EAI Endorsed Transactions
on Security and Safety

Research Article **EAI.EU**

Exploration of Singular Spectrum Analysis for Online Anomaly Detection in CRNs

Qi Dong[1], Zekun Yang[1], Yu Chen[1], Xiaohua Li[1], Kai Zeng[2]

[1]Dept. of Electrical and Computer Engineering, Binghamton University, Binghamton, NY 13902
[2]Volgenau School of Engineering, George Mason University, Fairfax, VA 22030

Abstract

Cognitive radio networks (CRNs) have been recognized as a promising technology that allows secondary users (SUs) extensively explore spectrum resource usage efficiency, while not introducing interference to licensed users. Due to the unregulated wireless network environment, CRNs are susceptible to various malicious entities. Thus, it is critical to detect anomalies in the first place. However, from the perspective of intrinsic features of CRNs, there is hardly in existence of an universal applicable anomaly detection scheme. Singular Spectrum Analysis (SSA) has been theoretically proven an optimal approach for accurate and quick detection of changes in the characteristics of a running (random) process. In addition, SSA is a model-free method and no parametric models have to be assumed for different types of anomalies, which makes it a universal anomaly detection scheme. In this paper, we introduce an adaptive parameter and component selection mechanism based on coherence for basic SSA method, upon which we built up a sliding window online anomaly detector in CRNs. Our experimental results indicate great accuracy of the SSA-based anomaly detector for multiple anomalies.

Keywords: Cognitive Radio Networks (CRNs), Anomaly Detection, Singular Spectrum Analysis (SSA).

Received on 21 December 2017; accepted on 25 December 2017; published on 28 December 2017

doi:10.4108/eai.28-12-2017.153516

1. Introduction

The rigid spectrum allocation scheme regulated by governmental agencies leads to great deficit on spectrum band resources. The emergence of new intelligent spectrum allocation/re-allocation scheme, especially cognitive radio networks (CRNs), is studied elaborately in the last decade, due to the ever-increasing wireless applications. CRN is a technology that allows wireless devices (unlicensed users) access spectrum resources dynamically without introducing major interference to licensed primary users. Because of the great difficulty and high complexity in dynamic spectrum access (DSA), and many open issues on security deployment, CRN study still stays in tentative phase and needs all-round improvements despite a large quantity of research studies. A well-designed CRN aims to serve for two purposes: to maximize the usage of spare spectrum resource as well as to protect the incumbent primary system from secondary network interference [1].

In order to meet the two requirements, and according to Federal Communications Commission (FCC): *"no modification to the incumbent signal should be required to accommodate opportunistic use of the spectrum by Secondary Users (SUs)"* [2], CRN system is expected to collect and process sufficient, highly accurate information of the spectrum environment, without imposing overhead on incumbent users by adding new features, such as redundant symbolic pads, or authentication protocols. Thus, this dynamic Big Data processing task is very challenging. In addition, due to the uncertainties in PU behaviours and unavoidable interloper including many malicious entities, it is, if not impossible, extremely hard to maintain a stable cognitive radio system. Existence of anomaly behaviours, which include traditional anomaly security threats and newly emerged CRN-specific security threats, such as jamming, primary user emulation (PUE) attacks, spectrum sensing data falsification (SSDF), common control channel jamming, selfish users, intruding nodes and more [3], make the situation even worse.

In general, "anomaly" refers to unexpected or unnormal situations. In CRNs, specifically, "anomaly situation" can be introduced by various malicious activities as well as by versatile unpredictable PU activities, both of which can cause degradation on link quality of CRNs. Sometimes, it is necessary to detect those anomaly situations without introducing much overhead to cognitive entities. This task belongs to change-point detection, which concerns the design and analysis of procedures for "on-the-go" detection of possible changes in the characteristics of a running (random) process. Specifically, the process is assumed to be continuously monitored through sequentially made observations (e.g., measurements), whose behaviors should suggest the process may have statistically changed. The aim is to conclude so within the fewest observations possible, subject to a tolerable level of the risk of false detection [4], [5]. The time instance at which the state of the process changes is referred to as the change-point, which is not known in advance.

The Singular Spectrum Analysis (SSA) theory is viewed as an appropriate solution for this quickest change-point detection issue. However, SSA is mostly used as a postmortem analysis technique and operates blocks of data, which makes SSA rarely be adopted in real world engineering applications. The main idea of SSA is to decompose the specific time series into additive components such as trends, oscillation and "structureless" noise [6]. In a typical situation, after employing SSA, the operator need to select the interested component for further analysis. Therefore it is difficult to build a real-time processing based on SSA, only if an associated real-time automatic parameter and component selection mechanism is involved. However, the SSA methods do not need either a parametric model or stationarity-type conditions of the operating data, which makes it a model-free method and applicable to different categories of CRN anomalies.

In this work, we explored to leverage the attractive capabilities of the SSA scheme to solve the online anomaly detection problem in CRNs. The key contribution lies in an adaptive parameter and component selection mechanism that enables the SSA cope with the complexity in anomaly detection in CRNs. Through extensive experimental study, we have also investigated the tradeoffs between the sensitivity to subtle changes and the detection performance. For the convenience of discussion, we assume an ON/OFF Markov PU activity model and PUE attacks anomaly model.

The rest of this paper is organized as the following: Section 2 provides background information and briefly discusses some most closely related work. Section 3 describes the system model with interference to secondary user (SU) activities in detail. Section 4 explores an advanced study of SSA on CRNs anomaly detection. Then Section 5 reports our experimental results on two different anomaly detections, PUE attack and PU activity anomaly, respectively. Finally, Section 6 concludes this paper.

2. Background and Related Work

To date, significant amount of work has been presented to address various CRN security issues. For instance, the counter-measures to SSDF attacks include deploying a reputation metric to denote the scale of trustworthiness of each user [7], or reporting continuous sensing result to minimal attacks [8]. Game theory and Q-learning algorithm are often utilized to discuss attacker-SU action patterns [9] and to detect selfish users [10]. In confront of common control channel jamming attack, traditional communication technique of channel hopping is proved efficient for SUs exchanging channel information via multiple common control channels stochastically [11]. From the perspective of intrinsic features of CRNs, to the best of our knowledge, no universal anomaly detection scheme has been proposed.

SSA is introduced by Broomhead and King for the first time in 1986 [12]. Since then, it has shown its significant ability in a wide field of time series processing, such as finding data structure, extracting periodic pattern and complex trends, smoothing and change point detection [13],[14]. Wu *et al.* applied SSA for data preprocessing, associating with artificial neural network (ANN), to predict daily rainfall-runoff transformation [15]. Oropeza and Sacchi presented multichannel singular spectrum analysis (MSSA) as a tool for simultaneously denoising and reconstructing seismic data [16]. Moskvina and Zhigljavsky developed an algorithm of change-point detection in time series, based on sequential application of SSA [17].

A PUE attack is that malicious entities mimic PU signals in order to either occupy spectrum resource selfishly or conduct Denial of Service (DoS) attacks. PUE attacks can be easily implemented in CRNs and introduce great overhead on cognitive radio communication and cause chaos in dynamic spectrum sensing. Many detection techniques are based on geometrical information of the PU transmitter. In [18], the authors claimed that in order to achieve a better attack result, attackers intend to adjust transmission power according to PU activities. By implementing a transmitter parameter detection method that can measure the received power at SUs, it achieved a good detection result. However, it requires priori knowledge of distances among the nodes in the wireless environment. On the other hand, it is not accurate to identify PU signal by measuring the received signal strength (RSS). Not only because signal strength may vary by a large magnitude over small area, but also the attackers can constantly change the transmitter position and transmission power to disguise themselves. Thus, researchers have tried to eliminate noisy from received signals by constructing a sensor network [19], which introduces overhead to CRNs.

