Application of Singular Spectrum Analysis to the Noise Reduction of Intrusion Detection Alarms

Jie Ma, Zhi Tang Li, Bing Bing Wang
Computer Science Department, Huazhong University of Science and Technology, Hubei Wuhan, China
mjhust@163.com

Abstract—Intrusion detection systems typically create a large volume of alarms and most of them are false alarms that can be seen as background noises caused by normal system behaviors. Manual analysis of a large number of alarms is both time consuming and labor intensive. This study focuses on the statistical analysis of the alarm flow. Using the Singular Spectrum Analysis (SSA) approach, we found that the alarm flow has a small intrinsic dimension, and the structure of alarm flow can be composed by leading components (normal components) and residual components (abnormal components). Only changes in abnormal components are worth of further study to confirm whether they are true or false alarm. To achieve this goal, an SSA-based anomalies detection algorithm was implemented and applied to catch anomalous changes in residua components, and thus interesting alarms were highlighted and noises were filtered out. Compared with detection approaches using stationary models, our SSA-based method can well deal with the non-stationary natures inherent in the alarm flow. Evaluation results from real network data show a significant increase in model accuracy, and more efficient filtering of alarm noise.

Index Terms—alarm noise, intrusion detection, SSA

I. INTRODUCTION

Because of the significant increase reliance on the Internet–based services, security and survivability of networks has become a primary concern. Intrusion Detection System (IDS) plays a vital role in the overall security infrastructure, as one last defense against attacks after secure network architecture design, secure program design and firewalls [1]. It gathers information form some key points in the computer networks, properly analyzes it and detect violations of the monitored system’s security policy, so as to extend the security management capability of the system administrators and improve the integrity of information security infrastructure.

It is estimated 90% of alarms generated by IDS are false alarms [2]. Since be overwhelmed by these false alarms, security administrators almost unable to recognize real attack in time. Indeed, a high rate of false alarms is considered to be the adverse factor for the performance of IDS. False alarms always cause an additional workload for security administrator, who must handle every single alarm to verify whether it is a true or false alarm. If this work is done manually, it will consume a lot of time and increase the probability of error. Therefore, we take for granted that reducing false alarms is a primary task for ensuring IDS efficiency and usability. In this paper we are concerned about dealing with false alarms caused by misuse intrusion detection systems, then the word IDS in context means signature-based IDS.

The most significant reason accounts for IDS false alarms is due to the lack of context knowledge of the working environment of the IDS. A simple and feasible method for reducing the false alarm rate can be done by optimizing detection rule sets of the IDS. As is known that some successful illegal invasions are performed by exploring vulnerabilities exist in a particular OS platform only. Then we can optimize the detection rule sets and adapt corresponding signatures to the specific environment and disabling the signatures that are not related to it. In practice the optimizing process is a trade-off between reducing false alarms and maintaining the security level. This often leaves administrators with the difficulty of determining a proper balance between an ideal detection rate and the possibility of having false alarms. Furthermore, optimizing requires a comprehensive review of the environment by experienced security administrator, and requires frequently updating to keep up with the flow of new vulnerabilities or threats discovered. Although this offers a good means of reducing the number of false alarms, the approach can not works well on processing high false alarm volumes caused by normal non-malicious background traffic. These alarms are not associated with any vulnerability, and consequently it fails to be verified from the context information of the monitored system. More seriously, these kinds of false alarms constitute the major part of the whole false positive alarms, and some true alarms caused by real attack events may be hidden in the midst of them. Besides, filtering these alarms only by some tuning techniques, such as optimizing the detection rules or disabling some signatures, may not only overburden the administrator’s work but also increase the risk of missing real attacks. In view of these facts, we prefer to take these kinds of false positive alarms as background noises, whose most notable feature is the ability to confuse the administrator’s Judgments and decision-making.

The final object of this work is to reduce alarm noises to improve alarm handling efficiency. We adopt that an alarm flow is composed by two different kinds of components that is normal and abnormal components.
Alarm noises are always related with the normal components of the flow. Only changes in abnormal components are worth considering for the administrator to see if anomalies rise. How to distinguish normal and abnormal components of alarm flow and execute anomalies detection are primary tasks for our noise reduction. The Main contributions of our work can be briefly described through the following points.

