Expert Systems With Applications 57 (2016) 324–336

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

# Self-adaptive statistical process control for anomaly detection in time series

Dequan Zheng[a], Fenghuan Li[b,a,*], Tiejun Zhao[a]

[a] MOE-MS Key Laboratory of Natural Language Processing and Speech, Harbin Institute of Technology, Harbin 150001, PR China
[b] School of Software Engineering, South China University of Technology, Guangzhou 510006, PR China

ARTICLE INFO

*Article history:*
Received 5 January 2015
Revised 15 March 2016
Accepted 16 March 2016
Available online 24 March 2016

*Keywords:*
Anomaly detection
Self-adaptive
Statistical process control
Fuzzy set theory
Statistical testing

ABSTRACT

Anomaly detection in time series has become a widespread problem in the areas such as intrusion detection and industrial process monitoring. Major challenges in anomaly detection systems include unknown data distribution, control limit determination, multiple parameters, training data and fuzziness of 'anomaly'. Motivated by these considerations, a novel model is developed, whose salient feature is a synergistic combination of statistical and fuzzy set-based techniques. We view anomaly detection problem as a certain statistical hypothesis testing. Meanwhile, 'anomaly' itself includes fuzziness, therefore, can be described with fuzzy sets, which bring a facet of robustness to the overall scheme. Intensive fuzzification is engaged and plays an important role in the successive step of hypothesis testing. Because of intensive fuzzification, the proposed algorithm is distribution-free and self-adaptive, which solves the limitation of control limit and multiple parameters. The framework is realized in an unsupervised mode, leading to great portability and scalability. The performance is assessed in terms of ROC curve on university of California Riverside repository. A series of experiments show that the proposed approach can significantly increase the AUC, while the false alarm rate is improved. In particular, it is capable of detecting anomalies at the earliest possible time.

## 1. Introduction

Anomaly detection in time series provides significant information for numerous applications. For example, it can be used to detect intrusions in network data (Abadeh, Mohamadi, & Habibi, 2011), fraud detection (Ahmed, Mahmood, & Islam, 2016), incident faults in industrial process (Brighenti & Sanz-Bobi, 2011). Anomalies in time series can manifest in terms of the changes in the amplitude of data, or can be associated with the changes in the shape of temporal waveforms. In light of this, we categorize anomalies into two types: anomalies in amplitude and anomalies in shape. For example, it is an anomaly in amplitude that is a premature ventricular contraction in electrocardiogram (ECG) signals in Fig. 1 and it is an anomaly in shape that is a premature Poppet withdrawal in a Space Shuttle Marotta Valve time series shown in Fig. 2. These anomalous parts are highlighted in red in both figures.

Anomalies are time series that are the least similar to all other time series and depart from the bounds of the state of statistical control which exists when certain critical process variables remain close to their target values and do not change perceptibly. Time series that stay in a state of statistical control are called in-control data (normal data), otherwise, are called out-of-control data (anomaly). In statistical process control, control charts are used to determine if a process is in a state of statistical control. As shown in Fig. 3, a control chart consists of:

(1) Points representing a statistic of measurements of a quality characteristic in samples taken from the process at different times or different data.
(2) The mean of this statistic using all the samples at which a center line is drawn.
(3) Upper control limits (UCL) and lower control limits (LCL) that indicate the threshold at which the process output is considered statistically 'unlikely'.

Anomaly detection in time series is more challenging due to several reasons. First, it makes control limits very important decision aids. Control limits provide information about the process behavior and have no intrinsic relationship to any specification targets. In practice, the process mean (the center line) may not coincide with the specified value of the quality characteristic, because the process design simply cannot deliver the process characteristic

[*] Corresponding author. Tel.: +8645186412449606; fax: +8645186412449608.
*E-mail addresses:* dqzheng2007@gmail.com (D. Zheng), finelee2012@gmail.com (F. Li), tjzhao@hit.edu.cn (T. Zhao).

http://dx.doi.org/10.1016/j.eswa.2016.03.029
0957-4174/© 2016 Elsevier Ltd. All rights reserved.

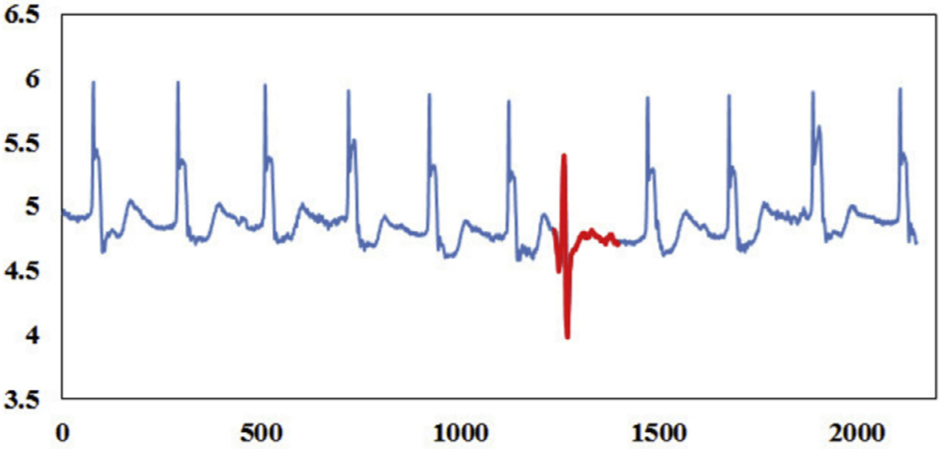

**Fig. 1.** The time series anomaly found in an excerpt of electrocardiogram.

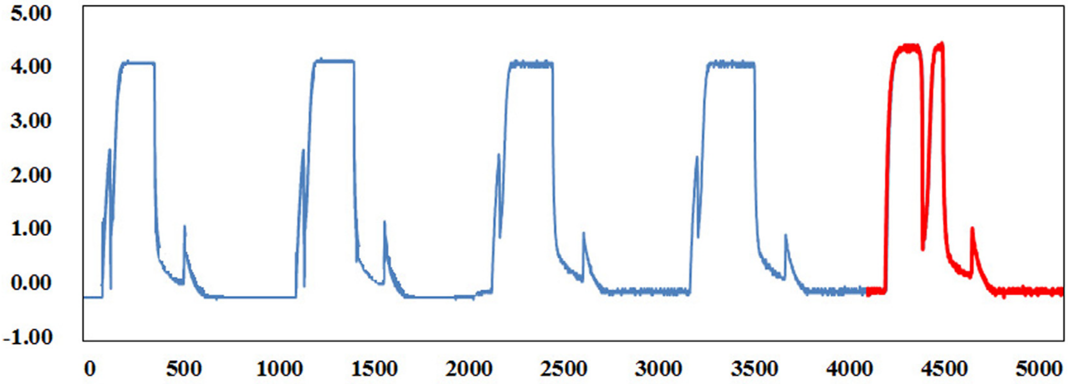

**Fig. 2.** An example of an annotated Marotta Valve time series.

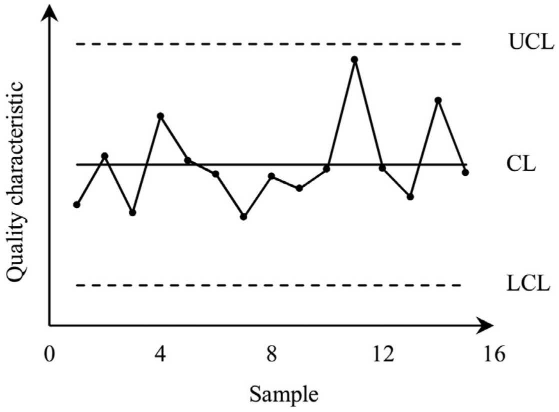

**Fig. 3.** An example of control chart.

at the desired level. It is also a key challenge to select a threshold instead of process mean. Second, anomaly is a more complex concept. For example, if one sample's characteristic is equal to UCL - $\varepsilon$ ($\varepsilon$ is an infinitesimal positive number), it is normal. But if one sample's characteristic is equal to UCL + $\varepsilon$, it becomes difficult to determine whether it is normal or abnormal. Third, many other algorithms require several parameters whose values are to be determined. This requires to acquire large amounts of training data, therefore, most of algorithms are realized in the supervised mode.

Due to these major challenges including unknown data distribution, control limit determination, multiple parameters, training data and fuzziness of 'anomaly' in anomaly detection systems, a synergistic combination of statistical and fuzzy set-based technique is proposed in this paper. We view anomaly detection as a statistical hypothesis testing and introduce a definition based on control chart in statistical process control. Because the process mean may not coincide with the specified value of the quality, we do not adopt the mean of samples' characteristic, but a threshold. Anomaly could be a more complex concept, so the threshold should be fuzzy. Fuzzy set theory is taken into account to provide a better characterization of the boundary between normal and abnormal. What's more, the inequality ($>$, $\leq$) in statistical hypothesis test is treated as a fuzzy predicate (the degree of inclusion). Intensive fuzzification process is adopted to realize related parameters determination which is self-adaptive. Therefore, the values of parameters are not required to be specified by the user. Due to the use of fuzzy set theory, statistical hypothesis testing in this paper is a distribution-free and totally unsupervised model. What's more, the overall scheme is self-adaptive. The utility is demonstrated using synthetic and real data sets. We have conducted a number of studies that show the effectiveness of our algorithm to detect anomalies in time series data.

The paper is structured as follows. Section 2 reviews some previous works on anomaly detection. Section 3 illustrates how anomaly detection can be viewed as a statistical hypothesis testing. A fuzzy-statistical algorithm for detecting anomalies is described in Section 4. We present some applications and perform extensive evaluation in Section 5 to demonstrate both the utility and ability to detect anomalies. Finally, the paper is summarized and concluded in Section 6.