An auxiliary detection method is proposed to against attacks including PUE attacks using both local and global database [20]. However, it did not consider many real world situations such as location variance or smart attackers. A physical layer authentication scheme was introduced by adding tags to PU signals to counter PUE attacks [21]. In this method, PUs leverage a hash function to generate a one-way hash chain composed by a series of irreversible hash values at the beginning of communication. Whenever a PU wants to transmit a signal, it tags a hash value from the hash chain in a reversed manner. Both SUs and attackers are able to receive the tag. SUs can identify PU identity by calculating the tag, because attackers cannot forge the tag as the hash chain is irreversible. Nevertheless, in addition to imposing authentication requirement to PUs, this method is only applicable to digital communication systems with quadrature amplitude modulation (QAM) system.

Comparing to all discussed PUE attack counter-measures, in our proposal, we do not aim at sifting adversaries' signal from PU signal. Instead, our method is able to detect abnormal activities without acquiring any priori information of PUs or malicious entities, and neither is there any overhead to PUs. For further exploration of our detection method, a rudimentary PU activity anomaly detection is proved feasible in our work, which is useful on prediction of PU activity and monitor PU network health status.

3. System Model

Let us consider a typical centralized CRN. Due to the opportunistic nature of cognitive radio spectrum access methods and intricate wireless channel traffic model, a method for detecting abnormal activities in CRNs is not always universally applicable to all situations. In an environment of non-deterministic PU traffic pattern, it is difficult to precisely predict channel idle periods. In a distributed CRN, all cognitive nodes share the spectrum resource with incumbent users, thus a single cognitive node can hardly be aware of an anomaly at system level. Therefore, a smart attacker can take advantage of the nature that channel idle period could fluctuate dramatically, and disguise as the PU to occupy spectrum resource selfishly. In our system model, an online detection technique is designed to fit many wireless application scenarios regardless of channel status fluctuation, by simply inspecting cognitive nodes' activities.

3.1. Assumptions

We made the following assumptions for our model:

- The CRN consists of several cognitive nodes that can dynamically access spectrum resource, and a fusion center (FC) that collects cognitive nodes' data flow information, and detects abnormal activates;

- The total available spectrum resource is composed by multiple PU channels. The channels are independent to each other on traffic pattern. The traffic model for each channel is non-deterministic with fluctuation in both busy and idle periods;

- The spectrum resource is intensively used by CRNs in which the cognitive node stores the sensed channel state locally and transmit data when the channel is not used by other cognitive nodes, while a minimal transmission spectrum opportunity time slot T_{min_tx} is required;

- Every cognitive node will broadcast a packet P_{num} to FC after a transmission period T_{period} via an idle channel. P_{num} contains information about the number of received data packets in last T_{period}. The overhead to each cognitive node is minimal as only one extra packet is required for every T_{period}. To further guarantee the integrity and security of the transmitted P_{num}, some authentication and authorization techniques can be introduced;

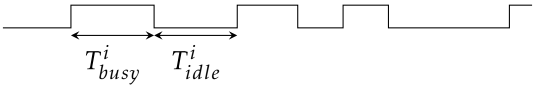

Figure 1. Channel model of idle & busy time.

- It is always feasible for each cognitive node to find an idle channel to broadcast P_{num}, because $T_{period} \gg T_{busy}$. (T_{period} is in the order of second, while T_{busy} is mostly in order of millisecond [22]);

- Attackers are smart enough to mimic PUs' signals, and they can conduct secret attack without introducing great fluctuation to the entire CRN;

- The FC can perform online anomaly detection based on the integrated statistics from all cognitive nodes.

3.2. Wireless Traffic Model and Analysis

Dynamic spectrum access (DSA) allows a cognitive radio to assign SUs some licensed bands temporarily, in an opportunistic and non-interfering manner. Therefore, the information about the spectrum occupancy patterns of the PUs is necessary. Because PU channels are independent to each other, it is feasible to analyze the model of an individual wireless channel. At a particular time point and a geographical location, a primary radio channel is either busy or idle. As illustrated in Fig. 1, T^i_{busy} denotes PU busy time slot on the i-th channel, which indicates there is PU activity exist in such channel. T^i_{idle} stands for PU idle time slot on i-th channel, which means no primary user is occupying this channel. Hence, the primary radio channel's spectrum occupancy pattern, which is also the PU's traffic pattern, describes the distribution of the durations of the busy and idle time slots.

Basically, there are two classes of traffic patterns in wireless environment: 1) Deterministic patterns, where the duration of either idle time (T_{idle}) or busy time (T_{busy}), if not both, is fixed; and 2) Stochastic patterns, where the start time and duration of both states are random and be modeled with statistical properties [23]. For the former category, the appearance of an attack will certainly change the deterministic patterns, making it easy to detect. For the later category, which is more likely in real world situation since the state exchanges randomly, it is hard to determine if the change of the occupancy pattern is related to an attack. Therefore, considering a continuous-time model, the channel remains in one state for a random period before switching to the other state. It is proven to be efficient that the PU activity

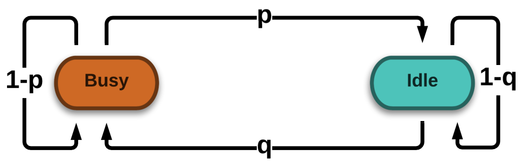

Figure 2. PU channel ON/OFF two-state Markov Renewal Model. p denotes the probability of state transition from 'busy' to 'idle'; q denotes the probability of state transition from 'idle' to 'busy'.

can be modeled as continuous-time, alternating ON/OFF two-state Markov Renewal Process (MRP) [24], as shown in Fig. 2.

In this model, both T^i_{idle} and T^i_{busy} can be regarded as independent and identically distributed (i.i.d.) processes, where the PU activity arrival follows a Poisson distribution. The continuous time span, i.e. T^i_{idle} and T^i_{busy}, follows exponential distribution, if PU activity arrival is a Poisson process [25]. The probability density function (PDF) of the distribution can be stated as:

$$f(T^i_{idle}) = \lambda_{idle} \cdot e^{-\lambda_{idle} T^i_{idle}} \tag{1}$$

$$f(T^i_{busy}) = \lambda_{busy} \cdot e^{-\lambda_{busy} T^i_{busy}} \tag{2}$$

where λ_{idle} and λ_{busy} are rate parameters, and their actual value is handy by stochastic measurement [26]. The mean value of idle $E[T^i_{idle}]$ and busy $E[T^i_{busy}]$ time can be calculated by:

$$\begin{aligned} E[T^i_{idle}] &= \int_0^\infty T^i_{idle} \cdot \lambda_{idle} \cdot e^{-\lambda_{idle} T^i_{idle}} \, \mathrm{d}T^i_{idle} \\ &= -(T^i_{idle} + 1/\lambda_{idle}) e^{-\lambda_{idle} T^i_{idle}} \Big|_0^\infty \\ &= 1/\lambda_{idle} && (3) \\ E[T^i_{busy}] &= 1/\lambda_{busy} && (4) \end{aligned}$$

The length of T^i_{idle} is critical for CRNs when exploiting the spectrum resource. A high time resolution of PU activity pattern may cause futile spectrum resource usage. For example, a successful packet transmission in CRNs requires a minimal transmission time span T_{min_tx}. In some intermittent PU activity channel, T^i_{idle} is usually too small for a complete transmission by SU, where $T^i_{idle} < T_{min\_tx}$. As shown in Fig. 3, a valid idle transmission slot requires $T^i_{idle} > T_{min\_tx}$. Thus, the available SU transmission time span can be calculated as:

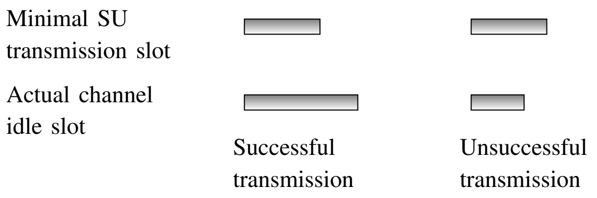

Figure 3. Effect of channel idle slot size on valid SU transmission.

$$\begin{aligned} T_{tx} &= \int_{T_{min\_tx}}^\infty T^i_{idle} \cdot \lambda_{idle} \cdot e^{-\lambda_{idle} T^i_{idle}} \, \mathrm{d}T^i_{idle} \\ &= -(T^i_{idle} + 1/\lambda_{idle}) e^{-\lambda_{idle} T^i_{idle}} \Big|_{T_{min\_tx}}^\infty \\ &= (T_{min\_tx} + 1/\lambda_{idle}) e^{-\lambda_{idle} T_{min\_tx}} && (5) \end{aligned}$$

With a stable PU transmission pattern, there is a corresponding stable spectrum resource pool for CRNs. Any anomaly behaviour introduced either by malicious entities or PU itself, will cause variation on rate parameters λ_{idle} and λ_{busy}, and afterwards the available SU transmission time span T_{tx}. In addition, a certain external symptom will show at FC as a change of the integrated packet flow P_{num}.

In this paper, we implement online anomaly detection based on the channel information P_{num} from all cognitive nodes, without knowing prior information λ_{idle} and λ_{busy}.

4. SSA based anomaly detection

When the channel fluctuation is not considered, it is straightforward to attribute a deterioration of CRN channel quality to an anomaly in certain category. From the viewpoint of the FC, an anomaly results in the decrease of the overall receiving packets rate of all the SUs. Therefore, SSA algorithm is introduced to detect the change of the overall packets rate.

4.1. Basic SSA Algorithm Description

The basic algorithm of SSA is described as following:

Assume $\mathbb{X} = (x_1, x_2, \cdots, x_N)$ is a real-value time series with the length of N. A sliding window, with a fixed window length of M, is adopted to truncate $\mathbb{X}$ and get a series of lagged vectors, and then, transform these vectors to a trajectory matrix X. The trajectory matrix X includes the whole information of the original time series $\mathbb{X}$.

The columns X_j of the trajectory matrix X can be considered as vectors in an M-dimensional space $\mathbf{R}_M$. A particular combination of a certain number l of the Singular Value Decomposition (SVD) eigenvectors determines an l-dimensional subspace $\mathcal{L}_l$ in $\mathbf{R}_M$, $l < M$. The M-dimensional data $X_1, \cdots, X_K$ is then projected onto the subspace $\mathcal{L}_l$.

In our system model, FC gets the packets rate in every T_{period} from each single SU, and calculates the whole network's packets rate $\mathbb{X} = (x_1, x_2, \cdots, x_N)$, which is the object of the SSA processing. The SSA processing of $\mathbb{X}$ includes the following steps.

Step 1: Embedding

Map the vector $\mathbb{X}$ into a $M \times K$ matrix X,

$$X = \left[\overrightarrow{X_1}, \overrightarrow{X_2}, \cdots, \overrightarrow{X_K} \right] = \left(x_{i,j} \right)_{i,j=1}^{M,K} \tag{6}$$

$$\overrightarrow{X_i} = (x_i, \cdots, x_{M+i-1})', i = 1, \cdots, K \tag{7}$$

where $K = N - M + 1$. The matrix X is called trajectory matrix and the vectors $\overrightarrow{X_i}$ are called lagged vectors. Note that X is a Hankel matrix, which has the equal elements on it's skew-diagonals $i + j = const$.

Step 2: Singular Value Decomposition

Apply SVD procedure on the trajectory matrix X and obtain M singular values $\sqrt{\lambda_1}, \sqrt{\lambda_2}, \cdots, \sqrt{\lambda_M}$ (in decreasing order) and the corresponding left singular vectors $U_1, U_2, \cdots, U_M$, and right singular vectors $V_1, V_2, \cdots, V_M$. The collection $(\sqrt{\lambda_i}, U_i, V_i)$, $i = 1, 2, \cdots, M$ is called the i-th eigentriple of the SVD. According to the standard SVD terminology, λ_i and U_i are the eigenvalues and eigenvectors of matrix $R = XX'$, respectively, while V_i are the eigenvectors of matrix $R' = X'X$, V_i are also called the principal components. The eigentriple satisfies $V_i = X'U_i/\sqrt{\lambda_i}$. Note that the rank of X is d, which is also the rank of R, then $\lambda_i = 0$, where $i > d$. So the trajectory matrix X will be:

$$X = X_1 + X_2 + \cdots + X_d \tag{8}$$

where $X_i = \sqrt{\lambda_i} U_i V_i'$ are rank-one biorthogonal matrices, $i = 1, \cdots, d$.

Step 3: Grouping

Select a subset indices I of $\{1, 2, \cdots, d\}$, with l elements

$$I = \{i_1, i_2, \cdots, i_l\} \tag{9}$$

such that

$$\bar{I} = \{1, 2, \cdots, d\}/I \tag{10}$$

Then the representation turns to

$$X = X_{i_1} + X_{i_2} + \cdots + X_{i_l} + X_{\bar{I}} \tag{11}$$

where $X_{\bar{I}} = \sum_{i \notin I} X_i$.

Step 4: Diagonal Averaging

Diagonal Averaging is used to transfer matrix $X_I = \sum_{i \in I} X_i$ into a time series (reconstruction). According to mathematic deduction, it is the component-sum of the original series $\mathbb{X}$.

$$x_i = \begin{cases} \frac{1}{i} \sum_{j=1}^{i} x_{j,i-j+1} & for\ 1 \leq i < M \\ \frac{1}{M} \sum_{j=1}^{M} x_{j,i-j+1} & for\ L \leq i \leq K \\ \frac{1}{N-i+1} \sum_{j=i-K+1}^{N-K+1} x_{j,i-j+1} & for\ K < i \leq N \end{cases} \tag{12}$$

That is, the reconstructed series element x_i equals to the average of the corresponding matrix elements sharing the same location with that x_i appears in the trajectory matrix X.

The main purpose of SSA is to decompose the original time series into several additive components, based on the assumption that this series is a sum of several simpler series. Specifically, a general descriptive model of the series that we use in SSA methodology is an additive model in which the components are trends, oscillations, and noise. Figure 4 shows a SSA decomposition of a section of CRN packet rate flow. The conception of "trend" denotes a components that is (i) not stationary and (ii) 'slowly varies' during the whole period of time that the series is being observed, as the first and second components shown in Fig. 4. Meanwhile, the 'oscillation' components can be divided into periodic and quasi-periodic, like the third and forth components shown in Fig. 4. Compared to them, the 'noise' component does not have a certain boundary with others. But generally speaking, 'noise' are aperiodic series, thus contribute less to the original series than others, like the fifth component.

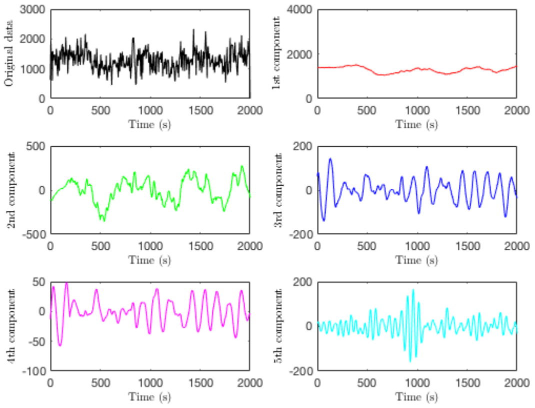

Figure 4. SSA decomposition example.