1) We processed IDS alarms using alarm flows which aggregate individual alarms along the timeline. This temporal and sequential handling of alarms enables the administrator to have a real-time view of security situation. The flow characterization can be beneficial in modeling alarm flow behaviors and addressing a variety of problems such as anomalies detection and alarm noise reduction.

2) Using the Singular Spectrum Analysis (SSA) approach, we found that the alarm flow has a small intrinsic dimension, and the structure of alarm flow can be profiled by two subsets of principal components, that is, the subset of leading components and the subset of residual components. We showed that the subset of leading components is responsible for the preservation of the normal behaviors of the original flow. Residual components capture the abnormal behaviors, which do not fit in the basic part of the alarm flow.

3) An SSA-based anomalies detection algorithm was implemented and applied to catch anomalous changes in residual components, and thus interesting alarms were highlighted and noises related to normal flow behaviors can be discarded and filtered. Experiments on real network data showed the efficiency of the approach.

The rest of the paper is organized as follows. Section 2 describes the problems we should solve and the motivation of our methods. In Section 3, we apply SSA to analyze the structures of alarm flows. Section 4 discusses the implementation details of the SSA-based anomalies detection algorithm. Section 5 evaluates the effectiveness of our approach through comparing the results with stationary AR models. We review the related work in Section 6 and conclude in Section 7.

II. PROBLEM AND MOTIVATION

In this section, we first give more details on our collection and observation of false positive alarms caused by background traffic, and then introduce the problem we need to solve, after which we discuss the motivation of our approach.

In practice, making comprehensive understanding of false positive alarms is a tough task. In order to make in-depth study of this issue, a series of experiments was conducted to analyze and evaluate IDS alarms generated by real network traffic. Snort IDS [6] was chosen as the main intrusion detector, which is a popular open-source Network Intrusion Detection System (NIDS). The reason for utilizing Snort was due to its openness and public availability. The data was accumulated from our campus network for one month by activating default rule sets of the Snort. Interestingly, more than 200 signatures had triggered alarms, and only five had generated 69% of the total number of false alarms. Fig.1 shows the top 5 alarms with their respective signatures.

![Fig.1 Top 5 False Alarms](image)

After in-depth exploration of these false alarms, it reveals that most of them were raised due to normal system operation or legitimate user behaviors, which merely occurred as a result of a network problem, not owing to the detection of real attacks. Take the ICMP traffic for instance. Reordering every connection associated with probing, such as all ping activities, will only produce a large number of false positive alarms. As discussed before, we refer these kinds of false positive alarms to alarm noises. These false alarms are often of low impact, and they does not need immediate reaction from the administrator, but they can affect administrators to discover the real attacks. Indeed, the unsuccessful attacks, or attempts that aim at an invincible target, might also cause the system to generate such noises.

It is impossible to make the difference between normal activities and attacks in the way they manifest themselves in alarm noises by focusing on the individual alarms. However, the distinction can be made by analyzing the aggregated alarm flow, which can be modeled as a time series model. The time series is a sequence of alarm intensity observations i.e. the number of alarms in a sampling interval as a time series. Only alarms generated by the same IDS and with the same signature can be aggregated to form alarm flow. The aggregated flow revealed not only the overall regularities but also the behaviors of malicious events, even though the significance of individual alarms was unclear.

Unusual changes in the intensity of the alarm flow can indicate a problem that is possibly security related. We have observed in our network environment rather significant regularities in alarm flows, having most likely non-malicious origins. Hence, it is desirable and possible to model these regularities, only changes in and deviations from the regularities are interesting for the administrator.