## 2. Related works

The broad categories of anomaly detection techniques are: classification-based techniques (Koc, Mazzuchi, & Sarkani, 2012; Dangelo, Palmieri, Ficco, & Rampone, 2015), nearest neighbor-based techniques (Ceclio, Ottewill, Pretlove, & Thornhill, 2014; Sajjad, Bouk, & Yousaf, 2015; Lin, Ke, & Tsai, 2015), clustering-based techniques (Ahmed, Mahmood, & Maher, 2015; Lee, Kim, & Kim, 2011), statistical techniques (Zhang, Lu, Zhang, & Ruan, 2016; Pierazzi, Casolari, Colajanni, & Marchetti, 2016), information theoretic techniques (Drugman, 2014; Ashfaq, Rizvi, Javed, Khayam, Ali, & Al-Shaer, 2014), spectral techniques (Wang, Xiang, Markert, & Liang, 2016; Pfund, Anderson, Detwiler, Jarman, McDonald, Milbrath, Myjak, Paradis, Robinson, & Woodring, 2016). There are numerous methods proposed in literature to realize anomaly detection in time series. Quinn and Sugiyama (2014) presented a probabilistic, nonparametric method for anomaly detection in static and sequential data, based on a squared-loss objective function which had a simple analytical solution. Appice, Guccione, Malerba, and Ciampi (2014) illustrated a novel method that addressed both anomaly detection and change analysis in geophysical applications. The method extended the traditional time series forecasting theory by accounting for the spatial information of geophysical data. Akouemo and Povinelli (2015) introduced a probabilistic approach to anomaly detection, specifically in natural gas time series data. The system detected anomalies using a linear regression model with weather inputs, after which the anomalies were tested for false positives and classified using a Bayesian classifier.

The basic idea of process control is that a process always stays in a state of statistical control, unless a special event occurs. Oprea and Aben (2012) used control charts to map the deviations from the mean of each probe to do data analysis for traffic anomaly detection. Spiric, Docic, and Stankovic (2015) analyzed time series without the seasonal component of consumers' power consumption at low voltage in order to detect fraud and illogical consumption by customers. Statistical process control was used, where the process represented the process of using electricity. Kadri, Harrou, Chaabane, Sun, and Tahon (2016) reported the design of a new methodology for the detection of abnormal situations based on the integration of time-series analysis models and statistical process control tools for the joint development of a monitoring system to help supervising of the behavior of emergency department services. Though statistical process control can focus on the ongoing and prompt monitoring, it is difficult to get accurate control limits and it is necessary to have a sufficient number of observations (Spiric, Docic, & Stankovic, 2015).

One of commonly used techniques for anomaly detection in time series data is to analyze their neighborhoods. Shuchita and Karanjit (2012) focused on nearest neighbor based outlier detection techniques in their research and identified the advantages and disadvantages of various nearest neighbor based techniques. To overcome two weaknesses of the size of nearest neighborhood and the outlierness scores, Ha, Seok, and Lee (2015) proposed a novel outlier detection method involving an iterative random sampling procedure. In order to offer a heuristic guideline to determine the best size of nearest neighborhood, they additionally proposed to use the entropy of observability factor scores. Sajjad et al. (2015) presented an intrusion detection technique based on the calculation of trust of the neighboring node. Each node observed the trust level of its neighboring nodes. Based on these trust values, neighboring nodes may be declared as trustworthy, risky or malicious. The nearest neighbor-based technique was a nonparametric method. Despite its simplicity, its critical shortcomings are the low tolerance to noise and the fact that it relies exclusively on the existing data, assuming that the training set defines the decision boundaries among the classes perfectly, which is not always the case(Derrac, Garca, & Herrera, 2014).

There are extensive studies in statistics community regarding anomaly detection. They fit a statistical model to given data, then apply a statistical inference test to determine if an instance belongs to this model. Federico, Ignacio, de-la Higuera Pablo, Marcos, Dimitriadis, and Carlos (2011) proposed a method to detect anomalies in network traffic, based on a non-restricted $\alpha$-stable first-order model and statistical hypothesis testing. Harrou, Kadri, Chaabane, Tahon, and Sun (2015) aimed to present a statistical anomaly-detection scheme based on a principal component analysis model that can detect abnormal hospital emergency department demands. Bondavalli, Ceccarelli, Brancati, Santoro, and Vadursi (2016) investigated the possibility to monitor the evolution of indicators in dependable complex systems. The statistical analysis was based on a best-fitting approach aiming to minimize the integral distance between the empirical data distribution and some reference distributions. The main drawbacks of statistical methods are that statistical features are unknown and various, what is more, may not coincide with the specified value of the process characteristic and several parameters need to be determined(Koulaouzidis, Das, Cappiello, Mazomenos, Maharatna, Puddu, & Morgan, 2015). However, it has the ability to detect unseen patterns.

The reason that 'anomaly' itself includes fuzziness makes fuzzy set theories have been successfully applied to solve anomaly detection problems. Albertetti, Grossrieder, Ribaux, and Stoffel (2016) proposed an on-line method for detecting and querying change points in crime-related time series with the use of a meaningful representation and a fuzzy inference system. The approach proposed in the system (Lemos, Caminhas, & Gomide, 2013) was to detect new operation modes of a process such as operation point changes and faults. The approach relied upon an incremental clustering procedure to generate fuzzy rules. In the study (Kumarage, Khalil, Tari, & Zomaya, 2013), unsupervised data partitioning was performed adapting fuzzy c-means clustering in an incremental model. Fuzzy logic theory (Huang, Huang, & Liu, 2014) can avoid hard threshold, thereby increase the tolerance towards contradictions in the data. However, fuzzy rules and membership functions are difficult to determine. If the information is simple and insufficient, the performance will become worse.

## 3. Anomaly detection based on statistical process control

According to statistical process control, the statistic of a quality characteristic should be defined, which represents the anomaly score of the data. When the anomaly score is larger, the data is more likely to be anomaly. So, we just consider how large the anomaly score is and do not need to care how small it is. That is to say, we also need to determine the upper control limit, with regard to lower control limit. Now we illustrate how any anomaly detection scheme can be viewed as a statistical hypothesis. A statistical hypothesis testing is to discover whether a sample is or not significantly different from the present in the population. The testing is evaluated by exploring the fundamental tradeoff between the false positive and false negative rates. Hypothesis testing explains how to pick thresholds decision when one is faced with balancing this particular tradeoff. In fact, any anomaly detection method requires a threshold to be selected. So hypothesis testing is available and useful to discover whether the current subsequence is or not significantly different from the whole information. Any anomaly detection method will at some point use a statistical testing to verify whether a hypothesis is true or false. The problem is to choose between a single hypothesis $H0$, which is associated to the estimated stochastic model and the composite hypothesis $H1$, which represents all the other possibilities.

### 3.1. Anomaly score

Let us consider a time series $X$, which is an ordered sequence of measurements taken for a real-valued variable. $n$ is the number of subsequences in $X$. $P$ is one subsequence of the time series with length $q$, or a collection of $q$ features. A key component of an anomaly detection algorithm is the similarity used to determine how closely two given subsequences are matched. Each subsequences is associated with a set of measurements for $q$ features. We measure the similarity between two subsequences using Euclidean distance.

For each subsequence, we define the degree to which the subsequence differs from other subsequences in the same time series, termed the anomaly score. Anomaly score is defined as the average of the distances between a subsequence and its $K$ nearest neighbors. Formally speaking, the anomaly score of a subsequence $P$ is computed in the form:

$$Score(P) = \frac{1}{K}\sum_{t=1}^{K} Distance(P, P_t)|P_t \in KNNSet(P) \tag{1}$$

where $KNNSet(P)$ denotes the set composed by $K$ nearest neighbors of $P$ in time series, $Distance(P,\ P_t)$ is measured by Euclidean distance.

### 3.2. Anomaly detection as statistical hypothesis testing

Traditional hypothesis testing relies heavily on prior knowledge of the data distribution and is not suitable for arbitrary or new and unknown data sets. What is more, hypothesis testing is based on statistical estimation of the distribution parameters. Because we do not know the distribution, it is impossible to estimate its parameters. In practice, the process mean may not coincide with the specified value of the quality characteristic. So we consider a threshold to do hypothesis testing to determine whether one subsequence is one anomaly or not. A state of statistical control exists when anomaly score $\leq$ threshold. This threshold is the needed upper control limit. We form the following hypothesis $H0$ for the case when there is an anomaly, and form an alternate hypothesis $H1$ for the case when there is no anomaly.

$$\begin{aligned} &H_0 : anomalyscore > threshold \\ &H_1 : anomalyscore \leq threshold \end{aligned} \tag{2}$$

We neither know the distribution nor the values of its parameters, such as mean and standard error. But according to the definition of anomaly score, we can determine the average of the distances between a subsequence and its $K$ nearest neighbors. Meanwhile, standard deviation of this subsequence can be calculated. $T - test$ is a statistical hypothesis test in which the test statistic follows a Student's $t$ distribution if null hypothesis is supported. It can be used to determine if two sets of data are significantly different from each other, and is most commonly applied when the test statistic would follow a normal distribution if values of a scaling term in the test statistic are known. When the scaling term is unknown and is replaced by an estimate based on the data, the test statistic (under certain conditions) follows a Student's $t - test$. Student's $t - test$ doesn't need known distribution parameters, we can use the subsequence mean and standard deviation to replace distribution parameters. In the following section, we will illustrate why we do not need to care whether the test statistic would follow a normal distribution or not. When considered in the problem, the rejection region comes in the form:

$$t(P) = \frac{Score(P) - threshold}{s/\sqrt{K}} \leq t_{\alpha/2}(K-1) \tag{3}$$