Figure 5 depicts the differences among the three kinds of components in the frequency domain. In general, the trend converge its energy at the low frequency region, as shown by the first and second components. The oscillation corresponds to a peak at particular frequency, and always occurs in pairs (the reason will be explained in next section), as shown by the forth and fifth component. Separating the whole series into these components and analyzing the linear homogeneous recurrence relations (LRRs) for interpretable components is helpful to obtain reliable and meaningful SSA results. In our case, only the general variation trend of channel traffic flow is considered, so that the gentle periodical fluctuation and random channel error are ignored.

4.2. On-line SSA

The basic SSA algorithm is useful to the determined (although stochastic) time series only, because it takes the

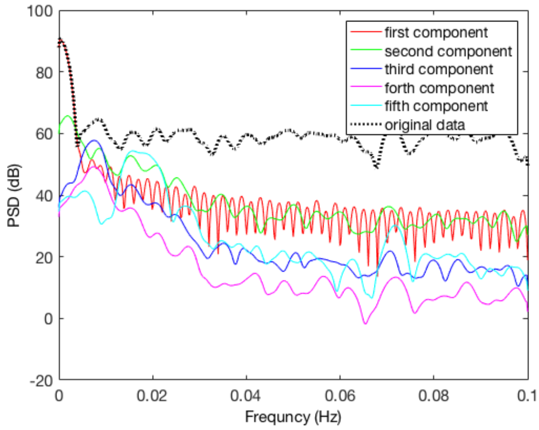

Figure 5. Frequency domain analysis example.

whole series as the object. A straightforward way to make SSA work for on-line, real-time processing is segment processing with a sliding window. But there are two issues. The first one is the window size, or the size of segmented time series prepared for SSA, is directly related to detection time resolution and it determines the trade-off between accuracy and delay. Meanwhile, the parameters of SSA should be adaptive to the dataset.

There are two parameters that are needed to be determined in basic SSA, the window length M and the group of I indices, which determine the subspace $\mathcal{L}_I$. According to the data complexity consistency, each segment's processing can share an uniform M, so it can be decided automatically at the start of the operation and keep till end, while the subspace indices are varying all the time.

While the example shown in Fig. 4 has two trend components, our experimental analysis revealed that many segments have only one trend component. In order to obtain the trend information of each segment, the grouping strategy is expected to be adaptive. Otherwise, using fixed number of components will either fail to obtain the complete trend information of some segments, or contaminate the trend information of some segments by unnecessary components. Therefore, the grouping strategy should ensure only and all the trend components are selected for reconstruction. To illustrate the adaptive grouping strategy, an indicator called separability is used.

Because SSA transforms a time series into a sum of different additive components, the separability of additive components plays a fundamental role. The SVD procedure provides a weak separability. Consider a fixed window length M and the grouping step can be defined in terms of multivariate geometry. Denoting each X_i the M-trajectory matrices of the series $\mathbb{X}_i$ and the grouping result $\mathbb{X} = \mathbb{X}_1 + \mathbb{X}_2 + \cdots + \mathbb{X}_d$. Since X_i are biorthogonal matrices, the corresponding $\mathbb{X}_i$ also inherit the independence. However, it does not necessarily mean they are separable and have no interaction with each other. Because X_i are rank-one matrices, all interpretable components (components without noise) can be often approximated by time series of small rank: a typical slow-varying trend in rank one and a sinusoid in rank two. A noisy component is too complex to be described with limited ranks.

Therefore, the trends show a clear separability, and an oscillation component occupies two successive $\mathbb{X}_i$ with strong interaction to each other but are separable to other components. When the eigentriple is arranged in the order descending of eigenvalues, the symmetrical pair of $\mathbb{X}_i$ have similar eigenvalues, so they are always neighbors. In this paper, we use the w-correlation [27] to quantitatively measure the separability.

Characteristics of Separability: w-correlation

Assume $\mathbb{X} = \mathbb{X}^{(1)} + \mathbb{X}^{(2)}$, and the degree of approximate separability between two series $\mathbb{X}^{(1)}$ and $\mathbb{X}^{(2)}$ need to be measured. Let's introduce the weights:

$$w_i = \begin{cases} i & for\ 1 \leq i < M \\ M & for\ M \leq i \leq K \\ N - i + 1 & for\ K < i \leq N \end{cases} \tag{13}$$

and define the inner product of series $\mathbb{X}^{(1)}$ and $\mathbb{X}^{(2)}$ of length N as:

$$(\mathbb{X}^{(1)}, \mathbb{X}^{(2)})_w = \sum_{i=1}^{N} w_i x_i^{(1)} x_i^{(2)} \tag{14}$$

If $(\mathbb{X}^{(1)}, \mathbb{X}^{(2)})_w = 0$, $\mathbb{X}^{(1)}$ and $\mathbb{X}^{(2)}$ are w-orthogonal. Meanwhile, separability implies w-orthogonality. Define the notion of w-correlation as follow:

$$\rho^{(w)}(\mathbb{X}^{(1)}, \mathbb{X}^{(2)}) = \frac{(\mathbb{X}^{(1)}, \mathbb{X}^{(2)})_w}{|\mathbb{X}^{(1)}|_w |\mathbb{X}^{(2)}|_w} \tag{15}$$

The w-correlation measures the separability between elementary reconstructed components $(\mathbb{X}_1, \mathbb{X}_2, \cdots, \mathbb{X}_d)$, and a w-correlation matrix can be generated. Take the example shown in Fig. 4, Fig. 6 presents the gray scale of the w-correlation (M=40). The first two eigentriples represent two trend elementary components, and the third and forth ones correspond to a harmonic. Since the last components either contribute little to the whole series or interact a lot with each other, they are considered as noise. It can be seen from the components' PDF. Therefore, the w-correlation is used to pick up trend components from all components. Typically, trend components contribute significantly to data structure, which means they are the first several components. In addition, the trend components are independent with other components, especially with its neighbors, which means its w-correlation values with neighbors are low.

4.3. Online Anomaly Detection Strategy in CRNs

Based on our assumptions, the anomalies in any categories will be detected by leveraging the CRN packet rate time-series $\mathbb{X} = \{x_1, x_2, x_3 \cdots\}$. In order to detect anomalies in

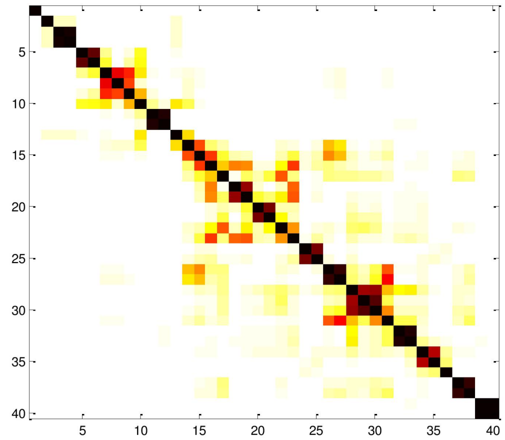

Figure 6. Matrix of w-correlation example.

time, we process the time-series in segments. Basically, the network state at a particular time point can be described by the SSA-rebuilt series around it, and it will be compared with the network state of previous time point. If the distance between these two network states is large enough, a change or anomaly is detected, which potentially indicate an attack. The detailed operation steps are as following.

Step 1: Window Extraction

$\mathbb{X} = \left\{\cdots, x_{p+1}, x_{p+2}, \cdots, x_{p+q}\right\}$ is a given CRN packet flow, where x_{p+q} is the currently arrived data. To evaluate the network at the time point $p+q$, a sliding window with length N truncates the original series twice in succession with an interval s. Fig. 7 shows two segments are obtained, the base window and the test window. Note that the interval s is fixed when the window moves, and $q = s + N - 1$.