Our work builds most closely on the series of works by Viinikka et al. in [3]. The underlying assumption is that regular flow behavior is related to normal system use, or at least that only changes in any regular behavior are worth further investigation. Once the modeling is done, the output of the model corresponds to normal system behavior, and the difference between observed behavior and model output represents the abnormal components of the alarm flow. Using these abnormal components, we aim to detect any changes or deviations from the normal profile. These changes or deviations are regarded as anomalies in alarm flow which should be reported to
administrators, and other parts of the flow can be filtered out as noises. To capture normal regularities in alarm flows, three models were used in [3]. The first used exponentially weighted moving average (EWMA) model to capture short-term trends in flow behavior, it is the simplest and computationally lightest of the three. But, its modeling capacity is very limited which shows a inconsistent anomalies detection behavior in the presence of strong variations in the alert intensity. The second used stationary autoregressive (AR) model to capture the normal behaviors in alarm flows. The modeling capacity of the stationary AR model is slightly better but it needed to remove the trend and periodic components from the flow. Because of this components removal, the results are inconsistent and therefore difficult to interpret, and the risk of introducing artifacts into the flow may increase. Furthermore, both EWMA and AR models are unable to adapt to changes in normal behaviors, they are seemed to be more adaptable to working with some stable alarm flows. The last model used in [3] is a hybrid model created by combined use of non-stationary autoregressive model and Kalman smoother. The model is more accurate and consistent than stationary AR model and less dependent on the data set than with the EWMA model. The improvement in model accuracy is mainly provided by the adaptive parameter estimation with the Kalman smoother. However, the approach are more complex and prone to cost more time for modeling. In addition, this adaptive parameter estimation introduces a risk of incorporating unwanted behavior into the model.

To overcome the above mentioned problems, we use singular spectrum analysis (SSA) method [5]. SSA belongs to the general category of Principal Component Analysis (PCA) methods [6], which the original data space can be transformed into a feature space by using a linear transformation. And the data set may be represented by a reduced number of meaningful features while retaining most of the information content of the data. In this paper, we first employ SSA to explore the intrinsic dimensionality and structure of the time-series corresponding to alarm flow, using data collected from real network traffic. Then we implement a sequential application of the SSA based on [7] to detect abnormal changes in the time series corresponding to alarm flow. Based on the approach, we do not need to know the parametric model of the considered time series and have the ability to autonomously adapt to shifts in the structure of the flow. In the same time, we can model the time series corresponding to alarm flow directly without removing any flow components. To the best of our knowledge, this is the first study that applies SSA on the analysis of IDS alarm flows.

III. SINGULAR SPECTRUM ANALYSIS OF ALARM FLOW

The SSA method is a powerful non-parametric technique of time series analysis, and based on principles of multivariate statistics. The aim we using SSA is to capture normal flow behavior. In the following sections, we apply basic version of SSA [5] to the original alarm flow and decompose it into two independent components, then normal and abnormal parts of alarm flow are manifested.

A. SSA-Based Flow Behavior Decomposition

The process consists of four main steps, which are performed as flows:

1) Embedding. Let $X(t)$ be the time series corresponding a alarm flow, and $M < N$ be an integer called ‘lag’, and let $K = N - M + 1$. Making Hankelization process to form $M$-lagged vectors $X_i = [X_{i+1}, X_{i+2}, \ldots, X_{i+M}]^T, 1 \leq i \leq K$. The trajectory matrix of the time series is of dimension $M \times K$ and has the following form:

$$E = [X_1, X_2, \ldots, X_K]$$

(1)

The trajectory space is defined as the linear space spanned by the columns of $E$.

2) Singular Value Decomposition (SVD). An SVD of the matrix $R = EE^T$ (we shall call the lag-covariance matrix) provides us with M eigenvalues and eigenvectors. Let $\lambda_1, \lambda_2, \ldots, \lambda_M$ be the eigenvalues of $R$ and they are arranged in the decreasing order, so that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_M$, and $r = \text{max} \{i: \lambda_i > 0\}$. Let $U_1, \ldots, U_r$ denote the corresponding eigenvectors (principal components) and $V_j = \sqrt{\lambda_j} U_j / \sqrt{\lambda_j}, j = 1, \ldots, r$. Then the elementary matrix $E_j$ are obtained from $E_j = \sqrt{\lambda_j} U_j V_j^T$, and the trajectory matrix $H$ can be written as

$$E = E_1 + \cdots + E_r$$

(2)