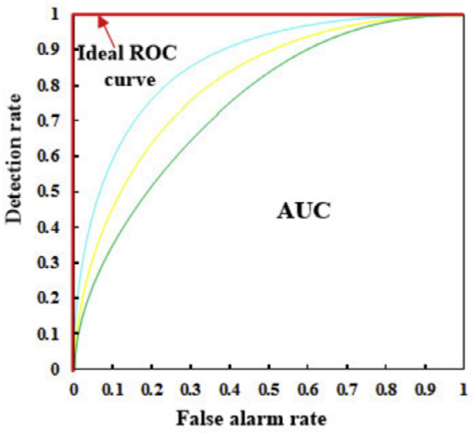

**Fig. 4.** Roc curves for different $K$.

Where $Score(P)$ is anomaly score of the subsequence $P$, $s$ is standard deviation of all distances between a subsequence and its $K$ nearest neighbors, and $\alpha$ is significance level. There are other two parameters to be determined, which are *threshold* and $K$. 'Anomaly' is a complex concept. For example, if anomaly score is equal to $threshold - \varepsilon$ ($\varepsilon$ is an infinitesimal positive number), this sample is normal. But if anomaly score is equal to $threshold + \varepsilon$, it becomes difficult to determine it is normal or not. In this sense fuzzy sets arise naturally in the study to avoid using a binary predicate 'below'. Correspondingly, the inequality ($>$, $\leq$) in statistical hypothesis test is treated as a fuzzy predicate (implying a certain degree of inclusion).

To evaluate the performance of the anomaly detection, we have used a Receiver Operating Characteristic (ROC) curve, which plots the true positive rate (TPR) (i.e. detection rate, recall, sensitivity) versus false positive rate (FPR) (i.e. one-specificity, false alarm rate) as shown in Fig. 4. A detection rate is the ratio between the number of correctly detected anomalies and the total number of anomalies. A false alarm rate is the ratio between the number of data records coming from normal class that are misclassified as anomalies and the total number of data records from normal class. ROC curve establishes a trade-off between detection rate and false alarm rate. An algorithm is considered to perform well if its ROC curve climbs rapidly towards the upper left corner of the graph. This means that we have detected a very high fraction of the true anomalies with only a few false positives. To quantify how quickly the ROC curve rises to the upper left corner, one determines the area under the curve (AUC). The larger the area, the better the algorithm. So we can view $K$ selection as an optimization problem to make the ROC curve climbing towards the upper left corner of the graph. The optimization model is defined as follows:

$$K = \arg\min_{K} \sqrt{(FPR - 0)^2 + (TPR - 1)^2} \tag{4}$$

The optimal ROC curves can be derived by optimizing $K$ whose value is estimated on a basis of the training set. Furthermore, only $K$ optimization needs training set in our algorithm. In this paper, we can consider a predefined value of $K$ which can assume any value. In this case, the algorithm is of unsupervised character.

## 4. Self-adaptive detection model

As mentioned earlier, 'anomaly' is a complex concept and therefore the threshold determination can be realized by engaging fuzzy sets. In this section, we show how to determine threshold using fuzzy set theory and a certain fuzzification process that leads to the treatment of inequalities ($>$, $\leq$) as fuzzy predicates. The aim of

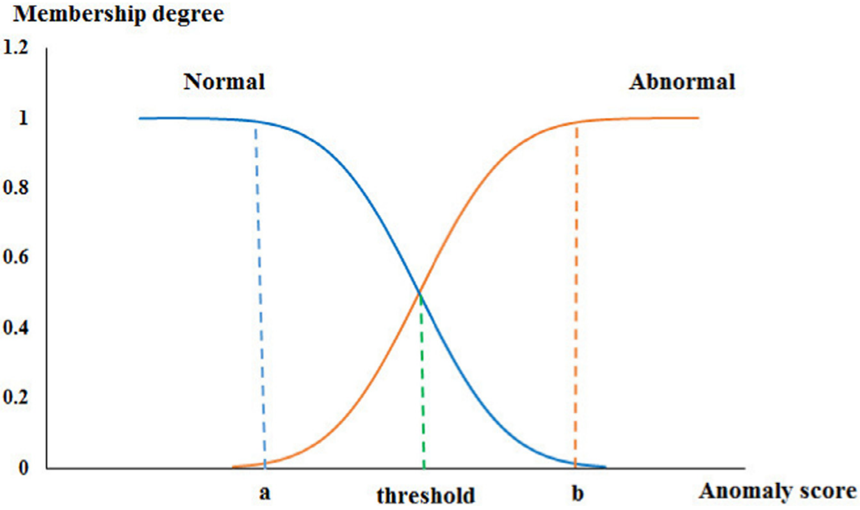

**Fig. 5.** Membership functions of the fuzzy sets.

fuzzification is to achieve optimized results. Let us also note that because of fuzzification, the algorithm is a distribution-free statistical testing.

### 4.1. Fuzzification

As our objective is to realize anomaly detection, we selected two fuzzy sets, namely normal and abnormal. Because of their smoothness, infinite support, and concise notation, Gaussian membership functions are their commonly encountered models. The Gaussian membership functions shown in Fig. 5, are described in the form:

$$normal(x) = \begin{cases} 1, & x \le a \\ exp(-(\frac{x-a}{\sigma})^2), & x > a \end{cases} \tag{5}$$

$$abnormal(x) = \begin{cases} exp(-(\frac{x-b}{\sigma})^2), & x \le b \\ 1, & x > b \end{cases} \tag{6}$$

Each membership function is characterized by two important parameters. The value $a$ (or $b$) represents the centre of the function, whereas $\sigma$ denotes the width (spread). Higher values of $\sigma$ correspond to larger spreads of the fuzzy sets. We select the same value of $\sigma$ for these two functions. In this case, we can regard $\sigma$ as a constant. So it is $a$ or $b$, not $\sigma$, which determines an overlap between these two functions. When $abnormal(x) > normal(x)$ , there is an anomaly. The threshold is given by the condition $abnormal(x) = normal(x)$.

Now we illustrate how to determine the values of parameters $a$ and $b$. In principle, a sound fuzzy membership function not only can reflect the fuzzy characteristics of a fuzzy concept, but also can describe the objective content as clear as possible. The clarity of objectivity can be measured by the fuzziness degree of one fuzzy set. When the fuzziness degree of one fuzzy set is smaller, the fuzzy set expresses the problem more clearly. So we use the principle of minimum of fuzziness degree to determine the values of related parameters. We select fuzzy entropy to express the fuzziness degree, leading to the optimization model in the following form:

$$\min H(a,b) = \frac{1}{N\ln 2}\sum_{i=1}^{N}\{s(normal(x_i)) + s(abnormal(x_i))\} \tag{7}$$

Where $N$ is the number of samples in the whole dataset, and

$$s(x) = \begin{cases} -x\ln x - (1-x)\ln(1-x), & x \in (0,1) \\ 0, & x = 0,1 \end{cases} \tag{8}$$

### 4.2. Intensive fuzzification

The fuzzy set-based scheme which is an intensive fuzzification process is proposed in this paper. It is shown in Fig. 6. The scheme consists of five major components as follows:

(1) Fuzziness degree inference: Fuzziness degree is expressed by membership functions in formula (5) and (6) whose parameters are unknown.
(2) Parameters optimization: Optimized $a$ and $b$ are achieved by the principle of minimum of fuzziness degree in formula (7).
(3) Fuzzy sets building: Fuzzy sets and membership functions are constructed according to optimized $a$, $b$ and predefined $\sigma$.
(4) Threshold determination: Threshold in formula (2) and (3) exists when two membership degrees are the same.
(5) Result estimation: Performance is evaluated and $K$ optimization is carried out according to the model in formula (4).

As shown in Fig. 6, the black line represents the first fuzzification in which anomaly score is the data object. The aim of the first fuzzification is to obtain an optimized threshold. The red line shows the second fuzzification. According to formula (3), we calculate the value of $t$ which is the data object for the second phase of the fuzzification. Because the optimized threshold is a numeric value and inequality ($\le$) is a fuzzy predicate, we conduct a second fuzzification to determine the degree of inclusion. The proposed scheme is developed in two stages: training stage and testing stage. Training stage and testing stage are almost the same. Both of them are intensive fuzzification scheme. Their only difference is $K$ optimization. Training stage is needed only when an optimized $K$ is required. Then, the optimized $K$ is used at testing stage. Otherwise, training stage and $K$ optimization are not necessary. $K$ will be predefined by users when testing. The detection algorithm is described in the form of Algorithm 1.

