Online Novelty Detection on Temporal Sequences

Junshui Ma
Aureon Biosciences Corp
Yonkers, NY 10701 USA
614-288-3593, +1
junshuima@yahoo.com

Simon Perkins
NIS-2, Los Alamos National Lab MS D436, NIS-2, LANL
Los Alamos, NM 87544 USA
505-667-9558, +1
s.perkins@lanl.gov

Abstract:

In this paper, we present a new framework for online novelty detection on temporal sequences. This framework includes a mechanism for associating each detection result with a confidence value. Based on this framework, we develop a concrete online detection algorithm, by modeling the temporal sequence using an online support vector regression algorithm. Experiments on both synthetic and real world data are performed to demonstrate the promising performance of our proposed detection algorithm.

Categories and Subject Descriptors

I.5.2 **[Pattern Recognition]**: Design Methodology

General Terms

Algorithms

Keywords:

Novelty detection, Anomaly Detection, Online algorithm, Support vector regression.

1. INTRODUCTION

Novelty detection, or anomaly detection, refers to the automatic identification of unforeseen or abnormal phenomena embedded in a large amount of normal data. [4, 8, 17] This paper targets the novelty detecting in temporal sequences [2, 4, 5, 8], which has many immediate applications. For example, in a safety-critical environment, it is helpful to have an automatic supervising system, which can screen the time series generated by monitoring sensors, and report any abnormal observations. Another promising application is to help scientists in different areas by liberating them from exhaustive examination of data, and drawing their attention only to unusual and "interesting" phenomena.

Novelty detection is a challenging topic, mainly because it is hard to obtain sufficient knowledge and an accurate representative of "novelty" given a problem [17], which rules out the use of many supervised techniques. Despite its challenge, in the past over ten years it has been a topic acquiring increasing attention, and quite a few techniques have been proposed and investigated. These techniques were experimentally proven to be effective in some cases, while they can fail in other cases. For example, some methods were designed based on the assumption of possessing precise models of the underlying problem [9], or knowing the novelty conditions [1, 6, 13], which are generally not true in real world. In some other studies [3, 7, 10, 14], novelty detection was simply interpreted as outlier detection. However, this simplification produces methods that cannot discover novel patterns formed by several continuous instances. In particular, the novelty detection method proposed in [14] and [3] is based on a technique called one-class support vector machine, whose formulation forces it to "identify" some novel individuals no matter how normal the whole data set is. A wavelet-based signal trend shift detection method is proposed in [15]. Nevertheless, this method cannot detect short novel patterns embedded in normal signals. An interesting idea for novelty detection, inspired by the negative-selection mechanism in the immune system, was proposed in [4]. However, this method can fail when the negative set goes to null with the increasing diversity of the normal set. Another method for target time-series novelty detection, called TARZAN [8], is based on converting the time series into a symbolic string. However, the procedure for discretizing and symbolizing real values in time series, as well as the various choices of string-representation parameters, can cause the loss of meaningful patterns in the original time series.

Furthermore, the ability to conduct online novelty detection is especially desirable for temporal sequences. However, few of the existing algorithms explicitly address this issue. The most relevant result is an incremental algorithm designed to detect normal events, instead of novel events. [5]

In this paper, we propose a direction to detect novelty in temporal sequences. As with other detection algorithms, it is impossible for our proposed direction to succeed in all scenarios. However, it can at least provide an alternative and complementary solution to some problems in which other available techniques may fail. The remainder of this paper is organized as follows. A general online novelty detection framework is proposed in Section 2. Based on the framework, a concrete online detection algorithm is proposed. Because the new algorithm utilizes a technique called support vector regression (SVR), Section 3 is devoted to a brief introduction to SVR. The new algorithm, as well as issues involved in its implementation, is presented in Section 4. Two variants of the original detection algorithms are proposed in Section 5 to adapt the original algorithm to different application scenarios. Experiments are proposed in Section 6.

The main contributions of this paper are (a) It proposes an online novelty detection framework for temporal sequences. This online framework is capable of associating a confidence level with each detection result. (b) It proposes a concrete online novelty detection algorithm based on the framework, and describes experiments to test the new algorithm.