3) Grouping. Since we assume that alarm flows are composed by normal and abnormal components, we split the set of indices $I = \{1, \ldots, r\}$ into two disjoint subsets, $I_1 = \{i_1, \ldots, i_l\}$ contains the first few of leading components, and $I_2 = I - I_1$ contains the residual components. Let $E_1 = E_{i_1} + \cdots + E_{i_l}$ be the approximation of the trajectory matrix $E$ based on the indices of $I_1$. Similarly, we have $E_2$ for the indices of $I_2$. Then the final decomposition of the initial trajectory matrix $E$ is

$$E = E_1 + E_2$$

(3)

4) Reconstruction. Approximation of the original time series can be performed by averaging the elements of $E_k (k=1,2)$, which are placed on the same anti-diagonal, that is, the elements $E_{ij}$ with $i + j = \text{constant}$. Then the decomposition of original time series is

$$X(t) = X_1^N(t) + X_2^N(t)$$

(4)

where $X_1^N (t)$ is the time-series reconstructed using the subset of leading components, which can be interpreted as the normal behavior part of the alarm flow $X(t)$, and $X_2^N (t)$ is the time series reconstructed using the subset of residual components, which can be interpreted as the abnormal part of $X(t)$.

B. Analysis

In this section we apply the above SSA decomposition algorithm on the alarms we have obtained before. We choose 5 minutes as the sampling interval, an alarm flows generated by the signature ICMP L3retriever Ping is used for more detailed analysis.
Before the decomposition, we should select the values of just two parameters, namely, the lag window length \( M \) and the number \( l \) of leading components contained in the subset \( I_l = \{i_1, \ldots, i_l\} \). The SVD performed on matrices obtained with a window length \( M \) is equivalent to that performed on matrices obtained with the complementary window \( K = N - M + 1 \). This means increasing the window length would reproduce results already tested with shorter window lengths, and too large window length may introduce some disturbance. Where, \( l \) is such that the first \( l \) components provide a good description of the normal part of the signal and the lower \( M-l \) components correspond to abnormal part. If \( l \) is too small (under-fitting), we miss a part of the normal signal. Alternatively, if \( l \) is too large (over-fitting), then we approximate a part of abnormal signal with the normal part. Both of these cases will make it difficult for us to detect the deviations from the normal profile.

Fig. 2 Percentage contribution of the eigenvalues: (a) \( M_1 = 60 \), (b) \( M_2 = 100 \).

To properly choose the parameters, we use the radio

\[
P_l = \frac{\lambda_i}{\sum_{i=1}^{M} \lambda_i}
\]

(5)
to estimate the energy contribution of the \( i \)-th candidate principal component to the original flow. Fig.2 presents the contribution of the eigenvalues \( \{\lambda_i\} \) corresponding to the time series of ICMP L3retriever Ping alarm flow. Two different widow lengths \( M_1 = 60 \) and \( M_2 = 100 \) are used for comparing and anglicizing the results.

As it can be seen from both cases, only the first few eigenvalues are responsible for the main part of the flow information. This means we can reconstruct the normal behavior part of the alarm flow by using the first few leading components. In fact, the first 3 eigenvalues account for more than 90 percent of the whole energy, therefore we prefer to choose \( M = 60 \) and \( l = 3 \) to continue the following experiments and analysis.

We use the first day’s data with the same alarm flow mentioned above to perform SSA-based decomposition and signal reconstructing. The reconstructed signal \( X^c(t) \) is plotted in Fig.3 (a) along with the original signal, and the residual series \( X^r(t) \) can be easily obtained from the equation (4) by \( X^r(t) = X(t) - X^c(t) \), which is plotted in Fig.3 (b).

![Fig.3 (a) Original alarm flow series (dotted line) and SSA-reconstructed series (continuous line) corresponding to normal flow behavior. (b) Residual series defined as the difference between the original and reconstructed series.](image)

Recognizing difference between the normal and abnormal behaviors from the original flow series may be a tough task. The residual series has a significantly different structure compared to its original version; even slight divergence can be revealed by it. This provides us valuable knowledge to explore the abnormal components of alarm flow, and so as to perform our final goal, that is, identifying the anomalies from the abnormal components. More details about the approach used to detect the anomalies will be explained in the next part of this article.