### 4.3. Distribution-free statistical testing

Now we illustrate why in our algorithm there is no need to concern about the distribution of the data. Only in the second fuzzification, we should consider the distribution of the data. As shown in formula (3), we have to calculate $t(P)$ and $t_{\alpha/2}(K-1)$. If $K$ and $\alpha$ are known, $t_{\alpha/2}(K-1)$ will be a constant. The predicted result of $t(P)$ will be the same as that of $t(P) - t_{\alpha/2}(K-1)$. Now we illustrate the reason. For example, $x$ and $y$ are two variables, and $y = x + h$, $h$ is a constant. If membership functions of

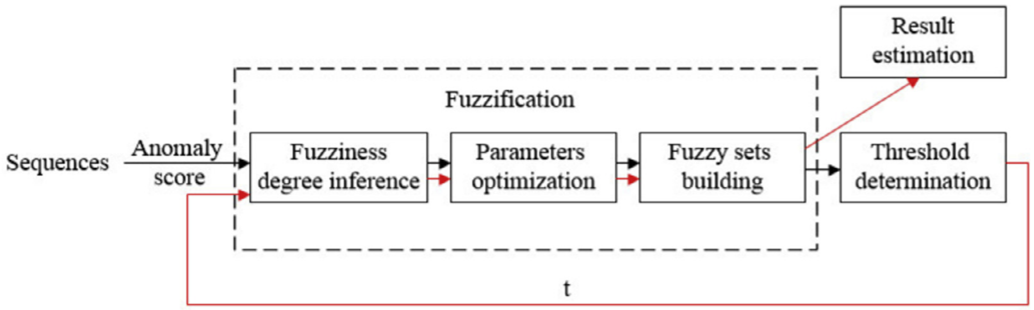

**Fig. 6.** Intensive fuzzification detection scheme.

**Algorithm 1** The procedure of self-adaptive detection algorithm.

```
Input: The time series $X = X_1, \ldots, X_n$
Output: The set of anomalies
for all $X_i$ do
    Calculate anomaly score $Score(X_i)$ according to formula (1)
    $normal(Score(X_i))$ and $abnormal(Score(X_i))$ of $Score(X_i)$ are expressed by formula (5) and (6), where $a$ and $b$ are unknown, and $\sigma$ is predefined
end for
$a$ and $b$ in the membership functions of anomaly score $Score(X_i)$ are optimized according to formula (7)
$threshold$ in formula (2) and (3) is the value of $x$ when $abnormal(x) = normal(x)$
for all $X_i$ do
    Calculate $t(X_i)$ according to formula (3)
    $normal(t(X_i))$ and $abnormal(t(X_i))$ of $t(X_i)$ are expressed by formula (5) and (6), where $a$ and $b$ are unknown, and $\sigma$ is predefined
end for
$a$ and $b$ in the membership functions of $t(X_i)$ are optimized according to formula (7)
for all $t(X_i)$ do
    if $abnormal(t(X_i)) > normal(t(X_i))$ then
        $X_i$ is an anomaly and accept $H0$
    else
        $X_i$ is normal and accept $H1$
    end if
end for
```

variable $x$ are $normal(x)$ and $abnormal(x)$, membership functions of $y$ are $normal(y-h)$ and $abnormal(y-h)$. $abnormal(y-h)$ and $normal(y-h)$ locate to the right of $abnormal(x)$ and $normal(x)$ as shown in Fig. 7. The distances between them are $h$ and shapes of their membership functions are the same. $CrossX$ and $CrossY$ are crossing points of their membership functions and distances between them are $h$. It means $CrossY = CrossX + h$. So the hypothesis $x < CrossX$ is equivalent to the hypothesis $y < CrossY$. Additionally, if their membership functions can be determined, their crossing points can be determined respectively without caring of the value of $h$. So we do not need to care whether the data follow a normal distribution or not. Therefore, we do not consider $t_{\alpha/2}(K-1)$ in formula (3), and just regard $t(P)$ as the test object.

## 5. Experiments and discussions

We begin experiments by showing the usefulness of the proposed algorithm for synthetic data and real-life data, including anomalies in shape and amplitude. Then, we perform several experiments to evaluate its performance. Finally, we contrast our algorithm against several baseline algorithms to show that the proposed algorithm is able to efficiently find anomalies. In our experiments, it is workable without training process. $K$ can assume any value.

### 5.1. Effectiveness

Here, we consider synthetic time series in which there are some anomalies in shape. In Fig. 8, we consider a simple time series consisting of a slightly noisy sine wave. We introduce an anomaly of length 40 by shifting the entire second half of the time series. When training, we extract all subsequences of length 40, then project them into 40-dimensional space and measure their $K$ nearest neighbors. $K$ may be all possible candidates. Anomaly scores of all subsequences are determined by the average of their $KNN$ distances. Then fuzzification process is adopted to determine all parameters. According to training set, $K$ should be 11. So when testing, we just assign 11 to $K$. The following process is the same as training process. In Fig. 8, if the predefined length of the subsequence is different, the results may be different.

We used a dataset from UC (University of California) Irvine KDD Archive (http://kdd.ics.uci.edu) to detect synthetic anomalies in amplitude. This dataset is called “Pesudo Periodic Synthetic Time Series”. Actual time series are generated by formula (9). The data is interesting because it has different structures at different resolutions. Time series plotted in blue in Fig. 9 are generated by independent invocations of the function. Where, $0 \le \bar{t} \le 1$ and $rand(x)$ produces a random integer between zero and $x$. The data appears highly periodic, but never exactly repeats itself. All data points are in the range -0.5 to +0.5. The subsequence in red in Fig. 9 is generated by interjecting a large amplitude noise. The length of the subsequence is 7,267. The value of $K$ can be estimated from the training set. This optimized value of $K$ is 4.

$$\bar{y} = \sum_{i=3}^{7} \frac{1}{2^i} \sin\left(2\pi\left(2^{2+i} + rand\left(2^i\right)\right)\bar{t}\right) \tag{9}$$

We detect anomalies in shape in some real-world datasets. In Fig. 2, it is a Space Shuttle Marotta Valve time series in which there are five subsequences whose lengths are 998. The data is available at the following URL (Keogh, 2005). In Fig. 2, the expert annotated the last in a series of 5 energize/deenergize cycles as “Poppet pulled significantly out of the solenoid before energizing”. The problem is immediately detected by the algorithm proposed in this paper. The optimized value of $K$ is 2.

Another experiment is to demonstrate the effectiveness of our algorithm for finding anomalies in amplitude in real world. Fig. 1 gives a visual intuition of an anomaly found in a human electrocardiogram. The data is available at the URL (Keogh, 2005). The fact that the detected anomaly in Fig. 1 coincides with the location annotated by a cardiologist as containing an anomalous heartbeat hints. In this figure, we could perhaps spot the anomaly by