2. A FRAMEWORK FOR ONLINE NOVELTY DETECTION WITH CONFIDENCE

We first reflect on the concept of *novelty*, as well as that of *novelty detection*, philosophically. First of all, *novelty* is always a relative concept with regard to our current knowledge. Therefore, *novelty* should be defined in the context of a representation of our current knowledge. Such representation can be a database, a knowledge base, or a model. In our framework, we prefer representing our knowledge with a model, simply because of its mathematical neatness. Moreover, in online applications, it is desirable that the representation of our knowledge can also be updated with the acquisition of new data. Second, there is generally no clear-cut line between novel events and normal events in real world application. Therefore, it is attractive to associate with each novel event a value to characterize how confident we are about our judgment. Finally, in a real world environment with noise the novel events in a temporal sequence are generally associated with segments, instead of individual time points. These understandings lay the foundation of a formal formulation of an online novel detection framework, which is defined in the remaining part of this section.

In this paper, a temporal sequences is represented as $\mathbf{X}(t)$, where $t = 1 \cdots N$. They can be time series, video sequences, or some other objects that can be indexed by time *t*. $\mathbf{X}(t)$ is a stochastic process, and $\mathbf{x}(t)$ is employed to represent one of its realizations.

As we mentioned previously, we utilize a model $\mathbf{M}_{\mathbf{x}}(t_0)$ to represent our knowledge about the underlying temporal sequence up to t_0. This model can be a physics-based model provided by domain experts, or a model constructed from available data $\mathbf{x}(t)$, where $t = 1 \cdots t_0$.

Surely, it is impossible for us to utilize a model with finite number of parameters to completely represent the information embedded in a random temporal sequence. Also, each model may only be good at representing certain types of information, because different models have different sensitivity to different type of information. For example, some models are more sensitive to extreme values, while some other models are more sensitive to unusual patterns. However, this selective property of model-based knowledge representation, when applied to novelty detection, can become an advantage. In a particular application, we may only be interested in some types of novelties among all of the different possible novelties. Thus, utilizing different models provides us with the opportunities of selecting different types of interested novelties.

Definition 1 (Matching Value and Matching Function)

The matching function, denoted as $F(\mathbf{M}_{\mathbf{x}}(t_0 - 1), \mathbf{x}(t_0))$, *is a function that can quantify how well the model* $\mathbf{M}_{\mathbf{x}}(t_0)$ *matches the temporal sequence. The match value* $V(t_0)$, *where* $V(t_0) \in \mathbf{R}$, *is defined as* $V(t_0) = F(\mathbf{M}_{\mathbf{x}}(t_0 - 1), \mathbf{x}(t_0))$.

In other words, the matching value $V(t_0)$ is employed to measure how well our knowledge about the time series at $t_0 - 1$, represented by $\mathbf{M}_{\mathbf{x}}(t_0 - 1)$, can describe the instance $\mathbf{x}(t_0)$.

Definition 2 (Occurrence)

Denoted by $O(t_0)$, *occurrence at* t_0 *is defined as*

$$O(t_0) \equiv I\{V(t_0) \notin (-\varepsilon(t_0), \varepsilon(t_0))\} \tag{1}$$

where $I\{\bullet\}$ *is the indicator function, and* $2\varepsilon(t_0) > 0$ *is a predefined tolerance width.*

$\varepsilon(t_0)$ can be determined based on the noise level, as well as the precision requirement of the underlying problem. Note that $O(t_0)$ is a random variable.

Definition 3 (Surprise)

A surprise is observed if $O(t_0) = 1$.

That is, a surprise happens when a new instance $\mathbf{x}(t_0)$ falls outside of the tolerance range $(-\varepsilon(t_0), \varepsilon(t_0))$.

Definition 4 (Event and Event Duration)

Denoted by $E_n(t_0)$, *an event at time* t_0 *is defined as*

$$E_n(t_0) = \left[O(t_0) \quad O(t_0 + 1) \quad \cdots \quad O(t_0 + n - 1) \right]^T \tag{2a}$$

where n is called the event duration. The 1-norm of $E_n(t_0)$ *is denoted as* $|E_n(t_0)|$. *That is*

$$|E_n(t_0)| = \sum_{i=0}^{n-1} O(t_0 + i) \tag{2b}$$

$|E_n(t_0)|$ *is the number of surprises happened in the event* $E_n(t_0)$.