IV. ANOMALY DETECTION

According to the discussion before, alarm flow can be modeled as a dynamic and non-stationary signal series which continuously evolves with time. This means the variation in the structure of the normal flow behavior over time is, however, non-negligible. In this section we implement a sequential application of the SSA [7] to adapt to changes in normal flow behavior. Based on which we detect anomalies from the abnormal components.

A. Main Idea

The method is to apply SSA to a windowed portion of the signal. The basic idea is that if at a certain time moment \( \tau \) the mechanism generating the time series \( x_\tau \) has changed, then an increase in the distance between the \( l \)-dimensional hyperplane spanned by the eigenvectors of the lag-covariance matrix, and the \( M \)-lagged vectors \( X_j \) \( (x_{j+1}, \ldots, x_{j+M}) \) is to be expected for \( j \geq \tau \). For our anomalies detection, the \( l \)-dimensional hyperplane spanned by the eigenvectors just forms the normal
components of the flow (we refer to it as normal subspace, which is a \(l\)-dimensional subspace of the \(M\)-dimensional space \(R^M\)), to determine the number \(l\) of dimension of the hyper-plane is just to confirm how many leading components the subset \(I_j\) should contains. Then, the distance between the normal subspace and the vectors \(X_j\) reflect abnormal characteristics of the flow which is inherently determined by abnormal (residual) components of the flow.

Therefore, we use these distances to execute anomalies detection, and there is no need to reconstruct the residual series as before. If the flow structure does not change further along the series, then the hyper-plane composed by \(M\)-lagged vectors further along will stay close to normal subspace. However, if the structure changes further along, it will not be well described by the computed subspace, and the distance of trajectory matrix vectors to it will increase. This increase will signal the anomaly.

To make the detection process more effectively deal with changes in the flow series structure, we implemented an modified sequential change-points detection algorithm based on [7]. For each \(n\) we apply SVD to the trajectory matrix computed in a time window \([n+1, n+N]\) of length \(N\), where \(n\) is the iteration number which is responsible for the window’s sliding to accommodate new structure changes. The following is the description of the online detection algorithm.

\[\text{B. Formal Description of the Detection Algorithm}\]

Let \(X_{t_0}, X_{t_1}, \ldots, X_{t_T}\) be a alarm flow series with \(T \geq 0\) and the window width \(N\), the lag parameter \(M\), lag \(p\), and \(q\) be integers so that \(1 \leq M \leq N/2\) and \(0 \leq p < q\). Initializing \(n_0 = 0\) and \(c_0 = 0\), we execute the following operations.

\**Step 1:** Let \(n = n_0, c_0 = 0\), and create the trajectory matrix on the time window \([n+1, n+N]\), i.e. \(B^{(n)}\), size \(M\) by \(K\).

\[
B^{(n)} = \begin{pmatrix}
X_{n+1} & X_{n+2} & X_{n+3} & \cdots & X_{n+K} \\
X_{n+1} & X_{n+2} & X_{n+3} & \cdots & X_{n+K+1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
X_{n+1} & X_{n+2} & X_{n+3} & \cdots & X_{n+K+N}
\end{pmatrix}
\]

where \(K = N - M + 1\). The columns of the matrix \(B^{(n)}\) are the vectors:

\[
b^{(n)}_j = (x_{n+1}, \ldots, x_{n+j-M})^T \quad \text{with} \quad j = 1, \ldots, K
\]

\**Step 2:** Compute lag-covariance matrix \(R_n = B^{(n)}(B^{(n)})^T\).

Then determine \(M\) eigenvalues and eigenvectors of \(R_n\) and sort the eigenvectors in decreasing order.

\**Step 3:** Compute the contribution of each eigenvalue by using equation (5). The greater this percentage the more "effective" is the component corresponding to the eigenvalue. Then select the number of components to construct the \(l\)-dimensional normal subspace spanned by the eigenvectors \(U_1 \cdots U_l\).

