Accuracy
TDNN	85.00%	3.11%	77.71%	1.01%
C4.5	97.98%	89.23%	89.44%	8.46%
New Method	96.23%	71.13%	96.43%	73.53%

In comparing the three methods, we see that in the training phase the C4.5 method performed best, whereas the TDNN performed quite poorly. Its accuracy in predicting events as events is less than 5 percent. The testing results show the superiority of the new method. The testing results are essentially equivalent to the training results for the new method, indicating that the discovered temporal pattern clusters are not artifacts of the training process, but rather valid event predictors. On the other hand, both the C4.5 and TDNN methods show substantially poorer results in the testing phase. This is especially true when examining the positive accuracy. The C4.5 method had a training positive accuracy of almost 90 percent, but a testing positive accuracy of less than 10 percent.

These results can be better understood by examining the augmented phase spaces illustrated in Figs. 13 and 17. Although showing only two time lags, it is obvious that there is no simple, consistent pattern that allows for the accurate prediction of events and nonevents. This is further illustrated by the presentation of the discovered temporal patterns in Fig. 14.

Using the stickout time series generated from a digital camera on a welding station, the problem of predicting when a droplet of metal will release from the welder is solved with a high degree of accuracy: 96.43 percent total prediction accuracy and 73.53 percent positive prediction accuracy. These results demonstrated that the new method could be used in an automated monitoring system of the welding seam, thereby improving the quality of the weld.

The results can be applied to improving welds in two ways. The first is by being able to tell how many droplets have been laid down on the welding seam. The second mechanism would be by using the welding droplet release prediction as an input into a control system.

5 Conclusions

It can be seen that our novel method works very well in characterizing and predicting complex time series events, especially in comparison to other well-known methods. The paper has presented the new framework including the key concepts of event characterization function, temporal pattern clusters, time-delay embedding, augmented phase space, and objective function. This framework has also been successfully applied to the financial domain. The application to the financial domain yielded greater than market investment returns, which were statistically significant [17], [24], [31].

Future work will involve improving the accuracy by using a system identification approach. This approach will identify rather than predict droplet releases. Alternative eventness functions also can be employed to improve accuracy. One example is an event function that characterizes all events one to five time steps ahead instead of in just one time step ahead. Additional research will be conducted into the performance improvements and scalability of the method, as well as the condition in which our method will overtrain.

References

[1] S.M. Pandit and S.-M. Wu, *Time Series and System Analysis, with Applications.* New York: Wiley, 1983.

[2] S.M. Weiss and N. Indurkhya, *Predictive Data Mining: A Practical Guide.* San Francisco: Morgan Kaufmann, 1998.

[3] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthursamy, *Advances in Knowledge Discovery and Data Mining.* Menlo Park, Calif.: AAAI Press, 1996.

[4] P. Cabena and International Business Machines Corporation, *Discovering Data Mining: From Concept to Implementation.* Upper Saddle River, NJ: Prentice Hall, 1998.

[5] D.J. Berndt and J. Clifford, "Finding Patterns Dynamic Programming Approach," *Advances in Knowledge Discovery and Data Mining,* U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthursamy, eds., pp. 229-248, Menlo Park, Calif.: AAAI Press, 1996.

[6] E. Keogh and P. Smyth, "A Probabilistic Approach to Fast Pattern Matching in Time Series Databases," *Proc. Third Int'l Conf. Knowledge Discovery and Data Mining,* 1997.

[7] E. Keogh, "A Fast and Robust Method for Pattern Matching in Time Series Databases," *Proc. Ninth Int'l Conf. Tools with Artificial Intelligence (TAI '97),* 1997.

[8] E.J. Keogh and M.J. Pazzani, "An Enhanced Representation of Time Series Which Allows Fast and Accurate Classification, Clustering and Relevance Feedback," *Proc. AAAI Workshop Predicting the Future: AI Approaches to Time-Series Analysis*, 1998.

[9] M.T. Rosenstein and P.R. Cohen, "Continuous Categories For a Mobile Robot," *Proc. 16th National Conf. Artificial Intelligence,* 1999.

[10] V. Guralnik, D. Wijesekera, and J. Srivastava, "Pattern Directed Mining of Sequence Data," *Proc. Int'l Conf. Knowledge Discovery and Data Mining,* pp. 51-57, 1998.

[11] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast Subsequence Matching in Time-Series Databases," *Proc. Sigmod Record (ACM Special Interest Group on Management of Data Conf.),* pp. 419-429, 1994.

[12] B.-K. Yi, H.V. Jagadish, and C. Faloutsos, "Efficient Retrieval of Similar Time Sequences Under Time Warping," *Proc. 14th Int'l Conf. Data Eng.,* pp. 201-208, 1998.

[13] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient Similarity Search In Sequence Databases," *Proc. Foundations of Data Organization and Algorithms Conf.,* pp. 69-84, 1993.