Event duration *n* is a predetermined algorithmic parameter. We omit the absolute operation on $O(t_0 + i)$ in (2b) because $O(t_0 + i)$ is nonnegative according to (1). $E_n(t_0)$ is a random binary vector with at most 2^n different realizations, while $|E_n(t_0)|$ is a random variable with at most n+1 different realizations. A realization of $E_n(t_0)$ (or $|E_n(t_0)|$) is denoted as $e_n(t_0)$ (or $|e_n(t_0)|$). The discrete density function of $|E_n(t_0)|$ is represented as $p_{E_n}(|e_n(t_0)|)$, where $|e_n(t_0)| = 0 \cdots n$. The formulation of $p_{E_n}(|e_n(t_0)|)$ can be determined by the occurrences $O(t_0 + i), i = 0 \cdots n-1$. For example, if the occurrences in $\{O(t_0 + i), i = 0 \cdots n-1\}$ are identical independent Bernoulli variables, $E_n(t_0)$ becomes a binomial random variable. $p_{E_n}(|e_n(t_0)|)$ will be different if the occurrences in $\{O(t_0 + i), i = 0 \cdots n-1\}$ are interdependent, such as following a Markov chain distribution.

Definition 5 (Novel Event with Confidence)

Given a confidence level $c(t_0)$, *where* $c(t_0) \in (0,1)$, *Event* $e_n(t_0)$ *is defined as a novel event with confidence* $c(t_0)$, *if* $e_n(t_0)$ *satisfies*

(a) $|e_n(t_0)| > \max(h, \mathbf{E}\{|E_n(t_0)|\})$, *where* $\mathbf{E}\{\bullet\}$ *is the mean, h is a fixed lower bound of* $|E_n(t_0)|$ *with* $h \in \mathbf{N}$, *and* (3a)

(b) $p_{E_n}(|e_n(t_0)|) < 1 - c(t_0)$. (3b)

The condition (3a) makes sure that at least certain number of surprises happen in the event $e_n(t_0)$, while the condition (3b) ensures that the probability for the number of surprises to happen in the event $e_n(t_0)$ is small enough to satisfy our confidence level $c(t_0)$. The h is an algorithmic parameter to define the lower bound of the number of surprises.

Four items in this framework need to be instantiated before it becomes a concrete algorithm. (a) The model $\mathbf{M}_\mathbf{x}(t_0)$ to represent the temporal sequence $\mathbf{X}(t)$; (b) The matching function $F(\mathbf{M}_\mathbf{x}(t_0-1),\mathbf{x}(t_0))$ to quantify the disagreement between the model output and the temporal sequence $\mathbf{X}(t)$ observation at t_0; (c) The tolerance width $2\varepsilon(t_0)$ at t_0; (d) The discrete density function of $|E_n(t_0)|$, $p_{E_n}(|e_n(t_0)|)$, where $m=0\cdots n$. In Section 4, a concrete algorithm is derived by instantiating all these four items in the framework.

3. A BRIEF INTRODUCTION TO SUPPORT VECTOR REGRESSION

Because the algorithm proposed in Section 4 is extensively based on a regression technique, called Support Vector Regression (SVR), we briefly introduce the basic concepts of SVR in this section. A more detailed presentation of SVR can be found in [16].

Given a training set $T=\{(\mathbf{x}_i,y_i), i=1...l\}$, where $\mathbf{x}_i\in\mathbf{R}^D$, and $y_i\in\mathbf{R}$, we construct a linear regression function with regard to $\mathbf{W}$ and $\Phi(\mathbf{x})$

$$f(\mathbf{x})=\mathbf{W}^T\Phi(\mathbf{x})+b \tag{4}$$

where $\mathbf{W}$ and $\Phi(\mathbf{x})$ are vectors in a huge dimensional feature space $\boldsymbol{F}$. Meanwhile, $\Phi(\mathbf{x})$ can also be considered as a mapping function, which maps $\mathbf{x}\in\mathbf{R}^D$ to a vector in $\boldsymbol{F}$. The $\mathbf{W}$ and b in (4) are obtained by solving an optimization problem:

$$\begin{aligned}\min_{\mathbf{W},b}\quad & P=\frac{1}{2}\mathbf{W}^T\mathbf{W}+C\sum_{i=1}^{l}(\xi_i+\xi_i^*)\\ s.t.\quad & y_i-(\mathbf{W}^T\Phi(\mathbf{x})+b)\le\varepsilon+\xi_i\\ & (\mathbf{W}^T\Phi(\mathbf{x})+b)-y_i\le\varepsilon+\xi_i^*\\ & \xi_i,\xi_i^*\ge 0,\quad i=1\cdots l\end{aligned} \tag{5}$$

