Detecting Network Anomalies Using Different Wavelet Basis Functions

Wei Lu, Mahbod Tavallaee and Ali A. Ghorbani
Information Security Center of Excellence
Faculty of Computer Science, University of New Brunswick
Fredericton, NB E3B 5A3, Canada

Abstract

Signal processing techniques have been applied recently for analyzing and detecting network anomalies due to their potential to find novel or unknown intrusions. In this paper, we present a novel network anomaly detection approach based on wavelet analysis, approximate autoregressive and outlier detection techniques. In order to characterize network traffic behaviors, we proposed fifteen features and applied them as the input signals in our wavelet-based approach. We then evaluate our approach with the 1999 DARPA intrusion detection dataset and conduct a comprehensive comparison for four different typical wavelet basis functions on detecting network intrusions. Our work aims to unveil a question when applying wavelet techniques for detecting network attacks, that is "do wavelet basis functions have an important impact on the intrusion detection performance?". Moreover, to the best of our knowledge, the work is the first to analyze the 1999 DARPA's network traffic using flow data instead of its original raw packet data.

1. Introduction

Intrusion detection has been extensively studied since the seminal report written by Anderson [1]. Traditionally, intrusion detection techniques are classified into two categories: misuse detection and anomaly detection. In misuse detection, attacks are detectable if their signatures can be identified by analyzing the audit trails or network traffic behaviors. However, misuse detection approaches are strictly limited to the latest known attacks. How to detect new attacks or variants of known attacks is one of the biggest challenges faced by misuse detection. To address the weakness of misuse detection, the concept of anomaly detection was formalized in the seminal report of Denning [2]. Denning assumed that security violations could be detected by inspecting abnormal system usage patterns from the audit data. As a result, most anomaly detection techniques attempt to establish normal activity profiles by computing various metrics and an intrusion is detected when the actual system behavior deviates from the normal profiles.

According to Axelsson, the early network anomaly detection systems are self-learning, that is, they automatically form an opinion of what the subject's normal behavior is [3]. Although machine learning techniques have achieved good results at detecting network anomalies so far, they are still faced with some major challenges, such as "limited capability for detecting previously unknown attacks due to large number of false alarms" [6], "behavioral non-similarity in training and testing data will totally fail leaning algorithms on anomaly detection" [5], and "can machine learning be secure?" [4]. Considered as an alternative to the traditional network anomaly detection approaches or a data preprocessing for conventional detection approaches, signal processing techniques have been successfully applied to the network anomaly detection (e.g. using CUSUM algorithm for DDoS detection [7]).

In this paper, we present a novel network anomaly detection approach based on wavelet analysis, approximate autoregressive and outlier detection techniques. Although applying wavelet techniques for modeling network anomalies has already been studied in recent literatures [11, 12, 13, 14], modeling the normal daily flow based network traffic using wavelet analysis and approximate autoregressive techniques is new in the network security community. Moreover, in this work, we address a basic question when applying wavelet techniques for detecting network intrusions, i.e. do wavelet basis functions have an important impact on the intrusion detection performance?, through evaluating and comparing four different typical wavelet basis functions based on the 1999 DARPA intrusion detection dataset.

The rest of the paper is organized as follows. Section 2 introduces related work, in which we summarize existing works on applying wavelet analysis techniques for intrusion detection. Section 3 presents the theoretical foundation of wavelet analysis and autoregressive techniques. Section 4 presents our detection approach. In particular, we describe the fifteen flow based features in detail, introduce the methodology for modeling the normal daily traffic and describe the outlier detection algorithm for intrusion decision. Section 5 presents the experimental evaluation of our approach and discusses the obtained results. Section 6 makes some concluding remarks and discusses future work.
2. Related work

In the work of Barford et al. [11], wavelet transform is applied for analyzing and characterizing the flow based traffic behaviors. The netflow data are collected from Cisco routers deployed on different locations in a large university network. The motivation to use wavelet analysis technique is its inherent time-frequency property that allows splitting signals into different components at several frequencies in which a deviation algorithm is presented to identify anomalies by setting a threshold for the signal composed from the wavelet coefficients at different frequency levels. The evaluation results show that some forms of DoS attacks and port scans are detected within mid-band and high-band components due to their inherent anomalous alterations generated in patterns of activity.

In [12], Nayyar and Ghorbani apply an approximate autoregressive model for detecting network intrusions. The model is based on wavelet analysis, and packet rate is used as the input network signal to the model. Experimental evaluation with the 1999 DARPA dataset shows that the wavelet based model is effective in detecting anomalies. However, only one network signal and one wavelet basis function are considered in the work.

