RESEARCH ARTICLE

Outlier Detection of Air Temperature Series Data using Probabilistic Finite State Automata-Based Algorithm

JUN SHEN,[1,2] MINHUA YANG,[1] BIN ZOU,[1] NENG WAN,[3] AND YUFANG LIAO[2]

[1]*Key Laboratory of Metallogenic Prediction of Nonferrous Metals, Ministry of Education, School of Geosciences and Info-Physics, Central South University, 410083 Changsha, China;* [2]*Hunan Climate Central, Meteorological Administration of Hunan, 410000 Changsha, China; and* [3]*Department of Health Services Research and Administration, College of Public Health, University of Nebraska Medical Center, 68198 Omaha, Nebraska*

Received May 31, 2011; revised November 24, 2011; accepted December 20, 2011

This article proposes a probability finite state automata-based algorithm (PFSAA) for detecting outliers of air temperature series data caused by sensor errors. This algorithm first divides the training samples of air temperature series data into subclusters that will be further used to build finite state automata by splitting and combining techniques. Then, it creates a dynamic transition matrix of PFSA based on probability theories. Finally, the outliers of the remaining test samples are detected by PFSAA. The proposed algorithm is quantitatively validated by the reference data and a traditional backpropagation neural net model. © *2012 Wiley Periodicals, Inc. Complexity 17: 48–57, 2012*

Key Words: AWS; chaos; PFSAA; meteorological data; time series

1. INTRODUCTION

Timely and accurate weather information, especially that in bad weather conditions and irrigation periods, is critical for human life and activities. Therefore, it is of great importance to monitor and deal with meteorological data (e.g., air temperature) to provide accurate weather forecast. Although continuous air temperature information can be obtained through wireless sensor net (WSN) with increased automatic weather stations (AWSs) [1], the quality of AWS data is influenced by the abnormal disturbances (i.e., outlier) resulting from sensor's errors [2]. As a result, false records or systematic abruptions might be introduced, which have negative influence on the accuracy of weather forecasts.

As AWS is a relatively recent application of WSN in meteorological observation, little attention has been paid to the detection of the outlier caused by sensor's errors from the observed data. However, in the fields of navigation system monitoring [3], signal segmentation [4], and mechanical vibration monitoring [5], methods for outlier detection have already been extensively proposed during the past

Correspondence to: Bin Zou, School of Geosciences and Info-Physics, Central South University, No. 932 South Lvshan Road, 410083 Changsha, China (e-mail: B.zou2010csu@gmail.com)

 Vol. 17, No. 5
DOI 10.1002/cplx.21390
Published online 21 February 2012 in Wiley Online Library
(wileyonlinelibrary.com)

decade. For example, Ray [6] presented a novel concept of outlier detection in complex dynamical systems using symbolic dynamics. Davy et al. [7] provided a support-vector-based model to optimize the anomaly detection. Rajagopalan and Ray [8] provided a new symbolic time series analysis method to validate the wavelet efficiency based on phase-space partitioning.

Unfortunately, almost all the methods mentioned above assume that the processed signal is stationary, which is not always valid in the reality. Thus, an effective method for processing chaos series data (e.g., air temperature series data) with highly time-varying characteristics is needed. Probability finite state automata (PFSA), which is based on statistical theories and state transformation techniques, works in a similar way to the Markov process. The merit of PFSA is that it can not only accurately trace the hidden variation module of a series data by ignoring the plot-added disturbances but also focus on the transformation between adjacent states. In this study, we propose a PFSA-based algorithm (PFSAA) to differentiate abnormal disturbances in chaos series data, which are caused by sensor errors. This algorithm is based on the three assumptions shown below.

a. The dynamic system is nearly stationary at a short time scale (e.g., 20 minutes) rather than among short time scales.
b. The state number of an observed series data can be identified based on the similarity of the adjacent states, and
c. Continuous anomaly [9] is a minor part in a time series data, which means the air temperature series data can be simulated using a specific model.

2. DATA

AWS is a network of automatic data collectors. It has been developed rapidly in recent years to provide meteorologists and the public with basic weather information, including precipitation, wind speed, wind direction, air temperature, etc. The air temperature series data used in this study was collected by six wireless sensors (i.e., sensors A, B, C, D, E, and F) from an AWS in Yiyang City of Hunan Province, China. Data from three of the six sensors were used as the reference data because these sensors have been sheltered against the influences of factors such as precipitation and wind. Data from other sensors observed at the same location were recognized as the air temperature data with outliers. Table 1 shows the details of the data used in this study.

Both the reference data and the experiment data were collected by sensors with a 2000-Ω platinum resistor with a readout resolution of 0.1°C and an interval of 20 minutes. All the air temperature series data were then rescaled to degree centigrade. The reason for choosing six different time periods is that the stability of data in each time span is different and this difference makes them suitable for the sensitivity evaluation of PFSAA. As shown in Figure 1, both air temperature series data show periodical variations and these variations are similar to each other in adjacent periods. The illustrated outliers of the air temperature series data (marked with black circles in Figure 1) are identified by comparing with the reference data.