The optimization criterion penalizes data points whose y-values differ from $f(\mathbf{x})$ by more than ε. The slack variables ξ and ξ^* represent the size of this excess deviation for positive and negative deviations respectively. Introducing Lagrange multipliers α and α^*, we get:

$$\begin{aligned}\min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*}\quad & D=\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}Q_{ij}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)\\ & +\varepsilon\sum_{i=1}^{l}(\alpha_i+\alpha_i^*)-\sum_{i=1}^{l}y_i(\alpha_i-\alpha_i^*)\\ s.t.\quad & 0\le\alpha_i,\alpha_i^*\le C\quad i=1\cdots l,\quad \sum_{i=1}^{l}(\alpha_i-\alpha_i^*)=0\end{aligned} \tag{6}$$

where $Q_{ij}=\Phi(\mathbf{x}_i)^T\Phi(\mathbf{x}_j)=K(\mathbf{x}_i,\mathbf{x}_j)$. Here $K(\mathbf{x}_i,\mathbf{x}_j)$ is a kernel function [16]. Given the solution of (6), the regression function (4) can be written:

$$f(\mathbf{x})=\sum_{i=1}^{l}\theta_i K(\mathbf{x}_i,\mathbf{x})+b \tag{7}$$

where the coefficient $\theta_i=\alpha_i-\alpha_i^*$. (7) is a nonlinear function with regard to $\mathbf{x}\in\mathbf{R}^D$ if the kernel function $K(\mathbf{x}_i,\mathbf{x}_j)$ is chosen as a nonlinear function. Normally, only a small fraction of the coefficients θ_i in (7) are finally nonzero. Those samples $\mathbf{x}_i$ with nonzero θ_i are called *support vectors* of the regression function, because it is these critical samples in the training set T that solely determine the formulation of (7).

4. AN SVR-BASED ONLINE NOVELTY DETECTION ALGORITHM

In this section, we instantiate the four items in the framework listed in Section 2 to devise a novelty detection algorithm.

First, SVR is employed to model a temporal sequence $\mathbf{X}(t)$. SVR has quite a few attractive features. For example, (a) the regression function $f(\mathbf{x})$ obtained by SVR can approximate any nonlinear relationship between the input vector $\mathbf{x}$ and its associated target value y if a proper kernel function $K(\mathbf{x}_i,\mathbf{x}_j)$ is chosen, (b) SVR has good generalization properties, and (c) SVR can handle high dimensional data efficiently.

More specifically, given a realization of a temporal sequence, or $\mathbf{x}(t)$, where $t=1\cdots t_0$, we can construct a set of training samples $T_D(t_0)$ from $\mathbf{x}(t)$

$$T_D(t_0)=\{(\mathbf{x}_D^t,\mathbf{y}^t),\quad t=D\cdots t_0-1\}, \tag{8}$$

where $\mathbf{x}_D^t=[\mathbf{x}(t-D+1)\ \cdots\ \mathbf{x}(t)]^T$, $\mathbf{y}^t=\mathbf{x}(t+1)$, and D is called the *embedding dimension* of the training set $T_D(t_0)$. According to Section 3, from the training set $T_D(t_0)$ SVR training algorithm can construct a regression function $f(\mathbf{x}_D^{t_0})$

$$\hat{\mathbf{y}}^{t_0}=\hat{\mathbf{x}}(t_0+1)=f(\mathbf{x}_D^{t_0})=\sum_{t=1}^{t_0-1}\theta_i K(\mathbf{x}_D^t,\mathbf{x}_D^{t_0})+b \tag{9}$$

Naturally, the model $\mathbf{M}_\mathbf{x}(t_0)$ representing the temporal sequence $\mathbf{X}(t)$ can be defined as

$$\mathbf{M}_\mathbf{x}(t_0)=f(\mathbf{x}_D^{t_0})=\sum_{i=E}^{t_0-1}\theta_i K(\mathbf{x}_D^t,\mathbf{x}_D^{t_0})+b \tag{10}$$

Equation (10) suggests that model $\mathbf{M}_\mathbf{x}(t_0)$ essentially incorporates all the points of the temporal sequence $\mathbf{x}(t)$, where $t=1\cdots t_0$, and thus is a good candidate for representing our acquired knowledge of $\mathbf{X}(t)$.