\**Step 4:** Define a test matrix \(T^{(n)}\) on the window \([n + p + 1, n + q + M - 1]\),

\[
T^{(n)} = \begin{pmatrix}
x_{n+p+1} & x_{n+p+2} & x_{n+p+3} & \cdots & x_{n+q+p+1} \\
x_{n+p+2} & x_{n+p+3} & x_{n+p+4} & \cdots & x_{n+q+p+2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
x_{n+p+M} & x_{n+p+M+1} & x_{n+p+M+2} & \cdots & x_{n+q+p+M-1}
\end{pmatrix}
\]

whose columns are the vectors \(t^{(n)}_j (j = p+1, \ldots, q)\).

\**Step 5:** Compute \(D^{(n)}\), the sum of squared Euclidean distance between the vectors of the test matrix \(T^{(n)}\) and \(l\)-dimensional normal subspace. Since the eigenvectors are orthonormal, the square of the Euclidean distance between an \(M\)-vector \(Y\) and normal subspace spanned by the \(l\) eigenvectors \(U_1 \cdots U_l\) is just \(Y^T Y = Y^T U^T U Y\) - \(Y^T U U^T Y\), where \(|\cdot|\) is the usual Euclidean norm and \(U\) is the \(M \times l\)-matrix. Therefore

\[
D^{(n)} = \sum_{j=p+1}^{q} \left((t^{(n)}_j)^T U U^T t^{(n)}_j\right)
\]

\**Step 6:** Detection decision. Compute CUSUM statistic,

\[
W_{n} = S_{n} - S_{n_0} = (n_0 + 1) - 1/3(MQ)
\]

where \(S_n = D^{(n)}/V^{(n)}\) is normalized sum of squared distances. \(V^{(n)}\) is a variance of residual series of the flow which can be calculated by the sum of squared Euclidean distance between the vectors of the trajectory matrix \(B^{(n)}\) and \(l\)-dimensional normal subspace. Then, if \(W_n\) exceeds a threshold \(h\) (for more details in [7]),

\[
h = \frac{2}{MQ} \frac{D}{(MQ - Q^2 + 1)}
\]

where \(t_q\) is the \((1 - a)\) quantile of the standard normal distribution, then a anomaly is signaled and let \(n_0 = n_0 + p\), go to step 1. Otherwise, continue with Step 7.

\**Step 7:** If there is no anomaly be signaled then let \(p = p + 1\), \(q = q + 1\) and \(c_0 = c_0 + 1\). And if \(c_0 > Q/2\) then let \(n_0 = n_0 + c_0\). Lastly go to Step 1.

\**V. EXPERIMENTAL EVALUATION**

In this section we present the results obtained by applying the detection algorithm. We first discuss the parameter selection problems for the algorithm, and then we select AR approach [3] as a stationary model to make comparing experiments with our algorithm. Finally, we present less detailed results on alarm reduction capability.

\**A. Parameter Selection**

To run the SSA-based detection algorithm, a choice of five parameters should be made: sliding window size \(N\), lag parameter \(M\), the number of leading components used to construct normal subspace \(l\), and parameters \(p\) and \(q\) defining the test interval (and test matrix).

The window size \(N\) is a very important parameter. On one hand it should be large enough to allow capturing enough of a flow structure to use for anomalies detection. On the other hand, if it is too large then anomalies cause by some changes in the flow may be smoothed out and undetected. If \(N\) is too small, outliers may be taken as characteristic of the signal and thus false positives will be
increased. The lag $M$ determines the structure of a flow, including both normal and residual components. It should be chosen properly to hold enough components to depict the flow accurately. We tend to choose $M = N/2$.

To choose $l$, we follow the routines mentioned in section III of this paper. We take components of the flow so that the sum of the eigenvalues of selected components exceeds a certain amount of the total sum (90% here).

The parameters $p$ and $q$ define a part of the signal following the window from which the trajectory matrix was constructed. In choosing the values for $p$ and $q$, $q$ should be slightly larger than $p$, but not too large, in which case the detection statistics will smooth out the anomalies. We prefer to use $p = N-M+1$, and $q = p+1$.

**B. Evaluation Results**

To verify the effectiveness of our SSA-based detection algorithm, we will make some comparing with the stationary AR approach. The main criteria to compare the two approaches is that the anomalies raised by the models and the known anomalies in the analyzed flows. We chose one week’s data obtained in section II for the experiments. Corresponding alarm flows were generated at five minutes sampling interval.