TABLE 1

A General Description of the Air Temperature Series Data Used in This Study

Sensors	Data Type	Length (20 minutes)	Time Span	
Sensor A	Experiment data	2235	January 1–31	July 1–31
Sensor B	Reference data	2235	January 1–31	July 1–31
Sensor C	Experiment data	1440	February 1–20	August 1–20
Sensor D	Reference data	1440	February 1–20	August 1–20
Sensor E	Experiment data	2235	March 1–31	May 1–31
Sensor F	Reference data	2235	March 1–31	May 1–31

3. PROBABILITY FINITE STATE AUTOMATA-BASED ALGORITHM

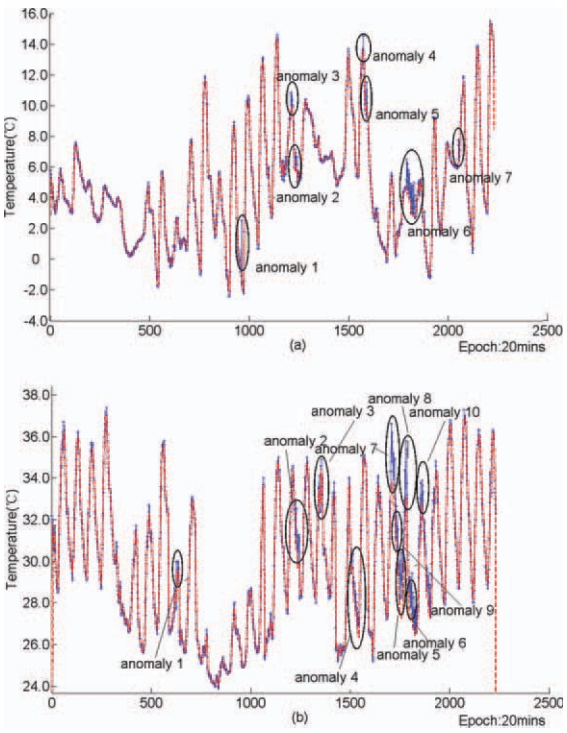
To overcome the drawbacks of conventional outlier detection methods summarized in Section 1, this study proposes a PFSAA method for outlier detection of air temperature series data. This new method is based on the combination of the time series theory [10] and the decision tree algorithm [11]. Specifically, it uses splitting and combining techniques and a learning-and-testing approach to realize the outlier detection. This method is composed of three steps. First, it splits the air temperature series data into some subclusters using the K-mean method [12]. Second, these subclusters are combined into relatively bigger clusters according to their similarities in relative entropy and variance. Finally, the PFSA are created based on the above two steps. Details of the three steps are described as follows.

3.1. Segmentation of Air Temperature Series Data

Generally, the key to an effective building of the dynamic module for time series data is to determine a reasonable segment level, which is actually the best cluster level [13]. The cluster level is vital for the segmentation of time series data due to its influence on the scale of finite state automata. Among traditional cluster methods, the K-mean algorithm has been widely used because it is computationally efficient in disclosing linear relationships [14]. This characteristic makes it suitable for the experiment data of this study. In this study, a slide window was used to classify the

DOI 10.1002/cplx

FIGURE 1



Air temperature series data in January (a) and July (b) 2010 and their outlier samples.

original air temperature series data into different clusters. When implementing the window, a parameter α was used to determine whether records should be classified within a same cluster or not. We set the value of α to be the average distance among all experiment data records.

3.2. Subclusters Combination

The approach of subcluster dividing described in Section 3.1 is limited in two ways. On one hand, the K-mean algorithm might mistakenly break a valid cluster with great disturbance into two subclusters. On the other hand, the slope alone might not be a reasonable criterion for cluster classification compared to the variation trend (e.g., slope) and scatter characteristics. To solve these problems, a similarity-driven cluster merging method was proposed to merge subclusters into relatively bigger ones based on relative entropy and variance (Figure 2). The relative entropy describes the density of a cluster which scatters along both sides of a fitting straight line. The relative variance represents the variations of variable values. The detailed descriptions of relative entropy and variance are provided below.

Definition 1

Let $\chi = \{\chi_1, \chi_2, \chi, \ldots, \chi_m\}$ be a sequence of observation, L_χ be the least-square fitting line of these points, and d_1, d_2, d_3, $\ldots$, d_m represent the corresponding distances from χ_1, χ_2, χ_3, $\ldots$, χ_m to L_χ, the relative entropy of χ can be defined as:

$$\mathrm{En} = -\sum_{k=1}^{m} P_k \log P_k \tag{1}$$

$$P_k = \frac{d_i}{\max\{d_1, d_2, \ldots, d_m\}} \tag{2}$$

Definition 2

Let $\chi = \{\chi_1, \chi, \chi_3, \ldots, \chi_m\}$ be a sequence of observation, L_χ be the least-square fitting line of these points, and d_1, d_2, d_3, $\ldots$, d_m represent the corresponding distances from χ_1, χ_2, χ_3, $\ldots$, χ_m to L_χ, the relative variance of χ can be defined as:

$$\mathrm{Ve} = \frac{1}{m}\sum_{k=1}^{m} (d_k - \overline{d})^2 \tag{3}$$

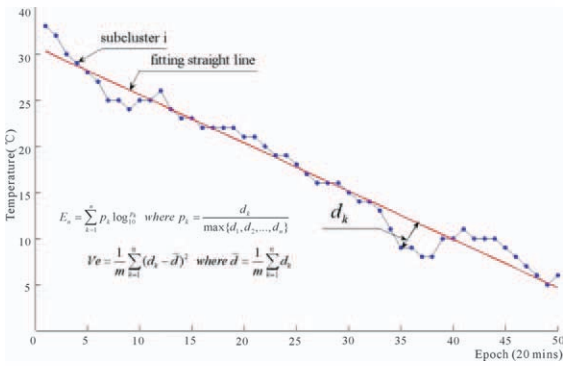
$$\overline{d} = \frac{1}{m}\sum_{k=1}^{m} d_k \tag{4}$$

Definition 3

Let $\chi = \{\chi_1, \chi_2, \chi_3, \ldots, \chi_m\}$ and $\gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_m\}$ be different sets of observations, the distance between any pair of them can be defined as:

$$\mathrm{Dis} = \left|\mathrm{Dis}_{\mathrm{inter}\chi} - Dis_{\mathrm{inter}\gamma}\right| + \left\|L_{k\chi} - L_{k\gamma}\right\|^2 \tag{5}$$

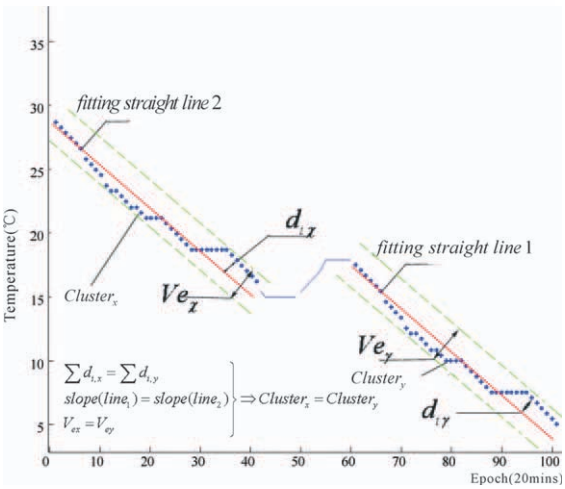
FIGURE 2



The basic principles of relative entropy and variance.

DOI 10.1002/cplx

FIGURE 3



Different clusters under similar slopes and relative entropies and variances.

where $\mathrm{Dis}_{\mathrm{inter}\chi}$ and $\mathrm{Dis}_{\mathrm{inter}\gamma}$ represent the distances between points within a same cluster. $\mathrm{Dis}_{\mathrm{inter}\chi}$ and $\mathrm{Dis}_{\mathrm{inter}\gamma}$ can be calculated by

$$\mathrm{Dis}_{\mathrm{inter}\chi} = \frac{1}{m}\sum_{k=2}^{m} |\chi_k - \chi_{k-1}| \tag{6}$$

$$\mathrm{Dis}_{\mathrm{inter}\gamma} = \frac{1}{m}\sum_{k=2}^{m} |\gamma_k - \gamma_{k-1}| \tag{7}$$

where $\|L_{k\chi} - L_{k\gamma}\|^2$ is the norm of the difference between $L_{k\chi}$ and $L_{k\gamma}$, which represent the fitting lines of χ and γ, respectively.

According to definitions 1, 2, and 3, as well as the illustration in Figure 2, it is clear that subclusters χ and γ are highly related to each other when their distance is small. In this way, subclusters χ and γ should have similar relative entropies and variances as shown in Figure 3. Figure 3 also shows that, if χ and γ have similar fitting straight lines, relative entropies, and variances, they keep similar structures. In addition, it is worth noting that the optimization threshold distance ν_{opt} used for subcluster combination was determined through an algorithm shown in Figure 4.

3.3. Generation of PFSA

Based on an adaptive multidimensional coded method [15], the combined subclusters can be tagged with a specific state number. The detailed training process of the probability matrix is shown in Figure 5. The main idea of this algorithm is to dynamically generate a probability matrix by computing the class of clusters within the training dataset. The training data were precleaned (i.e., removing the noises) by a kernel function $f = [1, 1, 1, 1, 1]$ (i.e., a source data x can be turned into $x' = \mathrm{conv}(x, f)/5$) and manually examined before being used to produce the reference PFSA. The threshold η in the algorithm was assigned as the value of ν_{opt} obtained in Section 3.2. After that, a probabilistic transfer matrix and a transfer diagram can be created using the Lambda algorithm [16]. The transfer diagram denotes the transformation between states of the time series data in a probabilistic way. The elements of the probabilistic matrix represent the probability of the state transfer relating to the

FIGURE 4

Algorithm 1:

Input: sub-cluster series Υ_i, threshold series $\theta_1, \theta_2, \ldots, \theta_n (\theta_1 < \theta_2 < \ldots < \theta_n)$

Output: the threshold ν_i

Step1. go through the element of threshold series $\theta_1, \theta_2, \ldots, \theta_n (\theta_1 < \theta_2 < \ldots < \theta_n)$;

Step1.1 for every sub-clusters of series Υ_i, combine two adjacent sub-cluster when the distance between them is less than the threshold θ_i and create a new sub-clusters Ψ_i;

Step1.2 compute the distance between two new sub-clusters both for intra-distance and inter-distance to get the total distance Φ_i;

Step2. for the distance serial Φ_i, plot it into Cartesian Coordinates;

Step3. fit the discrete point series Φ_i with polynomial-curve and get the express of fitting function;

Step4. compute the extreme value of the fitting function with derivate and get the final threshold ν_i which is used to produce the reasonable number of segmentation;

An algorithm for determining the optimization threshold distance, ν_{opt}, between subclusters.