Second, the matching function and the matching value are defined as

$$\begin{aligned}V(t_0)&=F(\mathbf{M}_\mathbf{x}(t_0-1),\mathbf{x}(t_0))\\ &=\mathbf{x}(t_0)-\mathbf{M}_\mathbf{x}(t_0-1)=\mathbf{x}(t_0)-\hat{\mathbf{x}}(t_0)\end{aligned} \tag{11}$$

Equation (11) indeed suggests that the matching value $V(t_0)$ is the residual of the regression function (9) at t_0.

Third, we need to define the tolerance width $2\varepsilon(t_0)$ at t_0. Fortunately, SVR formulation generally adopts an ε-insensitive loss function [16], which possesses exactly the same flavor as our tolerant range defined in Definition 2. Thus, we just simply merge the concepts of two tolerant ranges together, and define the tolerant width in Definition 2 as 2ε for any t_0, where ε is the insensitive parameter in the ε-insensitive loss function of SVR. A direct consequence of this definition is that any sample $(\mathbf{x}_D^{t_0}, \mathbf{y}^{t_0})$ in $T_D(t_0)$ that turns to be a *surprise* to model $\mathbf{M}_\mathbf{x}(t_0)$ will be a support vector in the updated model $\mathbf{M}_\mathbf{x}(t_0+1)$ [16].

Finally, in order to make our algorithm theoretically tractable, we simply define $E_n(t_0)$ as a sequence of independent Bernoulli random variables with the same parameter, and the discrete density function of $|E_n(t_0)|$, $p_{E_n}(|e_n(t_0)|)$, can thus be formulated as [12]

$$p_{E_n}(|e_n(t_0)|) = \begin{cases} \dbinom{n}{|e_n(t_0)|} q(t_0)^{|e_n(t_0)|}(1-q(t_0))^{n-|e_n(t_0)|}, \\ \qquad where \quad |e_n(t_0)| = 0 \cdots n \\ 0, \quad otherwise \end{cases} \tag{12}$$

where $q(t_0)$ is the probability of an occurrence in the event $E_n(t_0)$ to be a surprise. The $q(t_0)$ can be approximately estimated as

$$\hat{q}(t_0) = \frac{N_{SV}(t_0)}{t_0 - D}, \tag{13}$$

where $N_{SV}(t_0)$ is the number of support vectors in model $\mathbf{M}_\mathbf{x}(t_0)$, and $t_0 - D$ is the number of training samples in $T_D(t_0)$ according to (8).

This definition is valid under the following two assumptions (a) The occurrences in an event are independent; (b) All occurrences in an event have approximately the same probability of being a surprise. Intuitively, the first assumption is reasonable if the regression function (9) can sufficiently capture the dependent relationship in a temporal sequence. This can be achieved by an adequate training stage to fully build the model $\mathbf{M}_\mathbf{x}(t_0)$ before it is utilized for detection. The second assumption is sensible if the event duration *n* is not too large. More discussion of the role that event duration *n* plays in the algorithm will be presented in Subsection 5.1. Failure of meeting either assumption deteriorates the accuracy of the confidence level associated with the detection output.

We now have a complete online novelty detection algorithm has been proposed based the framework in Section 2. This algorithm requires a set of algorithmic parameters, which include (a) Embedding dimension *D*; (b) Event duration *n*; (c) Tolerant width 2ε; (d) Kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$; (e) Confidence level c; and (f) Fixed lower bound of the number of surprises *h*. The algorithmic procedures can be summarized as follows:

1) At training stage,

i) When a new point in the temporal sequence $\mathbf{x}(t_0)$ becomes available, the training set $T_D(t_0-1)$ will be updated to $T_D(t_0)$ following (8). Accordingly, the model $\mathbf{M}_\mathbf{x}(t_0-1)$ will also be updated to $\mathbf{M}_\mathbf{x}(t_0)$ based on the $T_D(t_0)$.

ii) If training stage does not finish, go to step i). Otherwise, move to the detection stage.