[14] C. Faloutsos, H.V. Jagadish, A.O. Mendelzon, and T. Milo, "A Signature Technique for Similarity-Based Queries," *Proc. Compression and Complexity of SEQUENCES Conf.,* pp. 2-20, 1997.

[15] H.D.I. Abarbanel, *Analysis of Observed Chaotic Data.* New York: Springer, 1996.

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

[17] R.J. Povinelli and X. Feng, "Temporal Pattern Identification of Time Series Data using Pattern Wavelets and Genetic Algorithms," *Proc. Artificial Neural Networks in Eng. Conf.,* pp. 691-696, 1998.

[18] H.D.I. Abarbanel, R. Brown, J.J. Sidorowich, and L.S. Tsimring, "The Analysis of Observed Chaotic Data in Physical Systems," *Rev. Modern Physics,* vol. 65, pp. 1331-1392, 1993.

[19] J. Iwanski and E. Bradley, "Recurrence Plot Analysis: To Embed or not to Embed?," Chaos, vol. 8, pp. 861-871, 1998.

[20] J.R. Quinlan, *C4.5: Programs for Machine Learning.* San Mateo, Calif.: Morgan Kaufmann Publishers, 1993.

[21] C.T. Lin and C.S.G. Lee, *Neural Fuzzy Systems—A Neuro-Fuzzy Synergism to Intelligent Systems.* Upper Saddle River, N.J.: Prentice-Hall, 1996.

[22] F. Takens, "Detecting Strange Attractors in Turbulence," *Proc. Dynamical Systems and Turbulence*, pp. 366-381, 1980.

[23] E. Bradley, "Analysis of Time Series," *An Introduction to Intelligent Data Analysis,* M. Berthold and D. Hand, eds., pp. 167-194, New York: Springer, 1999.

[24] R.J. Povinelli and X. Feng, "Data Mining of Multiple Nonstationary Time Series," Proc. Artificial Neural Networks in Eng., pp. 511-516, 1999.

[25] D.J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures.* Boca Raton, Fla.: CRC Press, 1997.

[26] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* Reading, Mass.: Addison-Wesley, 1989.

[27] R.J. Povinelli and X. Feng, "Improving Genetic Algorithms Performance By Hashing Fitness Values," *Proc. Artificial Neural Networks in Eng. Conf.,* pp. 399-404, 1999.

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

[29] D.G. Luenberger, *Optimization by Vector Space Methods.* New York: Wiley, 1969.

[30] R.J. Povinelli, "Time Series Data Mining: Identifying Temporal Patterns for Characterization and Prediction of Time Series Events," PhD dissertation, Marquette Univ., 1999.

[31] R.J. Povinelli, "Identifying Temporal Patterns for Characterization and Prediction of Financial Time Series Events," *Proc. Temporal, Spatial and Spatio-Temporal Data Mining: First Int'l Workshop; revised papers, (TSDM '00),* pp. 46-61, 2000.

Richard J. Povinelli (S'85, M'87, SM'01) received the BS degree in electrical engineering and the BA degree in psychology from the University of Illinois, Champaign-Urbana, in 1987, the MS degree in computer and systems engineering from Rensselaer Polytechnic Institute in Troy, New York, in 1989, and the PhD degree in electrical and computer engineering from Marquette University in 1999. From 1987 to 1990, he was a software engineer with General Electric Corporate Research and Development. From 1990 to 1994, he was with GE Medical Systems, where he served as a program manager and then as a global project leader. From 1995 to 1998, he held the positions of lecturer and adjunct assistant professor with the Department of Electrical and Computer Engineering, Marquette University, Milwaukee, Wisconsin, where, since 1999, he has been an assistant professor. His research interests include data mining of time series, chaos and dynamical systems, computational intelligence, and financial engineering. He is a member of the ACM, American Society of Engineering Education, Tau Beta Pi, Phi Beta Kappa, Sigma Xi, Eta Kappa Nu, Upsilon Pi Epsilon, and Golden Key. He is a senior member of the IEEE and a member of the IEEE Computer Society.

Xin Feng (S'85, M'87, SM'01) received the college diploma in computer engineering in 1976, the Master of Engineering degree in automatic control in 1981, both from Beijing Institute of Technology, China. He received the MS degree in 1984 and the DSc degree in 1987, both in systems and computer science from Washington University in St. Louis, Missouri. He joined the faculty of the Department of Electrical and Computer Engineering, Marquette University in Milwaukee, Wisconsin, as assistant professor in 1987. He is currently an associate professor in the department. His area of research interests include intelligent systems, data mining, artificial neural networks, fuzzy systems, and optimization. Xin Feng is a senior member of the IEEE. He is also a member of the IEEE Computer Society, and a member of the ACM.