DOI 10.1002/cplx

FIGURE 5

Algorithm 2:

// the purpose of the algorithm is to build a probability matrix with Lambda algorithm. The function of //sizeof(M) returns the numbers of row in matrix M. The s_i which is element of data set S is //corresponding to the i^{th} row of Maxtrix M .Cluster can be represented in forms of vector //$\mathbf{V}(k, ve, \chi)$ which make up of slope of fitting line, relative variant and relative entropy , that is //k, ve, χ respectively.

Input: state series denoted with value $\mathbf{V}(k, ve, \chi)$; η is a given threshold .

L is a map from V to S, i.e. L: $V \to S$

$\tilde{L}$ is a map from S to V,i.e. $\tilde{L}$:

$$\tilde{L}(s_i) = \frac{1}{card(V_i)} \sum_{v_i \in V_i} v_i \quad (V_i = \{v_i \mid L(v_i) = s_i\})$$

Output: M (probability matrix) and state value S

// Form the state transition matrix

N =number of vector V:

M (1. 1) =0:

$s_1 = L(v_1)$:

$S = \{s_1\}$:

$i = 1$;

While ($i <$ N) do

i = i +1;

If exist s_k in S satisfy with $s_k \in \{s_j \mid \| V_i - \tilde{L}(s_j) \| \le \eta\} \wedge \| V_i - \tilde{L}(s_k) \| = \min(\| V_i - \tilde{L}(s_j) \|)$

and i<N

Then $M(k,l) = M(k,l) + 1$;

Else

$i = i - 1$:

$\mathbf{M} = \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}$;

h=sizeof(M):

$\mathbf{M}(h, j) = 1$:

End

Else

h=sizeof(M):

$M = \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}$

$s_h = L(v_i)$

$S = S \cup \{s_h\}$

h=h+1;

End

End

// Form the state transition probability matrix from state transition matrix

state = [];

N=size (M):

$state(i) = \sum_{j}^{N} M(i, j) \quad i = 1, 2, \ldots, N$:

For i = 1 to N

For j = 1 to N

If $state(i)$ not equal to 0)

$M(i, j) = M(i, j) / state(i)$:

Else

$M(i, j) = 0$;

End

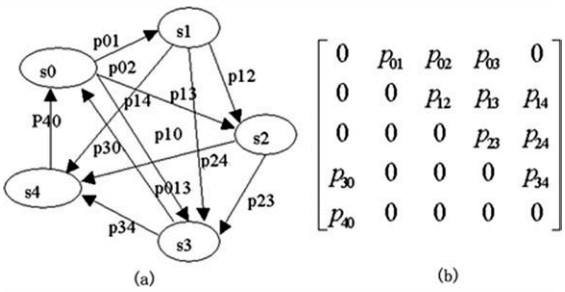
End

End

An algorithm for training probability matrix by computing the class of clusters.

DOI 10.1002/cplx

FIGURE 6



$$\begin{bmatrix} 0 & p_{01} & p_{02} & p_{03} & 0 \\ 0 & 0 & p_{12} & p_{13} & p_{14} \\ 0 & 0 & 0 & p_{23} & p_{24} \\ p_{30} & 0 & 0 & 0 & p_{34} \\ p_{40} & 0 & 0 & 0 & 0 \end{bmatrix}$$

An illustration of a transfer diagram and its transfer matrix (note: p_{ij} means the transformation probability from state S_i to S_j).

row and the column. An example of transfer diagram and its corresponding transfer matrix for time series data is illustrated in Figure 6.

3.4. Anomaly Detection

After the building of PFSA, the state of the air temperature series data and their transfer matrix could be created. A MATLAB (MATLAB, Version 7.0) program was then developed to implement the outlier detection process described in Sections 3.1, 3.2, and 3.3. The detailed process is clearly demonstrated in Figure 7.

4. EXPERIMENT DESIGN

In this study, the performance of PFSAA was first validated by comparing the detected outliers with the reference data. Then, we further evaluated the performance of PFSAA by comparing the detected outliers and those detected by the autoregressive (AR)-based back propagation (BP) neural net model [17], which has been proved robust in dealing with chaos series data.

Wilcoxon test, a nonparametric test for the similarity of two related samples, was used to compare the PFSAA-derived outliers with other values. Specifically, the values of outliers detected by PFSAA or the AR-based (BP) neural net model were marked as 1, while the reference values at relative locations were marked as 2. As the purpose of outlier detection is to identify data which is significantly deviated from the normal value, a significant difference between the detected outliers and the reference data (which represents the normal values) indicates a good performance of the outlier detection model, and vice versa. In this case, when the P value is equal to or less than 0.05, the outliers detected by PFSAA or AR-based BP neural net model are significantly different to the reference data, which suggests a good performance of the specific model. However, a P value greater than 0.05 indicates that there are no significant differences between the detected outliers and the reference data, therefore suggesting a poor performance of the specific model.