2) At detection stage,

i) When a new point in the temporal sequence $\mathbf{x}(t_0)$ becomes available, calculate the matching value $V(t_0)$ following (11).

ii) Based on $V(t_0)$ and the tolerant width 2ε, determine the value of occurrence $O(t_0)$ following (1).

iii) Based on the values of occurrence $O(t_0 - i)$, where $i = 0 \cdots n-1$, determine the value of event $E_n(t_0 - n + 1)$ following (2). If $E_n(t_0 - n + 1)$ meets the conditions in (3), a novel event, which starts at $t_0 - n + 1$ and ends at t_0, is detected with a confidence of *c*.

iv) The training set $T_D(t_0 - 1)$ is updated to $T_D(t_0)$ by considering the new point $\mathbf{x}(t_0)$ following (8). Accordingly, the model $\mathbf{M}_\mathbf{x}(t_0-1)$ is then updated to $\mathbf{M}_\mathbf{x}(t_0)$ based on $T_D(t_0)$.

v) If the detection stage does not finish, go to step i). Otherwise, quit.

Finally, it is worth mentioning that the SVR training algorithm introduced in Section 3 is a batch algorithm. That is, whenever a new sample is added into the training set, the existing regression function can only be updated by retraining the whole training set, which is not an efficient way to implement our detection algorithm. Fortunately, we have recently derived an incremental SVR training algorithm [11], which can efficiently update the regression function whenever a sample is added to or removed from the training set.

5. TWO VARIANTS OF THE NOVELTY DETECTION ALGORITHM

When we apply this algorithm to particular real world scenarios, it is sometimes necessary to fine-tune some parts of the algorithms to meet the special requirements of an application. In this section, we propose two variants of the original algorithm.

5.1 Robust Online Novelty Detection

The event duration *n* is a critical parameter in the detection algorithm. In some cases where the novel events are outstanding, the original algorithm is not sensitive to event duration *n*. This can be illustrated by experiments in Section 6. However, in the other cases, an improper choice of this parameter does lead to the deterioration of the algorithmic performance. Meanwhile, it is hard to know in advance what kind of novel events will be detected in a temporal sequence, which is indeed the nature of novelty detection.

Although, given a particular problem, how to select an optimal event duration *n* for the original detection algorithm is still an open topic, in some applications it is possible for us to know the range in which an optimal event duration *n* may fall in. Based on this assumption, a robust version of the original detection algorithm can be intuitively devised. In this variant, we evenly pick up *r* different event duration *n*'s from the available range, and apply each of them to the original detection algorithm, and thus obtain *r* detection outputs. The final robust detection output is obtained by a voting procedure among all the generated outputs. When implementing this idea, it is not necessary to literally repeat the original detection algorithm for each event duration *n*. Repeating merely the step iii) at the detection stage for each event duration *n* is adequate.

5.2 Fixed-Resource Online Novelty Detection

One problem with our original detection algorithm is the longer the prediction goes on, the bigger the training set $T_D(t_0)$ will become, and the more SVs will be involved in the SVR regression function (9). In some environments with limited memory and computational power, it is possible to stress out the system resources with the complexity of the SVR model (9) growing in this way. One way to deal with this problem is to impose a "forgetting" time *W*. When training set $T_D(t_0)$ grows to this maximum *W*, then the SVR model (9) will first be trained to remove the oldest sample before the next new sample is used to update the model. Accordingly, the $q(t_0)$ in (13) becomes $\hat{q}(t_0)=\frac{N_{SV}(t_0)}{W}$, where $N_{SV}(t_0)$ is still the number of support vectors in model $\mathbf{M}_{\mathbf{x}}(t_0)$.

This variant is intrinsically suitable for non-stationary temporal sequences, as it can be updated in real-time to fit the most recent behavior of a temporal sequence.

6. EXPERIMENTS

Experiments based on both synthetic and measured data are presented to demonstrate the performance of our algorithm. According to our knowledge, a comparable automatic online novelty detection algorithm supported by confidence is still not available in other literatures when we prepare this paper. Therefore, it is difficult for us to implement comparative experiments to justify our performance. Thus, we here do it by comparing our detection results with visual detection by humans.