We start with the alarm flow of signature *ICMP Destination Unreachable* (flow DU). For the SSA-based detection algorithm, setting the parameters, $N = 24, M = 12, p = N-M+1, \text{ and } q = p + M$. For the stationary AR approach, we implemented the model with AR(12). Denoting $E_t$ the isolated abnormal components (i.e. residual series) of the flow, an anomaly is signaled if the current value $e_t$ differs more than $n$ (here $n = 3$) standard deviations from the average of the past values. Fig. 4 (a) and Fig. 5 (a) represents the anomalies raised by AR and SSA models, respectively. The black line indicates the alarm intensity observed in each sampling interval over seven days. The raised anomalies by AR models are marked with diamond markers, while SSA models with
square markers. As shown in the figure, there are 5 distinct anomalies in this relatively stable flow. The two models are almost equal that both detected all 5 known anomalies. However, AR models signaled additional 7 points when the flow returned to the stable levels from the detected peaks. These have little attraction for us, and we regard them as false positives due to the model errors. To visualize the deviations from the normal flow behaviors, residual series of the AR models and CUSUM statistics respectively.

Another experiment was conducted on the alarm flow of signature ICMP L3rP (flow L3rP) that is a more complex and less stable flow. We used the same parameters as above. Fig.6 (a) and Fig.7 (b) depict the analysis results using the two models, respectively. The observed alarm intensity is drawn a black line and the raised anomalies are marked with diamond and square markers. The total number of anomalies raised by AR models is much higher than that of SSA models, 27 against 13. The SSA models identified all 13 known anomalies. The AR models, besides some false positives, signaled 9 but missed 4 points, which are marked with dotted arrow lines in Fig.6 (a). This can be explained that the AR models do not capture the normal flow behaviors properly when the structures of flow shift, which can also be observed by the deviations from the normal flow behaviors showed in Fig.6 (b) and Fig.7 (b).

C. Alarm Reduction

It is worth notice that given the subset of alarms we are analyzing, the anomalies are not necessarily attacks, but more likely more general problems. We aim by selectively filtering out alarms noise to free time and resources for both operators and other correlation approaches to focus on those more relevant alarms. Thus the alarm reduction and noise filtering are considered as more important objective than the anomaly detection.

Table 1: Alarm reduction results

<table>
<thead>
<tr>
<th>NO</th>
<th>Flow name</th>
<th>A</th>
<th>N</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>L3rP</td>
<td>140660</td>
<td>13</td>
<td>0.64%</td>
</tr>
<tr>
<td>2</td>
<td>SNMP request</td>
<td>31816</td>
<td>25</td>
<td>1.24%</td>
</tr>
<tr>
<td>3</td>
<td>DU</td>
<td>21714</td>
<td>5</td>
<td>0.25%</td>
</tr>
<tr>
<td>4</td>
<td>(http inspect)</td>
<td>20094</td>
<td>19</td>
<td>0.94%</td>
</tr>
<tr>
<td>5</td>
<td>SNMP trap</td>
<td>16745</td>
<td>7</td>
<td>0.35%</td>
</tr>
</tbody>
</table>

Take the flow L3rP being processed to illustrate alarm reduction capability. The example contains 140660 alarms. After processing, the SSA-based method marked 13 sampling intervals as anomalous points. The proportion of points necessitating operator’s attention is reduced to 0.64% (under 5-minute sampling interval). Without removing any components of the flow, all abrupt alarm bursts can be reported as anomalous and remains can be filtered as noises. Table 1 present filtering results using one week’s alarm data with signatures in Fig.1. The first column in the table is flow names. The number of alarm in the flows is given in column A. The method reported N anomalies and the proportion of anomalous points is given in column P. From the results, we conclude that the proportion of points that can be neglected with the processing method is significant, and the saved time may greatly help to reduce the burden of administrators.

VI. RELATED WORKS

As the objective of this work is only on alarm reduction, we focus on alarm correlation approaches addressing the same problem. Ning et al. developed an intrusion alarm correlator [8] to help human analysts to recognize multi-step attacks. Lee et al. [9][10] built a framework based on data mining techniques, such as sequential patterns mining and episodes rules, to search causal relationships between alarms to improve attack detection while maintaining a low false positive rate. Different with the two approaches, we model the normal behaviors in one specific type of alarm flow instead of trying to track attacks spreading across multiple steps. In other words, the targets and the used techniques are different.