Additionally, the sensitivity of PFSAA to different data inputs was assessed based on the six air temperature series data listed in Table 1. The detailed settings of PFSAA parameters are shown in Table 2. It has to be noted that, in this study, the lengths of the training data listed in Table 2 were set up based on the stability of the experiment data from sensors. The criterion for determining the training data length is that the training data should cover at least

FIGURE 7

Algorithm3:
// **The purpose of this algorithm is to validate the observed state series.**
Input: a state set S is relative to probability transfer matrix M ; observed state series O ; **L is a map from V to S, i.e. L:** $V \to S$
Output:a state sequence T with tagged value '0' (i.e. normal state) or '1' (i.e. abnormal state) of state series O.
Step 1 : V=zeros(1,length(O)) //initiate the tag series with "0",which assumes all the
// observed states are normal
Step2: go through the element o_i of O
If L($L(o_i) \notin S$) ($o_i \in \mathrm{O}$)
$T_{i+1} = 1$;
Else
If $i > 1$ and M(L(o_i),L(o_{i+1}))=0
$T_{i+1} = 1$;
End ;
End;

An algorithm for anomaly detection implementation based on PFSA.

DOI 10.1002/cplx

TABLE 2

Parameters Settings of PFSAA in Outlier Detection

Sensors	Time Span	Training Samples (No.)	Test Samples (No.)	K-Means Threshold (α)	ν_{opt}
Sensor A	January 1–31	1100	1132	1.25	0.12
	July 1–31	1100	1132	1.75	0.22
Sensor C	February 1–20	500	940	0.75	0.1
	August 1–20	500	940	1.25	0.15
Sensor E	March 1–31	1100	1132	1.25	0.12
	May 1–31	1100	1132	1.75	0.12

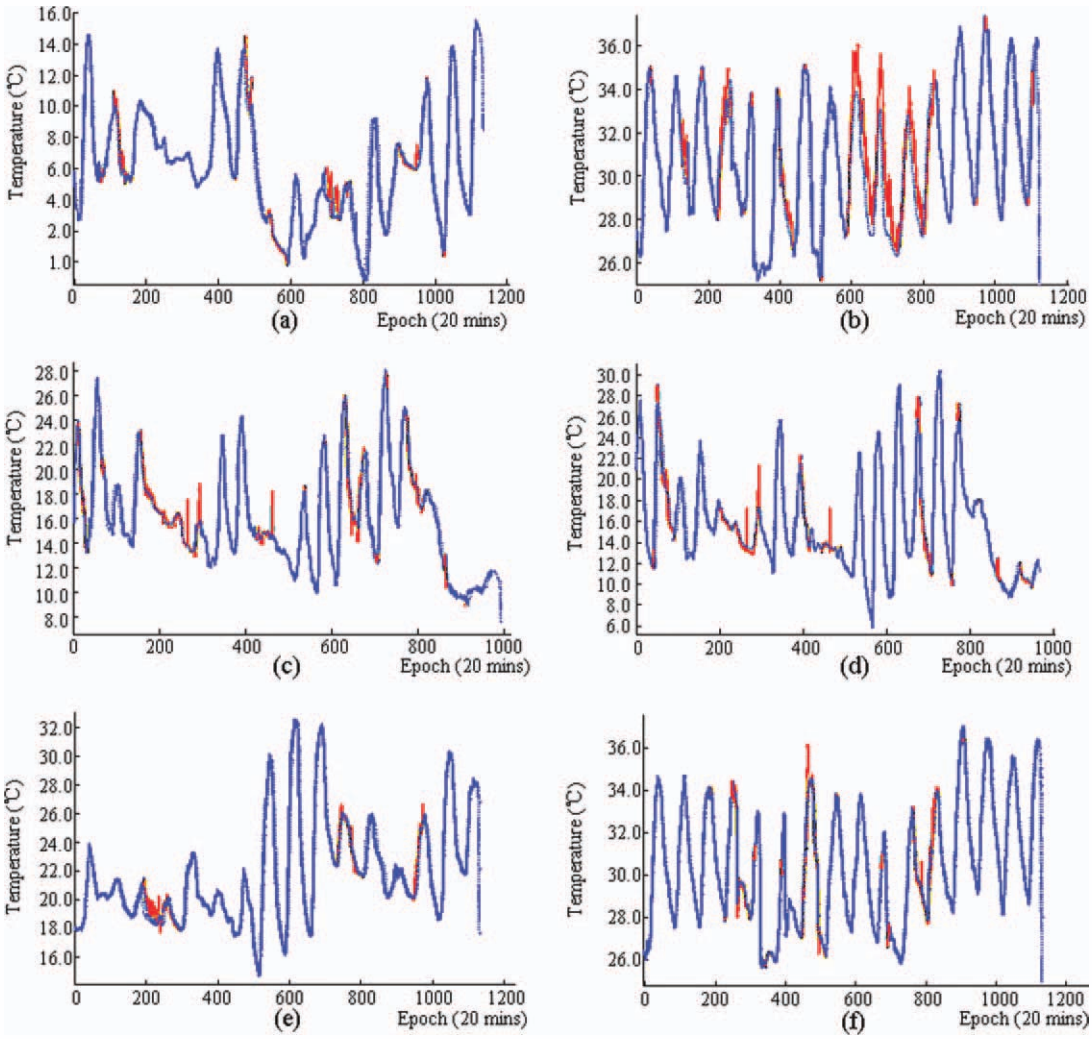
90% of the hidden states of the entire experiment data. For the AR-based BP neural net model, six orders were set to build the regression model of the training samples. The net is set with three layers and 10 hidden nodes.