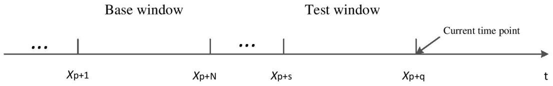

Figure 7. Sliding window.

Step 2: SSA Trend Extraction

Then, embed these two segments into two trajectory matrices in size $M \times K$, where $N = M + K - 1$. As mentioned before, by increasing the matrices size parameter M, the original data will be described in more details but also computing complexity will get increased. We take a tradeoff between accuracy and complexity, making M the floor of $q/6$. The base matrix is,

$$X_{base} = \begin{pmatrix} x_{p+1} & x_{p+2} & \cdots & x_{p+K} \\ x_{p+2} & x_{p+3} & \cdots & x_{p+K+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{p+M} & x_{p+M+1} & \cdots & x_{p+N} \end{pmatrix} \tag{16}$$

The test matrix is,

$$X_{test} = \begin{pmatrix} x_{p+s} & x_{p+s+1} & \cdots & x_{p+s+K-1} \\ x_{p+s+1} & x_{p+s+2} & \cdots & x_{p+s+K} \\ \vdots & \vdots & \ddots & \vdots \\ x_{p+s+M-1} & x_{p+s+M} & \cdots & x_{p+q} \end{pmatrix} \tag{17}$$

Then, SSA procedure is applied to the base and test matrices, respectively. Because only the trend components are needed to reconstruct the original series, w-correlation method is employed for automatic component grouping. Set $\mathbb{Y}_{base}$ and $\mathbb{Y}_{test}$ as the rebuilt series:

$$\mathbb{Y}_{base} = \left\{y_{base}^{(1)}, y_{base}^{(2)}, \cdots, y_{base}^{(N)}\right\} \tag{18}$$

$$\mathbb{Y}_{test} = \left\{y_{test}^{(1)}, y_{test}^{(2)}, \cdots, y_{test}^{(N)}\right\} \tag{19}$$

Step 3: Distance Computation and Anomaly Location

Once the base window and the test window are rebuilt by SSA, the difference of these two series with certain inverval can be evaluated using the Euclidean distance and the direction cosine as differential-based detector.

The Euclidean distance is calculated by:

$$\mathcal{D}_{test} = \sqrt{\sum_{i=1}^{N}(y_{test}^{(i)} - y_{base}^{(i)})^2} \tag{20}$$

If treat the two rebuilt time series y_{base} and y_{test} as two vectors in the subspace $\mathbf{R}_M$, then the angle θ can be generalized as below:

$$cos\theta = \frac{(\mathbb{Y}_{base}, \mathbb{Y}_{test})}{\|\mathbb{Y}_{base}\|\|\mathbb{Y}_{test}\|} = \frac{\sum_{i=1}^{N}(y_{test}^{(i)} y_{base}^{(i)})}{\sqrt{\sum_{i=1}^{N}(y_{base}^{(i)})^2}\sqrt{\sum_{i=1}^{N}(y_{test}^{(i)})^2}} \tag{21}$$

Anomaly will inevitably change the time series pattern, thus will be detectable because it shall enlarge window distance. Besides, other anomaly detection methods, which based only on feature of "test" window, such as data range, mean or variance of time series window, are worth trial. They are denoted as non-differential-based detection methods.

The non-differential-based detector are designed to detection fluctuation within a single sliding window; once an anomaly occurs, the primary features of a sliding window will present great fluctuation. The differential-based detector try to measure the distance between "test" and "base" sliding windows. Anomaly shall increase feature distance between a normal time series and an abnormal one.

5. experimental results

To evaluate the proposed CRN anomaly detection method, our experiment analysis is divided into two parts. In the first part, the basic SSA method is employed to deal with off-line

Table 1. Network Setting

Simulation time	$10000\ seconds\ for\ scenario\ I$
	$8000\ seconds\ for\ scenario\ II$
	$6500\ seconds\ for\ scenario\ III$
Number of PUs	5
Number of SUs	10
Number of tranmission channles	5
SU spectrum sensing duration	$10\ ms$
Transmission cycle of PUs	$40\ ms\ to\ 50\ ms\ (not\ fixed)$
Idle cycle of PUs	$50\ ms\ to\ 70\ ms\ (not\ fixed)$
Data packet size	$2000\ bytes$
SU data channel rate	$1.00\ Mbps$
SU transmission period T_{period}	$5\ second\ for\ scenario\ I$
	$4\ second\ for\ scenario\ II\&III$
Attacker hopping	Yes

anomaly detection problem, in order to discuss the capability of SSA on capturing the operating trend of random packet rate. In the second part, we promote our anomaly detector to online, in discussion of the real-time deployment of SSA based detection method.

A CRN is constructed using OMNET++ 4.6. Two categories of anomalies are considered in our study. One is PUE attacks and the other is sudden change of PU activities. In our implementation, SUs are randomly distributed in the environment and they can access to all channels in an opportunistic spectrum sensing manner. Each channel is legally allocated to a PU, whose duty cycle and idle cycle are stochastically distributed in a range. All SUs know neither geometry information of PUs, nor their broadcasting patterns.

Table 2. Off-line Detector Setting

SSA Setting	Widow length	$2000\ samples$
	Trajectory Matrix size M	$300\ samples$
	Grouping element I	$\{1, 2\}$
Differentiation Setting	Interval	$4\ samples$
Detector Setting	Threshold	15.12

5.1. Off-line Detection

1. System Setting

   Scenario Setting: This off-line anomaly detector is applied to a PUE attack specified environment as:

   Scenario I — a malicious party is a smart attacker that can start PUE attack at a randomly chosen time, and stop after accomplished sufficient attacks. For disguise purpose, the attacker hops among all spectrum channels randomly and implement secret PUE attacks. In this simulation, we applied 20 different seeds to construct 20 different CRN scenarios for anomaly detection. Among all scenarios, the attacker conducts PUE attack at 1500s, 5000s, 8000s respectively, and ends at 3000s, 7000s, 9300s respectively. Specific parameter setting is referred in Table 1.

   Detector Setting: An efficient detector requires a sufficiently long sliding window N in order to catch the principal system features, and the row size of trajectory matrix M should be restricted to $M < N/2$. In this off-line detection, since a long window is preferred, N can be pre-set as large as 2000 samples, and M can be pre-set as 300 samples. The detection threshold setting is critical in our detector. It can be generated by history observation data in normal non-attacking scenarios with proper guarding pad. Table 2 presents all detector parameters.

2. Result Analysis

   We present detection results with and without PUE attack respectively. In Fig. 9, the left-top subfigure shows an original packet flow synthesized by FC, which consists of oscillations and spikes. The right-top subfigure shows the data sequence reconstructed by the SSA process. Based on the features carried by principal components of the original data, the first and the second principal components, which are shown by the two subfigures in the bottom of Fig. 9, are selected for reconstruction due to the great oscillations in other components. The reconstructed data sequence shows clear changes around the time points: $1500s$, $3000s$, $5000s$, $7000s$, $8000s$, and $9300s$, when the PUE attacks started or ended. In contrast, Fig. 10 shows an original packet flow without PUE attack and the reconstructed data sequence from SSA. The reconstructed data indicates a smooth and steady CRN behavior. Besides the reconstruction process, our detection model contains a differential detector as shown in Fig. 8. Figures 11 and 12 show the outputs of the differential detector corresponding to cases with and without PUE attack respectively. While the detector output in Fig. 11 also shows spikes at those six anomaly points, the output in Fig. 12 presents only small fluctuations. Threshold configuration in differential detector is critical. Thus, a training session is required to find threshold properly.