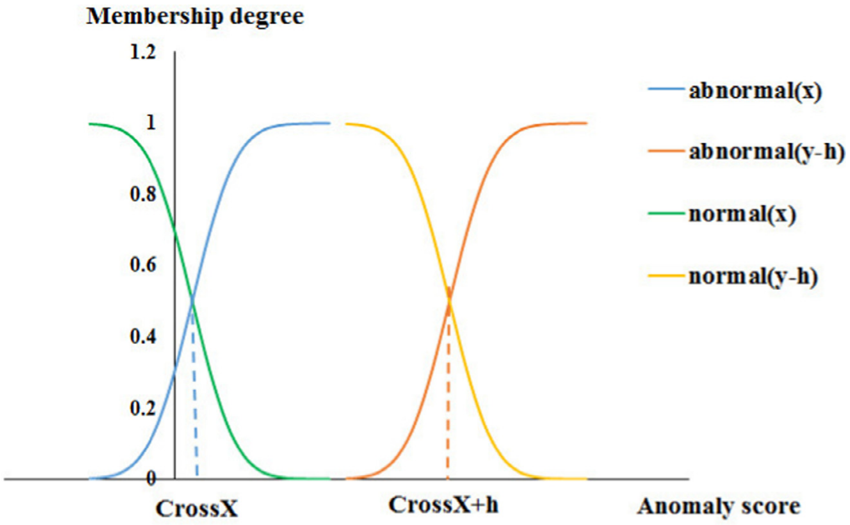

**Fig. 7.** Membership functions with different variables.

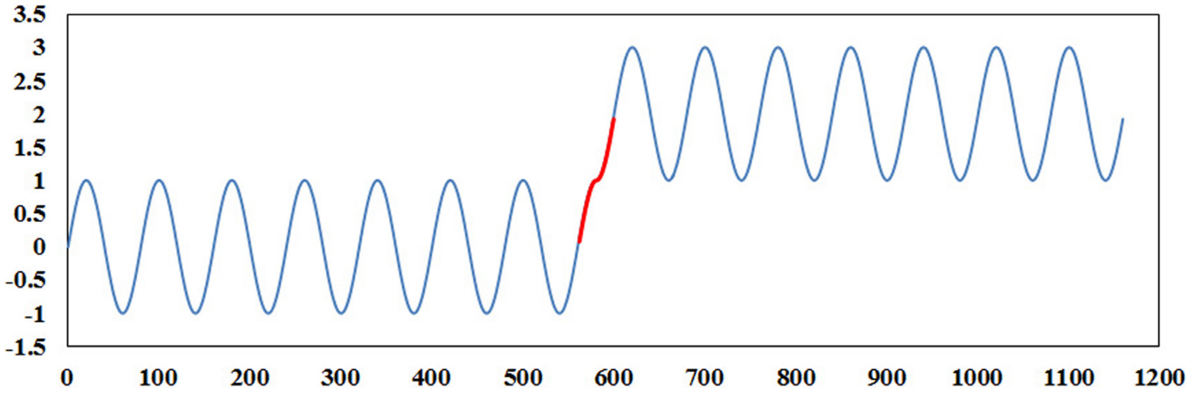

**Fig. 8.** A synthetic time series with an obvious anomaly in shape highlighted in red.

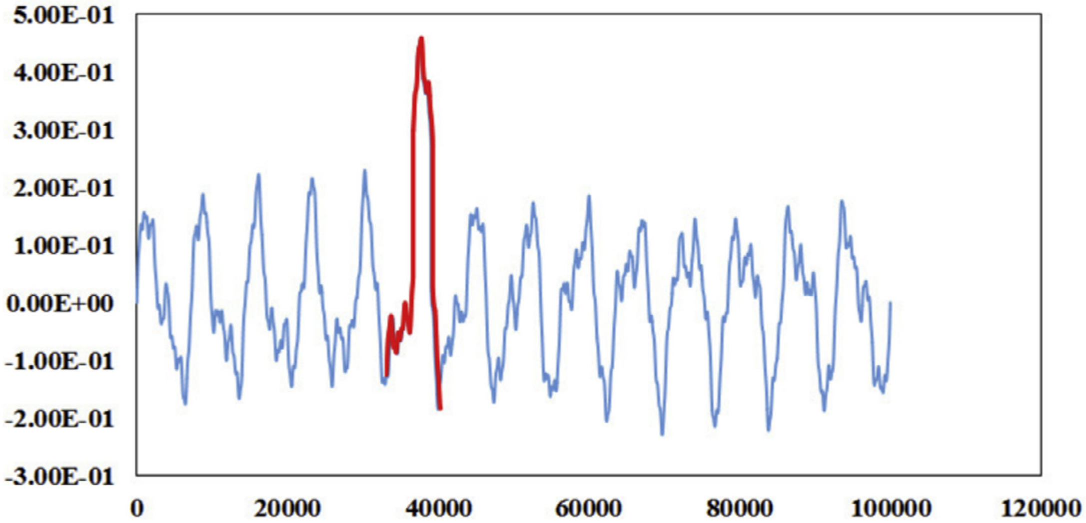

**Fig. 9.** A synthetic time series with an obvious anomaly in amplitude highlighted in red.

eye. However, the full time series is much longer, and impossible to scrutinize without a scrollbar and much patience. In this case, we simply set the length of the subsequences to be approximately one full heartbeat which is 393. And the optimized value of $K$ is 6. Two persons' heartbeats may be different, so the length of subsequences in two electrocardiograms may be different.

### 5.2. Performance analysis

We now analyze the performance of some techniques in our algorithm. A key feature of the algorithm is a combination of both statistics-based techniques and fuzzy set theory. Because every data set may have its own characteristics, we do not use a pre-

**Table 1**
Information of each dataset in the first part of UCR repository.

| No. | Name | Number of classes | Size of training set | Size of testing set | Time series length | Normal class | Abnormal class |
|---|---|---|---|---|---|---|---|
| 1 | 50words | 50 | 450 | 455 | 270 | 1 | 44 |
| 2 | Adiac | 37 | 390 | 391 | 176 | 25 | 4 |
| 3 | Beef | 5 | 30 | 30 | 470 | 1 | 5 |
| 4 | CBF | 3 | 30 | 900 | 128 | 3 | 2 |
| 5 | Coffee | 2 | 28 | 28 | 286 | 0 | 1 |
| 6 | ECG200 | 2 | 100 | 100 | 96 | 1 | −1 |
| 7 | FaceAll | 14 | 560 | 1690 | 131 | 13 | 11 |
| 8 | FaceFour | 4 | 24 | 88 | 350 | 2 | 1 |
| 9 | fish | 7 | 175 | 175 | 463 | 4 | 7 |
| 10 | Gun_Point | 2 | 50 | 150 | 150 | 1 | 2 |
| 11 | Lighting2 | 2 | 60 | 61 | 637 | 1 | −1 |
| 12 | Lighting7 | 7 | 70 | 73 | 319 | 5 | 2 |
| 13 | OliveOil | 4 | 30 | 30 | 570 | 4 | 3 |
| 14 | OSULeaf | 6 | 200 | 242 | 427 | 2 | 6 |
| 15 | SwedishLeaf | 15 | 500 | 625 | 128 | 10 | 14 |
| 16 | synthetic_control | 6 | 300 | 300 | 60 | 1 | 6 |
| 17 | Trace | 4 | 100 | 100 | 275 | 2 | 4 |
| 18 | Two_Patterns | 4 | 1000 | 4000 | 128 | 1 | 4 |
| 19 | wafer | 2 | 1000 | 6174 | 152 | 1 | −1 |
| 20 | yoga | 2 | 300 | 3000 | 426 | 2 | 1 |

defined and same $K$ for all datasets. An optimization model is constructed to select $K$. Of course, if there are limited training data, one can use a predefined $K$ for a dataset and the same $K$ for all datasets. In this case, the algorithm will be a totally unsupervised model. According to characteristics of our algorithm, we perform four experiments in this section in order to show the necessary of these techniques. The four experiments are listed as follows:

(1) The algorithm based on statistical testing and a predefined $K$ is called SP.
(2) The algorithm based on statistical testing and an optimized $K$ is called SO.
(3) The algorithm based on statistical testing, fuzzy set theory and a predefined $K$ is called SFP.
(4) The algorithm based on statistical testing, fuzzy set theory and an optimized $K$ is called SFO.

We perform these four experiments using datasets obtained from university of California Riverside(UCR) time series repository (Chen, Keogh, Hu, Begum, Bagnall, Mueen, & Batista, 2015). The repository contains 45 datasets. Because of space limit, we list the results of 20 datasets in the first part of the repository. Information of every dataset is listed in Table 1. We select the class with the most instances as the normal class, and the class with the fewest instances as the anomaly class. Each dataset is fairly organized so that the resulting dataset consists less than 20% anomalies. We analyze the performance from two points of view: $K$ selection and fuzzification. In the first (SP) and third (SFP) experiments, we select a fixed $K(9)$ for the experiments. In other two experiments, optimized $K$ is used. Of course, in these two experiments, optimized $K$ may be different for different datasets. The performance is measured in terms of TPR, FPR and AUC. We list all results in 20 datasets in Table 2. These results show the necessity of $K$ optimization and fuzzification.

Obviously, when $K$ is different, the performance may be much different. We can come up with this conclusion from performance comparison of SO with SP(9) and comparison of SFO with SFP(9) in Table 2. Taking the comparison of SO with SP(9) as example, better results with $K$ optimization are marked in green. All AUC values of SO are better than those of SP(9). Almost all TPR values and FPR values remain unchanged or are slightly better. The performance is improved greatly. The same conclusions occur in the performance comparison of SFO with SFP(9). That means, if training data are sufficient, $K$ optimization is necessary for performance improvement. If there is no training data, $K$ can be any value which one likes. In this case, detection rate and false alarm rate almost are not affected by different $K$.

The performance analysis of fuzzification is shown by comparing SFP(9) with SP(9) and comparing of SFO with SO in Table 2. Taking the comparison of SFO with SO as example the comparison of SFO with SO, where better results with fuzzification are marked in red. Most AUC values of SFO are better than those obtained by the SO. All FPR values are smaller and most TPR values stay the same. There is the largest improvement in false alarm rate. This means that the fuzzification can effectively reduce false alarm rate. The SFO algorithm outperforms other three algorithms on most of the datasets. Fuzzification is workable and effective for performance improvement.

## 5.3. Performance comparison

We compare our algorithm against 4 competing methods, which are LOF (Lee, Kang, & Kang, 2011), Brute force (Li, Xiong, & Liu, 2012), Random Walk(Moonesinghe & Tan, 2008) and $K$-distance (Nurunnabi, West, & Belton, 2015) using 20 datasets described in Table 1. LOF computes the anomaly score in terms of the ratio between the density of an object to the density of its $K$ nearest neighbors. $K$-distance uses the distance between an object and its $Kth$ nearest neighbor as the anomaly score. Brute force finds the distance to the nearest non-self match. The subsequence that has the greatest such value is the discord. LOF and $K$-distance use a threshold called minimum points ($K$) to define the neighborhood. Selection of threshold-$K$ for these approaches must be specified by the user. Random walk computes the connectivity value of each node. Nodes with high connectivity are considered normal whereas those with low connectivity are declared as anomalous. All baseline methods need the parameter $N$ to determine the number of predicted anomalies. Parameter $N$ must also be specified by the user. Our algorithm is distribution-free, unsupervised, self-adaptive and does not require any predefined values of the parameters.

### 5.3.1. Self-adaptivity

Different $N$ may result in different conclusions. What is more, sizes of different datasets are different or sizes of the datasets are unknown. This leads to the difficulty in the selection of the value of $N$. The ROC curve is generated by plotting the cumulative distribution function of the detection probability and the false alarm probability. AUC is merely determined by anomaly score. Therefore, different values of $N$ may lead to different values of TPR and