5. RESULTS AND DISCUSSION

5.1. Performance of PFSAA as Validated by Reference Data

Figure 8 shows the comparison results between the outliers detected from six air temperature series experiment data and their corresponding reference data. The blue dot-

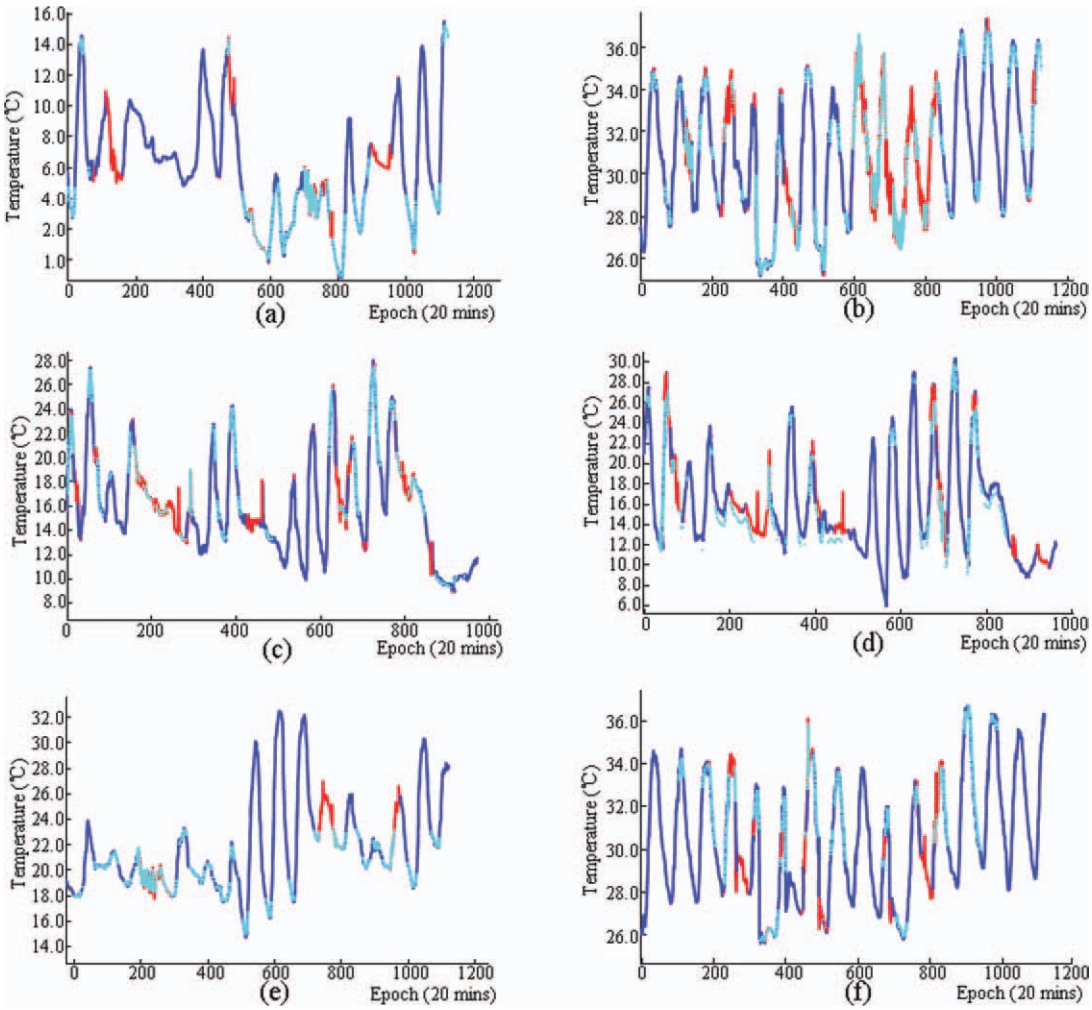
FIGURE 8



The comparison of outliers detected by PFSAA from the air temperature series experiment data with its relative reference data. (a), (b), (c), (d), (e), and (f) represent the results for data inputs in January, July, February, August, March and May, respectively.

DOI 10.1002/cplx

FIGURE 9



The comparison of outliers detected by PFSAA and the AR-based BP neural net model from the air temperature series experiment data. (a), (b), (c), (d), (e), and (f) represent the results for data inputs in January, July, February, August, March and May, respectively.

ted line represents the reference data. The red solid line denotes the outliers detected by PFSAA based on the experiment data. A single blue line represents those recognized as normal ones by PFSAA.