6.1 Experiments Based on Synthetic Data

Three synthetic time series are generated from the following stochastic processes respectively.

$$\mathbf{X}_1(t)=\sin(\frac{40\pi}{N}t)+n(t) \tag{14a}$$

$$\mathbf{X}_2(t)=\sin(\frac{40\pi}{N}t)+n(t)+e_1(t) \tag{14b}$$

$$\mathbf{X}_3(t)=\sin(\frac{40\pi}{N}t)+n(t)+e_1(t)+e_2(t) \tag{14c}$$

where $t=1\cdots N$, $N=1200$, and $n(t)$ is an additive Gaussian noise with zero-mean and a SDT of 0.1. $e_1(t)$ and $e_2(t)$ are two novel events,

$$e_1(t)=\begin{cases} n_1(t), & t\in[600,620] \\ 0, & otheriwse \end{cases}$$

where $n_1(t)$ follows a normal distribution of $N(0,0.5)$.

$$e_2(t)=\begin{cases} 0.4*\sin(\frac{40\pi}{N}t), & t\in[820,870] \\ 0, & otheriwse \end{cases}$$

We use the first 400 points in the three time series to train our models, and conduct the novelty detection on the remaining 800 points. The algorithmic parameters are arbitrarily set as (a) Embedding dimension *D=8*; (b) Event duration *n=6*; (c) Tolerant width 2ε=0.2; (d) Kernel function $K(\mathbf{x}_i,\mathbf{x}_j)=\exp\{-|\mathbf{x}_i-\mathbf{x}_i|^2\}$; (e) Confidence level c = 95%; (f) Fixed lower bound of number of surprises *h =n/2*. The experimental results are demonstrated in Figure 1.

The blue curves in Figure 1 are the synthetic time series, and the peaks on the red lines indicate the positions and durations of the detected novel events. Because the first 400 points of each time series are taken out for training the support vector regression functions, no detection outputs are produced at that segment.

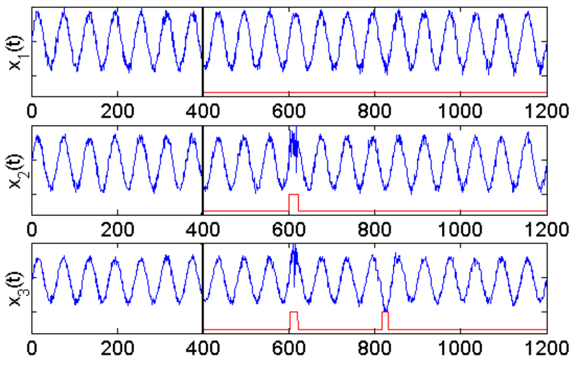

Figure 1. Experimental Results on Synthetic Time Series

The plots in Figure 1 show that, our detection algorithm successfully detects all the novel events in $\mathbf{x}_2(t)$ and $\mathbf{x}_3(t)$. Meanwhile, our algorithm properly figures out that no any part of the $\mathbf{x}_1(t)$ can be considered as a novel event. As suggested in Subsection 5.1, because the novel events in these synthetic time series are fairly distinguishable, the detection output of the original detection algorithm is not sensitive to the choice of event duration *n*. It can be shown that the same detection output can be produced when event duration *n* is set to 8 or 10.

6.2 An Experiment Based on Measured Data

The experiment is to apply the original detection algorithm to the famous Santa Fe Institute Competition data, which is 1000-point time series. We define the first 200 points as the train segment, and the remaining 800 points as the detection segment. The algorithmic parameters are set exactly the same as the experiments done in Subsection 6.1. The time series, along with the detection results, are plotted in Figure 2.

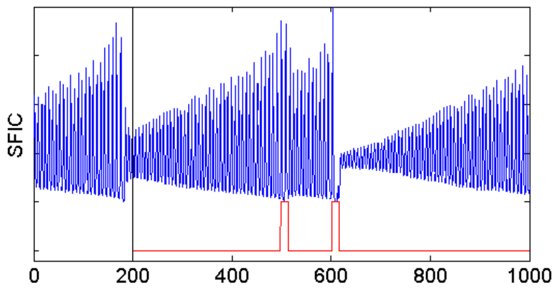

Figure 2. Experimental Results on Santa Fe Institute Time Series

In this experiment our algorithm claims that two novel events happen in the detection segment of the time series. It is easy to visually examine its validity. Also, similar to the experiments done in Subsection 6.1, this Santa Fe Institute Time Series is also insensitive to different choice of event duration *n*. The same result can be obtained if we set the event duration *n* to 8 or 10.