   Based on our detection method, with respective history data training, our SSA based detection method achieved a detection rate of 84%, with false alarm rate of 8%, as shown in Table 3. However, we suffered an average delay of $45.96s$. The result of offline detection shows the appilcability of SSA based anomaly detection in CRN. In the next section,

Table 3. Off-line Simulation Result

Detection Rate	84%
False Alarm	8%
Average Delay	45.96s

an upgraded online anomaly detection experiment is illustrated with great details.

5.2. Online Detection

1. System Setting

   Scenario Setting: The configuration details of two anomaly situations are shown in Table 1. They are explained as following:

   a) In scenario II — a malicious party is a smart attacker that can start PUE attack at a randomly chosen time, and stop after accomplished sufficient attack. For disguise purpose, the attacker hops among all spectrum channels randomly and launches secret PUE attacks. In this experiment, we applied different seeds to construct 26 different CRN scenarios for anomaly detection. Among all scenarios, the attacker conducted PUE attacks at 5500s.

   b) In scenario III — during normal cases, the PUs will remain a versatile (not fixed, refer to table 1) ON/OFF pattern until they confront different network situations. In our experiment set-up, the PUs change their ON/OFF pattern randomly with 30 different seeds at 3500s, and recover to normal pattern at 5500s.

   Detector Setting: Although a large enough sliding window is preferred, we tentatively choose the sliding window length as 100 samples and trajectory matrix size as 16 samples, in order to reduce the computation overhead while maintaining the efficiency of SSA algorithm. Further, we applied four different detection methods: i). max-min range detection, ii). variance detection, which does not differentiate two consecutive windows, whereas to focus on the structure of current time series of signal, and iii). Euclidean distance detection and iv). cosine distance detection, which will differentiate two consecutive sliding windows. To achieve extensive of analysis, we test on a wide range of threshold based on small amount of training data.

2. Results and Analysis

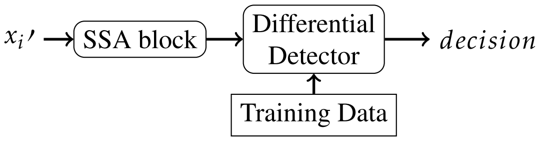

Figure 8. Offline Detector Structure.

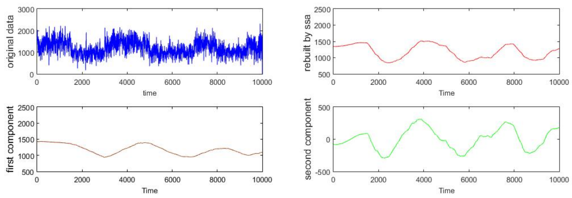

Figure 9. Original data flow, SSA reconstructed data flow, and the first and the second principal components with attack.

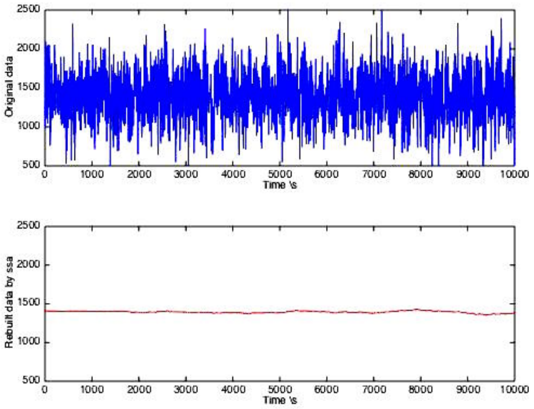

Figure 10. Original data flow and SSA reconstructed data flow without attack.

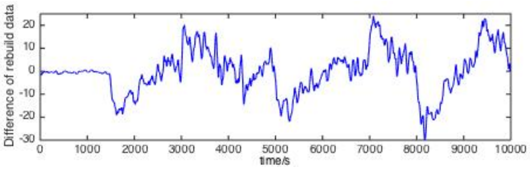

Figure 11. Differential detector result with attack.

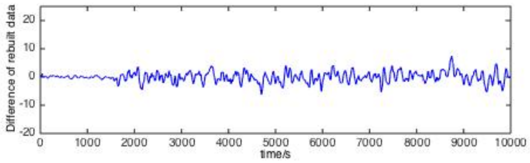

Figure 12. Differential detector result without attack.

We have tested 26 sets of CRN packet flow data for two scenarios, one for PUE attack and the other for PU anomaly, respectively. In our experiment, we mainly focused on testing hard-to-detect anomaly detections (deteriorate the network performance less than 30%), so the PUE attack and PU anomaly are subtle; note that the versatile primary user broadcasting pattern incurs even more difficulties. All four detection methods are applied to the proposed anomaly detector.

Table 4. Online Detector Setting

Sliding window length	100 $samples$
Trajectory Matrix size M	16 $samples$
Grouping conditions	The selected component i should satisfy: (1).$\lambda_i > 0.2\, sum\, of\ \lambda,\ and$ (2).$\rho_{i,i-1}^{(w)} < 0.6,\ and$ (3).$\rho_{i,i+1}^{(w)} < 0.6$

Figure 13 shows the detection result of scenario I, PUE attack. The attacker launched a PUE attack at 5500s. In the figure, all detection curves show spikes at data flow change point, i.e. next to 5500s. Otherwise, the detection curves remain stable when no anomaly appears. Figure 14 presents the detection result of scenario II, PU anomaly, in which detection spikes appear in all detection methods' curves, since PUs changed their communication pattern at 3500s and resumed the normal pattern at 5500s. The spikes indicate PU abnormal start and end points. Both figures show that our detector is sensitive to even subtle change of the original data flow.

However, more noticeable anomalies will result in more conspicuous detection results (deteriorate the network performance greater than 30%). Figure 15 shows the detection of a strong PUE attack and Fig. 16 shows the detection of a strong PU behavior anomaly. Both figure imply a good result of the SSA-based detector.

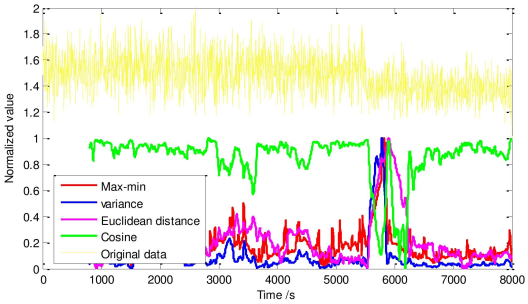

Figure 13. Detection example of weak PUE attack.

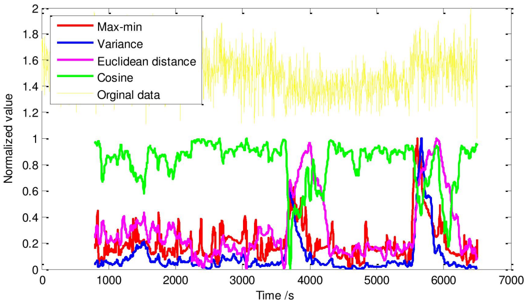

Figure 14. Detection example of subtle PU anomaly.

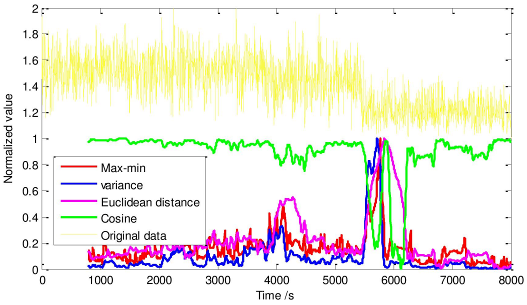

Figure 15. Detection example of strong PUE attack.