As shown in Figure 8, PFSAA successfully detected most parts of outliers (i.e., giant abnormal disturbances) caused by the error of sensor A. For example, outliers marked as red solid line in Figure 8(a) around locations between the 100th and the 140th interval, the 460th and the 490th interval, the 680th and the 730th interval, as well as the 1015th and the 1020th interval were correctly detected by PFSAA. However, Figure 8(a) also shows that a few outliers such as those around locations between the 530th and the 590th interval and those around locations between the 910th and the 970th interval were omitted by PFSAA although they represent a very small portion of the entire experiment data (i.e., total 120 versus 2235 interval). This omission may be attributed to the settings of the K-mean threshold (α) and the threshold distance ν_{opt} mentioned in Sections 3.1 and 3.2. These settings could be further optimized. Similar results that verify the feasibility of PFSAA in outlier detection can also be found in Figure 8(b–f).

5.2. Performance of PFSAA Evaluated by AR-Based BP Neural Net Model

Figure 9 shows the comparison results of outlier detection between PFSAA and the AR-based BP neural net model. The red solid line denotes the outliers detected by PFSAA, and the green dotted line represents those detected by the

DOI 10.1002/cplx

TABLE 3

Results of Nonparametric Wilcoxon Tests Comparing the Mean Air Temperature Values Detected by PFSAA as Outliers and Its Relative Reference Data

Sensors	Time Span	Outlier Detecting Methods	Z value	P value
Sensor A vs. B	January 1–31	PFSAA	5.409	0.000
		BP	2.917	0.055
	July 1–31	PFSAA	3.719	0.001
		BP	6.586	0.000
Sensor C vs. D	February 1–20	PFSAA	6.950	0.000
		BP	2.819	0.051
	August 1–20	PFSAA	1.689	0.001
		BP	12.462	0.000
Sensor E vs. F	March 1–31	PFSAA	6.699	0.000
		BP	2.635	0.077
	May 1–31	PFSAA	1.810	0.007
		BP	1.569	0.082

AR-based BP neural net model. A single blue line represents those recognized as normal ones by both methods.

Figure 9(a) reveals significant differences in the outliers detected by PFSAA and the AR-based BP neural net model. A comparison between the results presented in Figures 8(a) and 9(a) indicates that while the AR-based BP neural net model detected significant outliers around locations between the 680th and the 730th interval, a lot of faulty outliers are also detected. A comparison among Figure 9(a–f) further demonstrates that AR-based BP neural net model tagged much more fault outliers in January, February, March, May than in July and August. However, PFSAA does not have such problem as its accuracy of outlier detection remains relatively stable. This can be explained by the fact that PFSAA can more effectively detect outliers from the chaos data with relatively minor variations than the AR-based BP neural net model.

The better performance of PFSAA in outlier detection from air temperature series data is echoed by the results of nonparametric Wilcoxon tests. The difference is well reflected in the Z values and P values of the tests. For instance, the Z value (i.e., 5.409) of sensor A in January with a P value of 0.000 shows that there were significant differences between the air temperature values at outlier interval locations determined by PFSAA and the reference data. Similar results are also found for sensors C and E. However, we did not find any significant differences between the air temperature values at outlier interval locations determined by the AR-based BP neural net model from the experiment data and the reference data ($P > 0.05$) for all months except July and August. This founding in fact confirms that PFSAA has better stability than the AR-based BP neural net model in outlier detection.

5.3. Sensitivity of PFSAA in Outlier Detection of Air Temperature Series Data

As shown in Figures 8 and 9 as well as in Table 3, it is quite clear that the performance of PFSAA varied with the stability of the input data although it has better stability compared to the AR-based BP neural net model. To give a further quantitative illustration of this, we listed in Table 4 the parameters generated in the process of outlier detection by PFSAA. The value of Lyapunov index denotes the stability of the air temperature series data (i.e., lower value corresponds to better stability), while the error rate measures the performance of PFSAA. Table 4 shows that the state number, the initial stages, the combined clusters, and the error tag varied with the stability of the input data. As a result, the error rate varied correspondingly.

As shown in Table 4, Lyapunov indices are lower for relatively stable data inputs in January [Lyapunov index: 0.267; illustrated in Figure 8(a)], February [Lyapunov Index: 0.301; illustrated in Figure 8(c)], and March [Lyapunov index: 0.272; illustrated in Figure 8(e)] than for relatively unstable inputs in July [Lyapunov index: 0.353;

TABLE 4

Parameters Generated in the Process of Outlier Detection by PFSAA

Sensors	Time Span	Lyapunov Index	State Number	Initial Stages	Combining Clusters	Error Tags	Error Rate (%)
Sensor A	January 1–31	0.267	15	375	165	12	7.27
Sensor A	July 1–31	0.353	27	412	272	26	9.50
Sensor C	February 1–20	0.301	19	278	121	9	7.43
Sensor C	August 1–20	0.362	21	327	152	17	11.1
Sensor E	March 1–31	0.272	16	361	159	11	6.91
Sensor E	May 1–31	0.371	21	422	263	23	8.74