Julisch proposed to find alarm clusters and generalized forms of false alarms to identify root causes [11][12], and those alarms which are not possible attributed to the root causes can be filtered out. However, the filters are based on alarm attributes and work on alarm-by-alarm basis, and besides, there is no finite set of root causes for some alarms. Instead of filtering, we propose to monitor the low impact alarms introduced by background noise, if it is possible to significantly reduce the number of alarms displayed to the administrator.

Pietraszek [13] built an alarm classifier system (ALAC, Adaptive Learner for Alert Classification) using a machine learning technique. The proposed system method works by classifying the alarms and sending the results to the analyst for further feedback. Through getting a feedback from the analyst, the system will subsequently update the classifier, which is then used to classify new alarm in the future. However, there are several limitations faced by this system. The ability of the analyst to correctly classify the alarms is the key for the method’s accuracy or performance, the analyst should be an expert in intrusion detection and able to make appropriate alarm verification to feedback. Hence, this system seems to be inefficient in reducing the human workload. In addition, as the system should perform a real-time analysis, adapting to the new knowledge as new data arrives is its biggest challenge. Moreover, applying additional background knowledge can become another challenge for the system in building an accurate alarm classifier.

Both our work and Viinikka [3] use aggregated alarm flows to model normal behaviors and detect deviations from the normal profile. But the manifest characteristics different from their works is that our SSA-based detection algorithm does not assume any parametric model or any structure, such as stationarity, instead we attempts to generate this model from intrinsic characteristics of the flow. In addition, the whole processing of our approach is based on a uniform model and it can be easily put into practice.
VII. CONCLUSIONS

Activities not considered as actual attacks always trigger IDS to create huge amounts of false alarms. The high false positive rate can even be exploited by attackers to overload security administrators. If these alarms are removed by simply tuning IDS detection rules or judging them as false by correlation engine, the administrator will lose some critical information.

In this paper we aim to modeling and filtering irrelevant alarm noises from alarm flows. The basic assumption is that regularities and smooth changes in the alarm intensity are considered as echoes of normal system use. This normal behavior is not observable at alarm level, and thus we monitor alarm flows. An alarm processing method based on SSA to summarize the behaviors of such alarm flows was presented to meet the objective.

According to the experience with the method, it can be used to highlight anomalies in high volume alarm flows. With this approach it is possible to make the high alarm levels associated with these flows more sustainable without deactivating them. Compared with the stationary AR model, our SSA-based method can well deal with the non-stationary natures inherent in the alarm flow. In addition the method does not assume any parametric model or any structure. Evaluation results from real network data show a significant increase in model accuracy, and more consistent filtering and detection behavior.

We believe that the method could be used as such, or in complement to other means of correlation, to monitor alarms considered as background noise of an operational system. The provided additional diagnostic capabilities may be modest, but more importantly via summarization the administrator can save time for more relevant tasks as he is informed only of significant changes in the noise level. In the future work, we plan to further improve our approach and apply it on more real life environments.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grant No. 60573120, and by the National High Technology Research and Development Program of China (863 Program) under Grant No. 2007AA01Z420, and by the key project sponsored by Natural Science Foundation of Hubei Province under Grant No. 2008CDA021.

REFERENCES


Jie Ma was born in wuhan, China in 1978. He received his M.E. degree in Computer Science from South Central University for nationalities, Wuhan, China, 2003. Currently, he is a PH.D. candidate in Information Security at Huazhong University of Science and Technology. His research interests include network security, p2p network and wireless network.

ZhiTang Li was born in JiZhou, Hubei Province, China in 1951, and received his M.E. degree in Computer Architecture in 1987 and PH.D. degree in Computer Architecture in 1992 from Huazhong University of Science and Technology, Wuhan, China. His research interests include network infrastructure, network security and Distributed Computing etc. He is currently the director of China education and Research Network(CERNET) in Central China. He has published more than one hundred papers in the area of computer architecture, network security and distributed Networks.