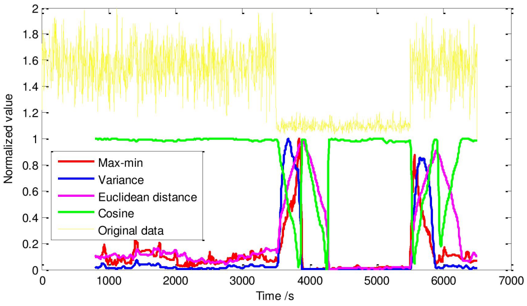

Figure 16. Detection example of clear PU anomaly.

In most of the experimental scenarios, our SSA based online detector is able to report the hard-to-detect anomaly correctly. But false negative and false positive were still observed when the anomalies are too subtle. Figure 17 shows an example of false negative that happened when the PUE attack is trivial. Similarly, Fig. 18 shows an example of false positive alert of PU anomaly. The false alarm and miss detection happen for two reasons: on the one hand, if CRN is suffering from radio environment fluctuation or adaptivity issues, a false alarm is somehow inevitable; on the other hand, if an anomaly is too subtle to cause harmful damage to CRN, a miss detection will get raised.

As illustrated by Fig. 17 and Fig. 18, The normal PU activity oscillation may change the packet flow as much as, sometimes even more than, an anomaly does. Therefore, although we can adjust the threshold to reduce the false alarm rate, it is not possible to eliminate the false alarms completely.

Obviously, the setting of thresholds is critical to reduce the false alarm rates while maintaining a desired detection accuracy. Since there is not a computable threshold for this method to distinguish abnormal and normal, we conduct empirical analysis on threshold selection based on the dataset of 26 series. Figures 19 and 20 shows different detector's receiver operating characteristic (ROC) curve in all PUE attack detection

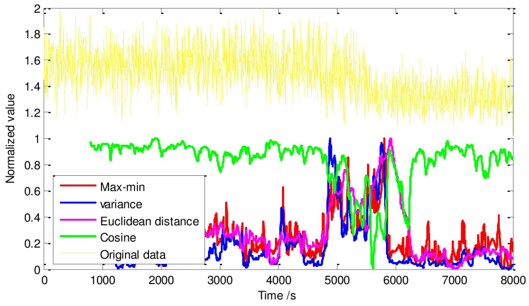

Figure 17. False positive in PUE attack detection.

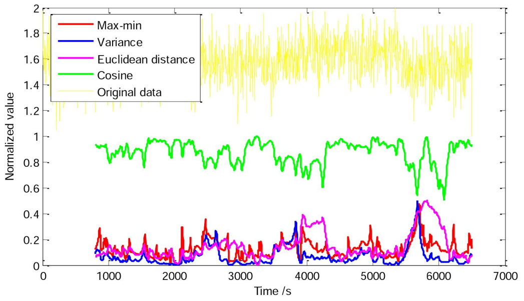

Figure 18. False Positive in PU anomaly detection.

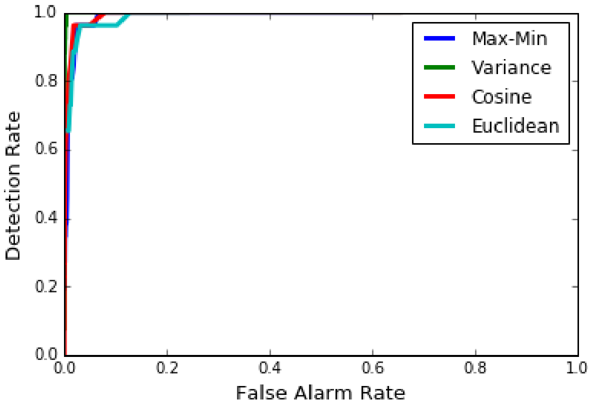

Figure 19. ROC curve of different detector on PUE attack.

and PU anomaly detection with tentative setting of different thresholds for different detection methods respectively. All the curve indicate good detection result, while non-differential-based methods, i.e. max-min range detection and variance detection show superior result, especially in simulated PUE attack detection scenarios. Figures 21 and 22 show the detection delay of different detection method in both scenarios. The detection delay vary in range of nearly 0s to over 100s. With some special threshold setting, the detection delay can be kept as low as several seconds, however will incur great false alarms.

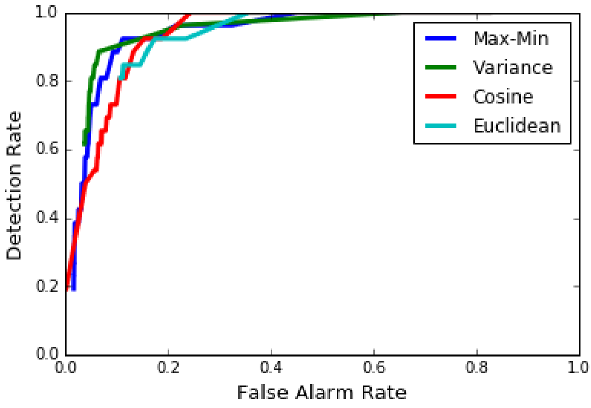

Figure 20. ROC curve of different detector on PU anomaly.

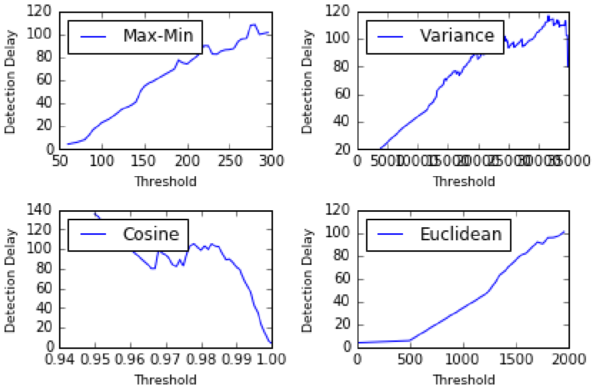

Figure 21. Detection delay of different detector on PUE attack (in seconds).

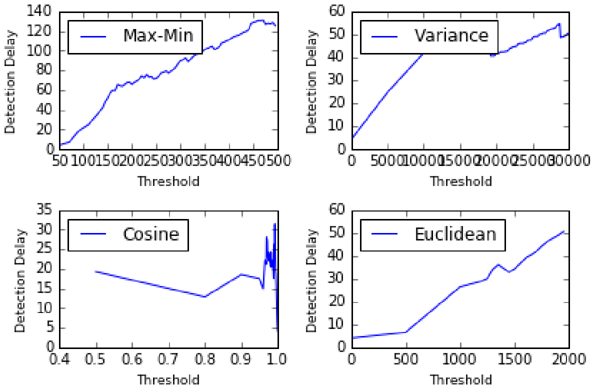

Figure 22. Detection delay of different detector on PU anomaly (in seconds).

Normally, a proper selection of threshold will generate reasonable detection rate and false alarm rate, while maintaining intermediate detection delay, as shown in Fig. 23 as an example of the relationship of

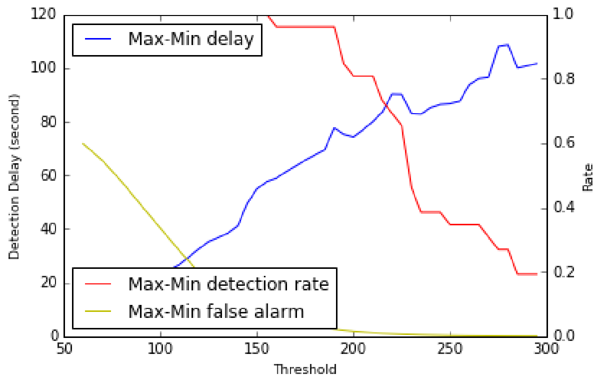

Figure 23. Detection example of max-min detector on PUE attack.

detection rate, false alarm rate, detection delay based on threshold setting of max-min detection method.