DOI 10.1002/cplx

illustrated in Figure 8(b)], August [Lyapunov index: 0.362; illustrated in Figure 8(d)], and May [Lyapunov index: 0.371; illustrated in Figure 8(f)]. The error rates illustrated from Figure 8(a), (b), (c), (d), (e), and (f) are 7.27%, 9.50%, 7.43%, 11.10%, 6.91%, and 8.74%, respectively. This finding confirms the influence of chaos stability on the performance of PFSAA. However, it has to be noted that the parameters (e.g., the error rate) listed in Table 4 might also vary with the change of the optimization threshold distance ν_{opt} mentioned in Section 3.2. Therefore, the parameter values generated in the process of outlier detection by PFSAA could be influenced by the setting of ν_{opt}, which deserves future investigation.

6. CONCLUSIONS

In this pilot study, we proposed a PFSAA method for outlier detection in air temperature series data. The case experiment with air temperature series data from three sensors shows that PFSAA can automatically detect outliers resulting from sensor errors at an acceptable level of accuracy compared to great time-consuming manual examination work. The sensitivity test also shows that the performance of PFSAA can vary with the stability of the input data. The PFSAA method represents a promising approach for tagging abruptly abnormal disturbances occurred in chaos series data. However, the lack of experiment samples prevented us from getting more reliable results. In addition, a possible extension of this work is to develop a probabilistic algorithm rather than a hard criteria for anomaly detection which is used by PFSAA, to compare the results of PFSAA with those of other robust models, and/or to quantitatively model the variation extent of PFSAA's performance in detecting outlier from chaos series data with different levels of stability.

ACKNOWLEDGMENTS

The work presented in this article was supported by the National Natural Science Foundation of China (No. 30570279). Bin Zou would like to thank the grants of the Freedom Explore Program (No. 1177-721500146) and the NieYing Talent Program (No. 1681-7601110176) from Central South University, Jun Shen would like to thanks the support of research program of Hunan meteorological administration (No. 201201).

REFERENCES

1. Cembrowski, G.; Chandler, E.; Westgard, J. Assessment of average of normals quality control procedures and guidelines for implementation. Am J Clin Pathol 1984, 81, 492–499.
2. Wu, J.; Aberer, K.; Tan, K. Towards integrated and efficient scientific sensor data processing: A database approach. Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, Saint Petersburg, Russia: ACM, 2009.
3. Sturza, M. Navigation system integrity monitoring using redundant measurements. Navigation 1988, 35, 483–501.
4. Di Francesco, R. Real-time speech segmentation using pitch and convexity jump models: Application to variable rate speech coding. Acoustics. IEEE Trans Speech Signal Process 2002, 38, 741–748.
5. Khatkhate, A.; Ray, A.; Keller, E.; Gupta, S.; Chin, S.C. Symbolic time-series analysis for anomaly detection in mechanical systems. IEEE/ASME Trans Mechatronics 2006, 11, 439–447.
6. Ray, A. Symbolic dynamic analysis of complex systems for anomaly detection. Signal Process 2004, 84, 1115–1130.
7. Davy, M.; Desobry, F.; Gretton, A.; Doncarli, C. An online support vector machine for abnormal events detection. Signal Process 2006, 86, 2009–2025.
8. Rajagopalan, V.; Ray, A. Symbolic time series analysis via wavelet-based partitioning. Signal Process 2006, 86, 3309–3320.
9. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. ACM Comput Surv (CSUR) 2009, 41, 1–58.
10. Salvador, S.; Chan, P. Learning states and rules for detecting anomalies in time series. Appl Intell 2005, 23, 241–255.
11. Dain, O.; Cunningham, R.; Boyer, S. Irep++ A faster rule learning algorithm. Proceedings of the Fourth SIAM International Conference on Data Mining, Lake Buena Vista, FL, 2004.
12. Zhou, S.; Zhao, Y.; Guan, J.; Huang, J. A neighborhood-based clustering algorithm. Adv Knowledge Discov Data Mining 2005, 3851, 361–371.
13. Tibshirani, R.; Walther, G.; Hastie, T. Estimating the number of clusters in a data set via the gap statistic. J Roy Stat Soc: Series B (Stat Methodol) 2001, 63, 411–423.
14. Kanungo, T.; Mount, M.D.; Netanyahu, S.N.; Piatko, C.; Silverman, R.; Wu, Y.A. An efficient k-means clustering algorithm: Analysis and implementation. IEEE Trans Pattern Anal Machine Intell 2002, 24, 881–892.
15. Hole, K.J.; Holm, H.; Oien, G. Adaptive multidimensional coded modulation over flat fading channels. IEEE J Sel Areas Commun 2000, 18, 1153–1158.
16. Heyman, D.; Lucantoni, D. Modeling multiple IP traffic streams with rate limits. IEEE/ACM Trans Network 2004, 11, 948–958.
17. Zhu, L.; Xue, K. A novel BP neural network model for traffic prediction of next generation network. In Fifth International Conference on Natural Computation, Tianjin, China, 2009.

DOI 10.1002/cplx