**Table 2**
Performance analysis of several techniques in our algorithm.

| No. | SP(9) | | | SO | | | SFP(9) | | | SFO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TPR | FPR | AUC | TPR | FPR | AUC | TPR | FPR | AUC | TPR | FPR | AUC |
| 1 | 1.00 | 0.47 | 0.98 | 1.00 | 0.47 | 0.98 | 1.00 | 0.02 | 0.99 | 1.00 | 0.02 | 0.99 |
| 2 | 1.00 | 0.56 | 0.73 | 1.00 | 0.56 | 0.77 | 1.00 | 0.31 | 0.69 | 1.00 | 0.44 | 0.81 |
| 3 | 1.00 | 0.00 | 1.00 | 1.00 | 0.00 | 1.00 | 1.00 | 0.00 | 1.00 | 1.00 | 0.00 | 1.00 |
| 4 | 1.00 | 0.34 | 0.92 | 1.00 | 0.26 | 0.95 | 1.00 | 0.27 | 0.91 | 1.00 | 0.04 | 1.00 |
| 5 | 0.67 | 0.40 | 0.53 | 0.67 | 0.40 | 0.64 | 0.67 | 0.40 | 0.54 | 0.67 | 0.40 | 0.64 |
| 6 | 0.92 | 0.31 | 0.80 | 0.92 | 0.30 | 0.87 | 0.92 | 0.33 | 0.89 | 1.00 | 0.28 | 0.92 |
| 7 | 1.00 | 0.49 | 0.88 | 1.00 | 0.47 | 0.91 | 1.00 | 0.19 | 0.74 | 1.00 | 0.16 | 0.93 |
| 8 | 1.00 | 0.50 | 0.67 | 1.00 | 0.50 | 0.70 | 0.80 | 0.38 | 0.69 | 1.00 | 0.38 | 0.71 |
| 9 | 0.60 | 0.28 | 0.72 | 0.60 | 0.28 | 0.77 | 0.00 | 0.03 | 0.48 | 0.00 | 0.03 | 0.48 |
| 10 | 0.73 | 0.34 | 0.78 | 0.73 | 0.38 | 0.81 | 0.60 | 0.33 | 0.63 | 0.60 | 0.37 | 0.65 |
| 11 | 0.33 | 0.58 | 0.40 | 0.33 | 0.58 | 0.42 | 0.33 | 0.55 | 0.42 | 0.33 | 0.52 | 0.45 |
| 12 | 1.00 | 0.42 | 0.86 | 1.00 | 0.42 | 0.88 | 0.67 | 0.16 | 0.88 | 0.67 | 0.16 | 0.90 |
| 13 | 0.50 | 0.17 | 0.88 | 0.50 | 0.17 | 0.88 | 0.50 | 0.08 | 0.94 | 0.50 | 0.08 | 0.94 |
| 14 | 0.82 | 0.56 | 0.76 | 0.82 | 0.56 | 0.76 | 0.55 | 0.20 | 0.74 | 0.82 | 0.44 | 0.81 |
| 15 | 0.44 | 0.67 | 0.27 | 0.44 | 0.67 | 0.27 | 0.00 | 0.55 | 0.28 | 0.00 | 0.55 | 0.28 |
| 16 | 0.00 | 1.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.88 | 0.00 | 0.00 | 0.90 | 0.00 |
| 17 | 1.00 | 0.34 | 0.87 | 1.00 | 0.31 | 0.97 | 1.00 | 0.21 | 0.98 | 1.00 | 0.07 | 1.00 |
| 18 | 0.96 | 0.58 | 0.86 | 0.96 | 0.57 | 0.88 | 0.85 | 0.22 | 0.89 | 0.85 | 0.22 | 0.89 |
| 19 | 0.71 | 0.47 | 0.67 | 0.71 | 0.44 | 0.69 | 0.93 | 0.72 | 0.62 | 0.66 | 0.39 | 0.64 |
| 20 | 0.82 | 0.72 | 0.58 | 0.81 | 0.71 | 0.58 | 0.59 | 0.40 | 0.64 | 0.58 | 0.39 | 0.64 |

FPR. However, there is no effect on the values of the AUC. In general, when $N$ is larger, TPR is better and FPR is worse. Otherwise, TPR becomes worse and FPR gets better. For comparison, we select the same value of $N$, which is automatically generated by our algorithm, for the four baseline algorithms. $K$ can be any value for SFP, LOF and $K$-distance. We just show the results when $K$ is 6 in Table 3. The performance is measured in terms of TPR, FPR and AUC.

As shown in Table 3, best TPR is marked in red, meanwhile, best FPR is marked in green and best AUC is marked in blue. More best results means the performance of this algorithm is better. SFP (6) has 12 best TPR, 13 best FPR and 13 best AUC. $K$-distance has 10 best TPR, 10 best FPR and 7 best AUC. LOF has 10 best TPR, 12 best FPR and 2 best AUC. Brute force has 9 best TPR, 10 best FPR and 4 best AUC. Random walk has 12 best TPR, 10 best FPR and 6 best AUC. Because of fuzzification, the proposed algorithm is self-adaptive. It means that the analysis procedure can process parameters according to the characteristic of the data and algorithm automatically, so that the algorithm coincides with the data and best performance is gained. If same $K$ and $N$ are predefined for all algorithms, the number of anomalies in every dataset and the characteristic of every algorithm may be different. If $K$ is too small, it is difficult to show the difference between data, so the abnormal data may be regarded as normal data, leading to bad FPR. If $N$ is too small, too many abnormal data will not be detected, leading to bad TPR. Otherwise, if $K$ or $N$ is too large when the number of data is limited, the detection may be out-of-range. If $K$ is too large, the boundary between 'normal' and 'abnormal' will become indistinct. Meanwhile, the performance is sensitive to $N$, therefore, large $N$ will cause bad FPR. Therefore, when $K$ is predefined and $N$ is the same for all algorithms, the performance of our algorithm is the best.Therefore, when $K$ is predefined and $N$ is the same for all algorithms, the performance of the proposed algorithm is the best.

### *5.3.2. Parameters optimization*

In the previous section, we use predefined value of $K$ and the same $N$ for all algorithms. $K$ can be optimized in the algorithm. In this section, we compare the best performance of all algorithms. $N$ has a significant influence on TPR and FPR which can not reach their maximum values at the same $N$. As we all know, when $N$ is larger, TPR is better and FPR is worse. Otherwise, TPR is worse and FPR is better. Therefore, there is no meaning to compare best TPR and best FPR of four compared algorithms with ours. We just show the best AUC values for all methods using the 20 datasets. We have

**Table 3**
Performance comparison when K is 6 and N is same to ours.

| No. | SFP(6) | | | K-distance | | | LOF | | | Brute Force | | | Random Walk | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TPR | FPR | AUC | TPR | FPR | AUC | TPR | FPR | AUC | TPR | FPR | AUC | TPR | FPR | AUC |
| 1 | 1.00 | 0.02 | 0.99 | 1.00 | 0.02 | 0.98 | 1.00 | 0.02 | 0.98 | 1.00 | 0.02 | 0.98 | 1.00 | 0.47 | 0.78 |
| 2 | 1.00 | 0.44 | 0.81 | 1.00 | 0.44 | 0.79 | 0.67 | 0.50 | 0.67 | 1.00 | 0.44 | 0.79 | 1.00 | 0.44 | 0.75 |
| 3 | 1.00 | 0.00 | 1.00 | 1.00 | 0.00 | 1.00 | 0.00 | 0.17 | 0.08 | 1.00 | 0.00 | 1.00 | 0.00 | 0.17 | 0.50 |
| 4 | 1.00 | 0.26 | 1.00 | 1.00 | 0.26 | 0.97 | 0.18 | 0.42 | 0.34 | 0.90 | 0.28 | 0.85 | 1.00 | 0.26 | 1.00 |
| 5 | 0.67 | 0.40 | 0.53 | 0.67 | 0.40 | 0.64 | 0.67 | 0.40 | 0.49 | 1.00 | 0.33 | 0.88 | 0.67 | 0.40 | 0.69 |
| 6 | 0.92 | 0.30 | 0.90 | 0.92 | 0.30 | 0.90 | 0.67 | 0.34 | 0.74 | 0.83 | 0.31 | 0.71 | 1.00 | 0.28 | 0.89 |
| 7 | 1.00 | 0.20 | 0.62 | 1.00 | 0.20 | 0.98 | 1.00 | 0.20 | 0.94 | 0.75 | 0.21 | 0.84 | 1.00 | 0.20 | 0.98 |
| 8 | 1.00 | 0.38 | 0.71 | 0.80 | 0.42 | 0.69 | 0.60 | 0.46 | 0.61 | 0.40 | 0.50 | 0.26 | 0.60 | 0.46 | 0.71 |
| 9 | 0.00 | 0.03 | 0.48 | 0.00 | 0.03 | 0.72 | 0.00 | 0.03 | 0.63 | 0.00 | 0.03 | 0.70 | 0.00 | 0.03 | 0.64 |
| 10 | 0.67 | 0.37 | 0.63 | 0.73 | 0.36 | 0.81 | 0.87 | 0.33 | 0.90 | 0.67 | 0.37 | 0.75 | 0.73 | 0.36 | 0.75 |
| 11 | 0.33 | 0.55 | 0.40 | 0.33 | 0.55 | 0.55 | 0.67 | 0.48 | 0.49 | 0.67 | 0.48 | 0.59 | 0.33 | 0.55 | 0.42 |
| 12 | 0.67 | 0.16 | 0.90 | 0.67 | 0.16 | 0.86 | 0.67 | 0.16 | 0.86 | 0.67 | 0.16 | 0.84 | 0.33 | 0.21 | 0.83 |
| 13 | 0.50 | 0.17 | 0.92 | 0.50 | 0.17 | 0.92 | 0.50 | 0.17 | 0.92 | 0.50 | 0.17 | 0.92 | 0.50 | 0.17 | 0.58 |
| 14 | 0.82 | 0.44 | 0.79 | 0.82 | 0.44 | 0.76 | 0.91 | 0.42 | 0.75 | 0.73 | 0.45 | 0.74 | 0.36 | 0.53 | 0.44 |
| 15 | 0.00 | 0.53 | 0.24 | 0.00 | 0.53 | 0.24 | 0.00 | 0.53 | 0.19 | 0.00 | 0.53 | 0.22 | 0.00 | 0.53 | 0.15 |
| 16 | 0.00 | 0.90 | 0.00 | 0.00 | 0.90 | 0.00 | 0.70 | 0.76 | 0.42 | 0.00 | 0.90 | 0.00 | 1.00 | 1.00 | 0.59 |
| 17 | 1.00 | 0.14 | 1.00 | 1.00 | 0.14 | 0.97 | 1.00 | 0.14 | 0.89 | 0.20 | 0.28 | 0.39 | 1.00 | 0.14 | 1.00 |
| 18 | 0.86 | 0.25 | 0.87 | 0.84 | 0.26 | 0.85 | 0.58 | 0.31 | 0.69 | 0.66 | 0.29 | 0.74 | 1.00 | 1.00 | 0.63 |
| 19 | 0.97 | 0.86 | 0.56 | 0.97 | 0.86 | 0.68 | 0.83 | 0.87 | 0.49 | 0.91 | 0.86 | 0.67 | 1.00 | 0.85 | 0.68 |
| 20 | 0.82 | 0.74 | 0.64 | 0.76 | 0.75 | 0.63 | 0.80 | 0.74 | 0.61 | 0.80 | 0.74 | 0.62 | 0.79 | 0.74 | 0.47 |

experimented with various *K* and *N* for each dataset and select the best values of *K* and *N* that maximize the performance. Best performance is marked in blue in Table 4. The results in Table 4 indicate that the proposed algorithm significantly outperforms other existing methods for the majority of the datasets.