To summarize, the detection results indicate that our proposed online detection method can achieve a high detection rate while keeping false alarm rate low, in both anomaly categories. However, all detectors are suffering a relatively long detection delay.

6. Discussions & Conclusions

6.1. Discussions

From the experiment result, both of our SSA based off-line and online anomaly detection method show high detection rate and low false alarm rate, even when encountered with subtle PUE attack. Although the simulation environment only considered two anomalies: PUE attack and PU abnormality, our proposed method can be used for many other anomaly detections, such as spectrum sensing data falsification (SSDF) and jamming, because those anomaly activities will inevitably deteriorate communication condition of CRNs. On the other hand, however, our proposed detection method suffers from relatively high detection delay. The delay is generally caused by two factors: 1) our light-weighted anomaly detector requires SUs' data aggregation at FC for every several seconds (4s in our simulated online detection scenario); this relative low report time resolution will bring a small amount of detection delay; 2) our detection method is built upon transportation-layer-based data flow, which is not very sensitive to physical layer anomaly activities (most common in CRNs).

In comparison to traditional attack-specified security insurance techniques, our SSA based online anomaly detection method is applicable to most anomaly environment in CRNs. However, this method is no more than a detector. Due to its lightweight and ease of use, this method can be used as a rudimentary CRN network monitor and anomaly detector; once anomaly is detected, further action is needed to eliminate the anomaly.

To summarize, SSA based online detection method is not elixir; it is best used as an auxiliary general CRN anomaly detector, however, needs further measurements to resolve abnormalities.

6.2. Conclusions

Singular Spectrum Analysis (SSA) method possesses attractive features, such as parametric model free that enables it to detect different categories of anomalies. However, the basic SSA algorithm is suitable for off-line batch data processing. We explored to extend SSA to handle online anomaly detection problem in CRNs. An adaptive parameter and component selection mechanism is introduced to enable real-time operation. Taking PUE attacks and PU anomaly as case studies, our experimental results verified the effectiveness of the extended SSA detector. While this algorithm can detect subtle changes with high accuracy, it suffers relatively long delays. Our on-going effort focuses on the delay issue and aiming at an efficient anomaly detection scheme with both satisfactory accuracy and acceptable delays.

Acknowledgement

Q. Dong, Y. Chen and X. Li are supported by NSF via grant CNS-1443885. K. Zeng is partially supported by the NSF under grant No. CNS-1502584 and CNS-1464487.

References

[1] Adelantado, F. and Verikoukis, C. (2013) Detection of malicious users in cognitive radio ad hoc networks: A non-parametric statistical approach. *Ad Hoc Networks* **11**(8): 2367–2380 1570–8705.

[2] Fragkiadakis, A.G., Tragos, E.Z. and Askoxylakis, I.G. (2013) A survey on security threats and detection techniques in cognitive radio networks. *Communications Surveys & Tutorials, IEEE* **15**(1): 428–445.

[3] Esch, J. (2012) A survey of security challenges in cognitive radio networks: Solutions and future research directions. *Proceedings of the IEEE* **12**(100): 3170–3171

[4] Basseville, M., Nikiforov, I.V. *et al.* (1993) *Detection of abrupt changes: theory and application*, **104** (Prentice Hall Englewood Cliffs).

[5] Poor, H.V. and Hadjiliadis, O. (2009) *Quickest detection*, **40** (Cambridge University Press Cambridge).

[6] Golyandina, N. and Zhigljavsky, A. (2013) *Singular Spectrum Analysis for time series* (Springer Science & Business Media).

[7] Rawat, A.S., Anand, P., Chen, H. and Varshney, P.K. (2010) *Countering byzantine attacks in cognitive radio networks* (IEEE).

[8] Min, A.W., Shin, K.G. and Hu, X. (2009) *Attack-tolerant distributed sensing for dynamic spectrum access networks* (IEEE).

[9] Wang, B., Wu, Y., Liu, K.R. and Clancy, T.C. (2011) An anti-jamming stochastic game for cognitive radio networks. *IEEE Journal on Selected Areas in Communications* **29**(4): 877–889

[10] Attar, A., Nakhai, M.R. and Aghvami, A.H. (2009) Cognitive radio game for secondary spectrum access problem. *IEEE Transactions on Wireless communications* **8**(4): 2121–2131 1536–1276.

[11] Cormio, C. and Chowdhury, K.R. (2010) Common control channel design for cognitive radio wireless ad hoc networks using adaptive frequency hopping. *Ad Hoc Networks* **8**(4): 430–438

[12] Broomhead, D.S. and King, G.P. (1986) Extracting qualitative dynamics from experimental data. *Physica D: Nonlinear Phenomena* **20**(2-3): 217–236.

[13] Rukhin, A.L. (2002) Analysis of time series structure ssa and related techniques. *Technometrics* **44**(3): 290–290.

[14] Yang, Z., Zhou, N., Polunchenko, A. and Chen, Y. (2015) Quick online detection of start time of disturbance in power grid. In *the IEEE GlobeCom 2015, Selected Areas in Communications Symposium: Smart Grid Communications Track* (IEEE): 1–6.

[15] Wu, C. and Chau, K. (2011) Rainfall–runoff modeling using artificial neural network coupled with singular spectrum analysis. *Journal of Hydrology* **399**(3): 394–409.

[16] Oropeza, V. and Sacchi, M. (2011) Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis. *Geophysics* **76**(3): V25–V32.

[17] Moskvina, V. and Zhigljavsky, A. (2003) An algorithm based on singular spectrum analysis for change-point detection. *Communications in Statistics-Simulation and Computation* **32**(2): 319–352.

[18] Chen, Z., Cooklev, T., Chen, C. and Pomalaza-Ráez, C. (2009) *Modeling primary user emulation attacks and defenses in cognitive radio networks* (IEEE).

[19] Chen, R., Park, J.M. and Reed, J.H. (2008) Defense against primary user emulation attacks in cognitive radio networks. *Selected Areas in Communications, IEEE Journal on* **26**(1): 25–37

[20] Min, A.W., Kim, K.H. and Shin, K.G. (2011) *Robust cooperative sensing via state estimation in cognitive radio networks* (IEEE).

[21] Borle, K.M., Chen, B. and Du, W. (2013) *A physical layer authentication scheme for countering primary user emulation attack* (IEEE).

[22] Mahamuni, S. and Mishra, V. (2014) Performance evaluation of spectrum detection in cognitive radio network. *Int'l J. of Communications, Network and System Sciences* **7**(11): 485.

[23] Marko Hoyhtya, S.P. and Mammela, A. (2011) Improving the performance of cognitive radios through classification, learning, and predictive channel selection. In *ADVANCES IN ELECTRONICS AND TELECOMMUNICATIONS*.

[24] Rehmani, M.H., Viana, A.C., Khalife, H. and Fdida, S. (2013) Surf: A distributed channel selection strategy for data dissemination in multi-hop cognitive radio networks. *Computer Communications* **36**(10): 1172–1185

[25] Sriram, K. and Whitt, W. (1986) Characterizing superposition arrival processes in packet multiplexers for voice and data. *IEEE journal on selected areas in communications* **4**(6): 833–846 0733–8716.

[26] Kim, H. and Shin, K.G. (2008) Efficient discovery of spectrum opportunities with mac-layer sensing in cognitive radio networks. *IEEE transactions on mobile computing* **7**(5): 533–545

[27] Hassani, H. (2007) Singular spectrum analysis: methodology and comparison .