7. CONCLUSIONS

This paper proposes a new direction for online novelty detection on temporal sequences. Primitive experimental results demonstrate the promising performance of this algorithm.

Meanwhile, many topics brought up by this paper are still open. For example, we notice that some relationship exists among the algorithm parameters, such as event duration *n*, tolerant width 2ε, and confidence level *c*. However, we still cannot figure out a method to make use of this relationship to define a set of optimal algorithmic parameters. Also, intuitively, for some temporal sequences, Markov chain can be a better model than Binomial distribution to describe the relationship among the occurrences $O(t_0)$. Thus, how to implement this intuition is another direction worth future investigation. Surely, new detection algorithms can also be devised by employing different temporal sequence models, such as data clustering and one-class support vector machine.

8. ACKNOWLEDGEMENTS

Most of this work was supported by the NASA project NRA-00-01-AISR-088. The final part of it was supported by Aureon Bioscience Corp.

9. REFERENCES

[1] Bishop, C. M., Novelty Detection and Neural Network Validation, IEE Proceedings - Vision, Image and Signal Processing, vol. 141, no. 4, pp. 217-222, August, 1994.

[2] Brotherton, Tom, Tom Johnson, and George Chadderdon, Classification and Novelty Detection Using Linear Models and a Class Dependent-Elliptical Basis Function Neural Network, in Proceedings of the International Conference on Neural Networks, Anchorage, May 1998.

[3] Campbell, Colin, Kristin P. Bennett, A Linear Programming Approach to Novelty Detection, in Advances in Neural Information Processing Systems, vol 14, 2001.

[4] Dasgupta, Dipanker, and Stephanie Forrest, Novelty Detection in Time Series Data Using Ideas from Immunology, In Proceedings of the 5th International Conference on Intelligent Systems, Reno, Nevada, June 19-21, 1996.

[5] Guralnik, Valery, Jaideep Srivastava, Event Detection from Time Series Data. In Proceedings of the International Conference Knowledge Discovery and Data Mining, San Diego, California, 1999.

[6] Isermann, Rolf, Process Fault Detection Based on Modeling and Estimation Method - A Survey, Automatica, vol. 20, pp.387-404, 1984.

[7] Jagadish, H. V., N. Kouda, and S. Muthukrishnan, Mining deviates in a time series database, in Proceedings of 25th International Conference on Very Large Data Bases, pp 102-113, 1999.

[8] Keogh, E., S Lonardi, and W Chiu, Finding Surprising Patterns in a Time Series Database In Linear Time and Space, In the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 550-556, Edmonton, Alberta, Canada, July 23 - 26, 2002.

[9] Kozma, R., M. Kitamura, M. Sakuma, and Y. Yokoyama, Anomaly Detection by Neural Network Models and Statistical Time Series Analysis, in Proceedings of IEEE International Conference on Neural Networks, Orlando, Florida, June 27-29, 1994.

[10] Lauer, Martin, A Mixture Approach to Novelty Detection Using Training Data With Outliers, Lecture Notes in Computer Science, vol. 2167, pp. 300-310, 2001.

[11] Ma, Junshui, James Theiler, and Simon Perkins, "Accurate Online Support Vector Regression," to appear in Neural Computation, 2003.

[12] Mood, A. M., F. A. Graybill, and D. C. Boes, Introduction to The Thoery of Statistics, 3rd Edition, McGraw-Hill, Inc, 1974.

[13] Roberts, S., and L. Tarassenko. A Probabilistic Resource Allocating Network for Novelty Detection, Neural Computation, vol. 6, pp. 270-284, 1994.

[14] Schölkopf, B., R.C. Williamson, A.J. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. In Neural Information Processing Systems, 2000.

[15] Shahabi. C., X. Tian, and W. Zhao, Tsa-tree: A Wavelet-based Approach to Improve the Efficiency of Multi-level Surprise and Trend Queries. In Proceedings of 12th International Conference on Scientific and Statistical Database Management, 2000.

[16] Smola, A. J., and B. Scholkopf (1998). A Tutorial on Support Vector Regression, NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK.

[17] Ypma, Alexander, and Rober P. Duin, Novelty Detection Using Self-Organizing Maps, in Progress in Connectionist-Based Information Systems, pp 1322-1325, London: Springer, 1997.