As mentioned above, different *N* may result in different TPR and FPR, but there is no effect on AUC. Brute force and Random walk just need *N* and do not need *K*, therefore, AUC of these two algorithms remains unchanged. To estimate the connectivity value of an instance, Random Walk views anomaly detection as performing a random walk on the Markov chain with a transition matrix. There are some correlations for all instances, but Markov property may not exist. Therefore, there is serious limitation in Random Walk. SFO, *K*-distance and Brute Force are distance-based methods. LOF is density-based, in some sense, it is based on the distance. Distance-based methods are simple and intelligible, but the performance is not stable, especially for sparse data. The advantage of a local instead of a global view is detailed in LOF, which depends on how isolated the object is with respect to the surrounding neighborhood. However, if the densities of two different instances are similar, LOF is not effective. *K*-distance and Brute Force just consider one distance, that is the *Kth* largest distance to the nearest neighbor, therefore they are not thoughtful. Overall, the proposed algorithm in this paper is better than four competing algorithms.

### 5.3.3. *Time complexity*

One of the critical performance aspects of any anomaly detection algorithm is the speed. Creating real-time requirements for anomaly detection algorithms is challenging. The time complexities of our algorithm and four compared algorithms are $O(n^2)$, where $n$ is the number of instances. The experiments are conducted on an Intel i5-3470 PC with 8.00GB main memory under windows 8 and implemented in Visual studio 2008.

Total detection time of every algorithm for all 20 datasets is shown in Fig. 10. Brute force takes every possible subsequence, and only one anomaly can be found when the algorithm is called one time. Therefore, it is rather time-consuming. The local reachability densities of an object's *K*-distance neighborhoods need be obtained in LOF. A transition matrix is needed to form in Random walk. Runtime trendline of every algorithm for different dataset sizes is shown in Fig. 11. When the data size is smaller than 1000K, the difference of runtime is slight, not more than 1s. The difference of runtime becomes gradually significant from the data size of 2000K. Each of the five methods shows a sharp rise in the runtime at the dataset size of about 6000K. As shown in Figs. 10 and 11, we can see that the runtime of our algorithm and *K*-distance is the shortest. They are almost the same. According to analysis, our method can outperform other existing methods in both AUC value and detection time. It means that our algorithm is workable and much effective to detect anomalies in time series.

**Table 4**
Best performance comparison of several algorithms using 20 datasets.

| No. | SFO | K-distance | LOF | Brute Force | Random Walk |
|---|---|---|---|---|---|
| 1 | 0.99 | 0.98 | 0.98 | 0.98 | 0.78 |
| 2 | 0.81 | 0.79 | 0.67 | 0.79 | 0.75 |
| 3 | 1.00 | 1.00 | 0.08 | 1.00 | 0.50 |
| 4 | 1.00 | 1.00 | 1.00 | 0.85 | 1.00 |
| 5 | 0.64 | 0.64 | 0.64 | 0.88 | 0.69 |
| 6 | 0.92 | 0.90 | 0.89 | 0.71 | 0.89 |
| 7 | 0.93 | 0.98 | 0.98 | 0.84 | 0.98 |
| 8 | 0.71 | 0.69 | 0.65 | 0.26 | 0.71 |
| 9 | 0.48 | 0.72 | 0.68 | 0.70 | 0.64 |
| 10 | 0.65 | 0.81 | 0.90 | 0.75 | 0.75 |
| 11 | 0.45 | 0.55 | 0.53 | 0.59 | 0.42 |
| 12 | 0.90 | 0.86 | 0.86 | 0.84 | 0.83 |
| 13 | 0.94 | 0.92 | 0.92 | 0.92 | 0.58 |
| 14 | 0.81 | 0.76 | 0.79 | 0.74 | 0.44 |
| 15 | 0.28 | 0.24 | 0.20 | 0.22 | 0.15 |
| 16 | 0.00 | 0.00 | 0.42 | 0.00 | 0.59 |
| 17 | 1.00 | 1.00 | 1.00 | 0.39 | 1.00 |
| 18 | 0.89 | 0.87 | 0.69 | 0.74 | 0.63 |
| 19 | 0.64 | 0.92 | 0.94 | 0.67 | 0.68 |
| 20 | 0.64 | 0.63 | 0.61 | 0.62 | 0.47 |

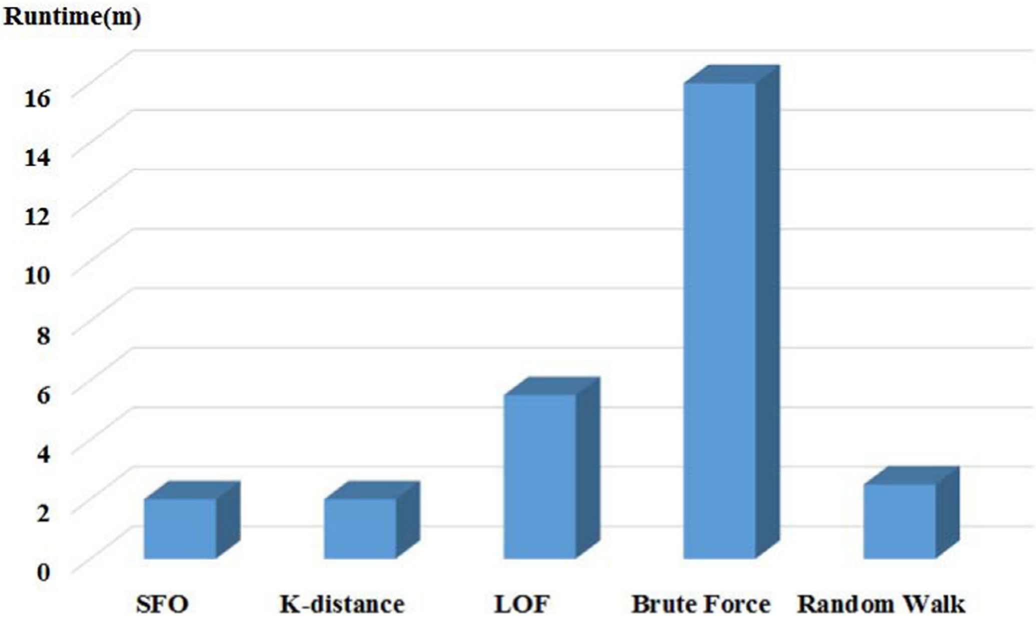

**Fig. 10.** Total runtime of every algorithm for all 20 datasets.

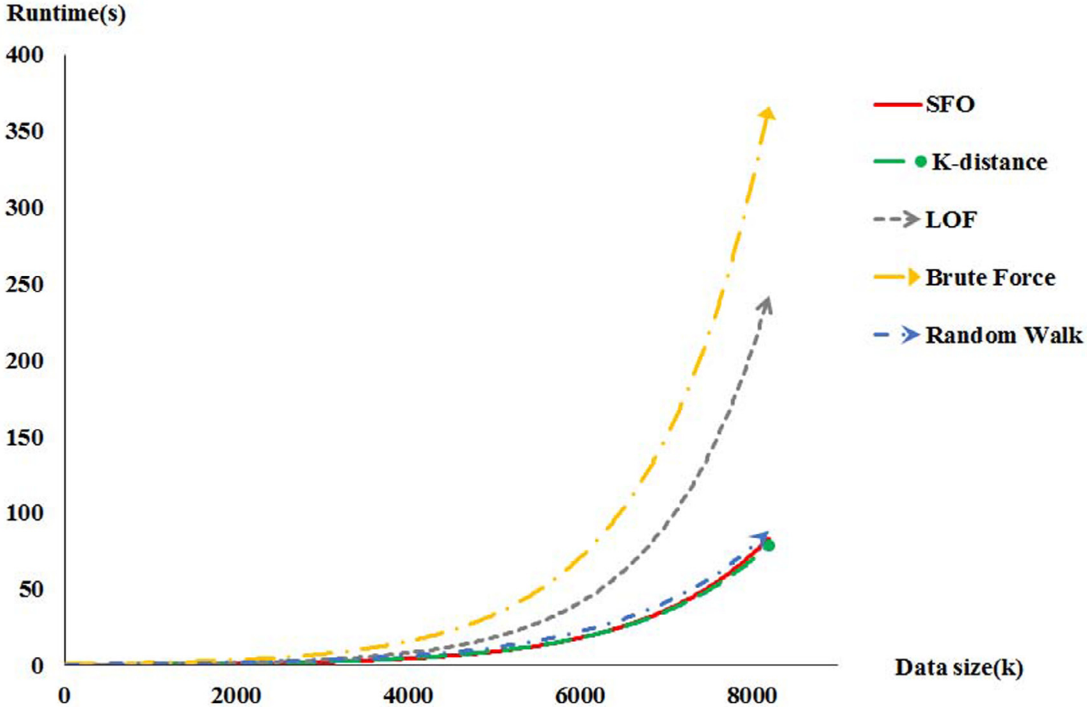

**Fig. 11.** Runtime trendline of every algorithm against the size of the dataset.

## 6. Conclusion

In this work, we have mainly developed a self-adaptive algorithm for finding anomalies in time series. A key feature of the algorithm is a synergistic combination of both statistical and fuzzy set-based theories. Exploiting fuzzy set theory in statistical process control, the detection is a distribution-free and unsupervised model. *K* optimization is necessary for AUC improvement. In this case, detection rate and false alarm rate almost are not affected by different *K*. False alarm rate has been significantly improved by intensive fuzzification process. Compared to other algorithms, the results indicate improved performance and detection time of the proposed algorithm.

Though the proposed algorithm performs well for anomaly detection in time series, it is distance-based method whose performance is not stable, especially for sparse data, meanwhile, the performance may be significantly affected by different proportion of anomaly data in datasets. These two issues are not considered in our work. In addition, the effort of anomaly detection in social network analysis is challenging, especially in the world of big data. In light of these findings, the future work will focus on sparse data, proportion of anomaly data and application in social network analysis.

## Acknowledgments

The work of this paper is funded by the project of National Natural Science Foundation of China (No.91520204) and the project of National High Technology Research and Development Program of China (863 Program) (No. 2015AA015405).

## References

Abadeh, M. S., Mohamadi, H., & Habibi, J. (2011). Design and analysis of genetic fuzzy systems for intrusion detection in computer networks. *Expert Systems with Applications, 38*(6), 7067–7075. doi:10.1016/j.eswa.2010.12.006.

Ahmed, M., Mahmood, A. N., & Islam, M. R. (2016). A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems, 55*, 278–288.

Ahmed, M., Mahmood, A. N., & Maher, M. J. (2015). Heart disease diagnosis using co-clustering. *Scalable information systems, Lecture notes of the institute for computer sciences, Social informatics and telecommunications engineering, 139*, 61–70.

Akouemo, H. N., & Povinelli, R. J. (2015). Probabilistic anomaly detection in natural gas time series data. *International Journal of Forecasting, 12*.

Albertetti, F., Grossrieder, L., Ribaux, O., & Stoffel, K. (2016). Change points detection in crime-related time series: an on-line fuzzy approach based on a shape space representation. *Applied Soft Computing, 40*, 441–454.

Appice, A., Guccione, P., Malerba, D., & Ciampi, A. (2014). Dealing with temporal and spatial correlations to classify outliers in geophysical data streams. *Information Sciences, 285*, 162–180.

Ashfaq, A. B., Rizvi, S., Javed, M., Khayam, S. A., Ali, M. Q., & Al-Shaer, E. (2014). Information theoretic feature space slicing for statistical anomaly detection. *Journal of Network and Computer Applications, 41*, 473–487.

Bondavalli, A., Ceccarelli, A., Brancati, F., Santoro, D., & Vadursi, M. (2016). Differential analysis of operating system indicators for anomaly detection in dependable systems: an experimental study. *Measurement, 80*, 229–240.

Brighenti, C., & Sanz-Bobi, M. (2011). Auto-regressive processes explained by self-organized maps: application to the detection of abnormal behavior in industrial processes. *IEEE Transactions on Neural Networks, 22*(12), 2078–2090. doi:10.1109/TNN.2011.2169810.

Ceclio, I. M., Ottewill, J. R., Pretlove, J., & Thornhill, N. F. (2014). Nearest neighbors method for detecting transient disturbances in process and electromechanical systems. *Journal of Process Control, 24*(9), 1382–1393.

Chen Y., Keogh E., Hu B., Begum N., Bagnall A., Mueen A., Batista G., 2015. The UCR time series classification archive, URL http://www.cs.575ucr.edu/~eamonn/time_series_data/.

Dangelo, G., Palmieri, F., Ficco, M., & Rampone, S. (2015). An uncertainty-managing batch relevance-based approach to network anomaly detection. *Applied Soft Computing, 36*, 408–418.

Derrac, J., Garca, S., & Herrera, F. (2014). Fuzzy nearest neighbor algorithms: taxonomy, experimental analysis and prospects. *Information Sciences, 260*, 98–119.

Drugman, T. (2014). Using mutual information in supervised temporal event detection: application to cough detection. *Biomedical Signal Processing and Control, 10*, 50–57.

Federico, S. W., Ignacio, A.-P. J., de-la Higuera Pablo, C., Marcos, M.-F., Dimitriadis, I. A., & Carlos, A.-L. (2011). Anomaly detection in network traffic based on statistical inference and $\alpha-$ stable modeling. *IEEE Transactions on Dependable and Secure Computing, 8*(4), 494–509.

Ha, J., Seok, S., & Lee, J. S. (2015). A precise ranking method for outlier detection. *Information Sciences, 324*, 88–107.

Harrou, F., Kadri, F., Chaabane, S., Tahon, C., & Sun, Y. (2015). Improved principal component analysis for anomaly detection: application to an emergency department. *Computers & Industrial Engineering, 88*, 63–77.

Huang, Y. P., Huang, C. Y., & Liu, S. I. (2014). Hybrid intelligent methods for arrhythmia detection and geriatric depression diagnosis. *Applied Soft Computing, 14, Part A*, 38–46.

Kadri, F., Harrou, F., Chaabane, S., Sun, Y., & Tahon, C. (2016). Seasonal ARMA-based spc charts for anomaly detection: application to emergency department systems. *Neurocomputing, 173*, 2102–2114.

Keogh E., 2005. The time series discords page. [online], URL http://www.cs.ucr.edu/~eamonn/discords/.

Koc, L., Mazzuchi, T. A., & Sarkani, S. (2012). A network intrusion detection system based on a hidden naïve bayes multiclass classifier. *Expert Systems with Applications, 39*(18), 13492–13500.

Koulaouzidis, G., Das, S., Cappiello, G., Mazomenos, E. B., Maharatna, K., Puddu, P. E., & Morgan, J. M. (2015). Prompt and accurate diagnosis of ventricular arrhythmias with a novel index based on phase space reconstruction of ECG. *International Journal of Cardiology, 182*, 38–43.

Kumarage, H., Khalil, I., Tari, Z., & Zomaya, A. (2013). Distributed anomaly detection for industrial wireless sensor networks based on fuzzy data modelling. *Journal of Parallel and Distributed Computing, 73*(6), 790–806. doi:10.1016/j.jpdc.2013.02.004.

Lee, J., Kang, B., & Kang, S. H. (2011a). Integrating independent component analysis and local outlier factor for plant-wide process monitoring. *Journal of Process Control, 21*(7), 1011–1021.

Lee, S., Kim, G., & Kim, S. (2011b). Self-adaptive and dynamic clustering for online anomaly detection. *Expert Systems with Applications, 38*(12), 14891–14898.

Lemos, A., Caminhas, W., & Gomide, F. (2013). Adaptive fault detection and diagnosis using an evolving fuzzy classifier. *Information Sciences, 220*, 64–85. doi:10.1016/j.ins.2011.08.030.

Li, Z., Xiong, H., & Liu, Y. (2012). Mining blackhole and volcano patterns in directed graphs: a general approach. *Data Mining and Knowledge Discovery, 25*(3), 577–602.

Lin, W. C., Ke, S. W., & Tsai, C. F. (2015). CANN: an intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-Based Systems, 78*, 13–21.

Moonesinghe, H., & Tan, P. (2008). Outrank: A graph-based outlier detection framework using random walks. *International Journal on Artificial Intelligence Tools, 17*(1), 19–36.

Nurunnabi, A., West, G., & Belton, D. (2015). Outlier detection and robust normal-curvature estimation in mobile laser scanning 3d point cloud data. *Pattern Recognition, 48*(4), 1404–1419.

Oprea, R. C., & Aben, E. (2012). Traffic anomaly detection using a distributed measurement network. *Research Project, System and Network Engineering, University of Amsterdam*, 1–29.

Pfund, D., Anderson, K., Detwiler, R., Jarman, K., McDonald, B., Milbrath, B., … Woodring, M. (2016). Improvements in the method of radiation anomaly detection by spectral comparison ratios. *Applied Radiation and Isotopes, 110*, 174–182.

Pierazzi, F., Casolari, S., Colajanni, M., & Marchetti, M. (2016). Exploratory security analytics for anomaly detection. *Computers & Security, 56*, 28–49.

Quinn, J. A., & Sugiyama, M. (2014). A least-squares approach to anomaly detection in static and sequential data. *Pattern Recognition Letters, 40*, 36–40.

Sajjad, S. M., Bouk, S. H., & Yousaf, M. (2015). Neighbor node trust based intrusion detection system for wsn. *Procedia Computer Science, 63*, 183–188.

Shuchita, U., & Karanjit, S. (2012). Nearest neighbour based outlier detection techniques. *International Journal of Computer Trends and Technology, 3*(2), 299–303.

Spiric, J. V., Docic, M. B., & Stankovic, S. S. (2015). Fraud detection in registered electricity time series. *International Journal of Electrical Power & Energy Systems, 71*, 42–50.

Wang, Y., Xiang, J., Markert, R., & Liang, M. (2016). Spectral kurtosis for fault detection, diagnosis and prognostics of rotating machines: a review with applications. *Mechanical Systems and Signal Processing, 66-67*, 679–698.

Zhang, Y., Lu, H., Zhang, L., & Ruan, X. (2016). Combining motion and appearance cues for anomaly detection. *Pattern Recognition, 51*, 443–452.