Focusing on specific types of network attacks, wavelet analysis was used to detect DoS or DDoS attacks in [13, 14]. In [13], Li and Lee found that aggregated traffic has strong bursty across a wide range of time scales. They applied wavelet analysis to capture complex temporal correlation across multiple time scales with very low computational complexity. Their experimental evaluation results with typical Internet traffic trace show that energy distribution variance changes always cause a spike when traffic behaviors affected by DDoS attacks. In contrast, normal traffic exhibits a remarkably stationary energy distribution. In [14], Dainotti et al. presented an automated system to detect volume-based anomalies in network traffic. The system combines the traditional approaches, such as adaptive threshold and cumulative sum, with a novel approach based on the continuous wavelet transform. They obtained good results in terms of correct detections and false alarms with several public intrusion detection datasets. However, considering only volume of traffic limits their system’s capability to find other types of attacks such as scans.

Not only applied for detecting network anomalies, wavelet analysis was also widely used in network measurement from different perspectives. For example, in [15], wavelets were used to separate the short term behaviour of the streams, where the jittering or chaff indeed masks the correlation, from the long-term behaviour of the streams, where the correlation remains, and in [16] the multi-resolution wavelet analysis was used to accomplish the desired intrusion detection and the subsequent construction of self-similarity in the simulated traffic.

3. Overview of wavelet analysis and ARX

3.1. Wavelet transform

The Fourier transform is well suited only to the study of stationary signals in which all frequencies are assumed to exist at all times. The short term Fourier transform proposed by Gabor localizes the Fourier analysis by taking into account a sliding window [17] and its major limitation is that it can either give a good frequency resolution or a good time resolution (depending upon the window width). In order to have a coherence time proportional to the period, Morlet proposed Wavelet transform that can achieve good frequency resolution at low frequencies and good time resolution at high frequencies [18]. Further details about Fourier analysis, short term Fourier analysis and Wavelet transform can be found in [19].

The Discrete Wavelet Transform (DWT) is used in our work since the network signals we consider have a cut-off frequency. DWT is a multi-stage algorithm that uses two basis functions called wavelet function \( \psi(t) \) and scaling function \( \phi(t) \) to dilate and shift signals as follows:

\[
\phi_{j,k}(t) = -\frac{j}{2^j} \phi(2^j t - k) \quad (1)
\]

\[
\psi_{j,k}(t) = -\frac{j}{2^j} \psi(2^j t - k) \quad (2)
\]

Equations (1) and (2) are then applied to transform input signals into a set of coefficients at each stage \( j \) that are given by:

\[
C_{j,k} = <X, \phi_{j,k}(t)> \quad (3)
\]

\[
D_{j,k} = <X, \psi_{j,k}(t)> \quad (4)
\]

where \(<*,*>\) is the inner product.

For reconstructing the signals, the approximation and detail signal in each stage \( j \) can be obtained using Equations (5) and (6):

\[
\text{Approx}_j(t) = \sum_k C_{j,k} \phi_{j,k}(t) \quad (5)
\]

\[
\text{Detail}_j(t) = \sum_k D_{j,k} \psi_{j,k}(t) \quad (6)
\]

Finally, the input signal \( X \) can be reconstructed as follows:

\[
X = \text{Approx}_j(t) + \sum_{j=0}^J \text{Detail}_j(t) \quad (7)
\]
The general implementation of wavelet transform is based on filter bank or pyramidal algorithm [19]. Signals are passed through a low pass filter (H) and a high pass filter (G) at each stage. Given a signal s, where \( i = 1, 2, 3, \ldots, 2^n \), we have:

\[
(G(s))_k = \sum_k s_{j-2k} \quad (8)
\]

\[
(H(s))_k = \sum_k h_{j-2k} s_j \quad (9)
\]

Given a signal with length \( l \), we can obtain a filtered signal with length \( l \). Since there are two filters in each filtering stage, the total number filtered signals are \( 2l \). In order to remove the redundancies in signals, we down sample the low pass and high pass filtered signals by half, without any information loss. The size of data can be reduced through down sampling since we are interested only in approximations. After the low level details have been filtered out, the rest of coefficients represent a high level summary of signal behaviours and thus we use them to establish a signal profile characterizing the expected level summary of signal behaviours and thus we use them to a-trous and redundant wavelet transforms that don't down sample signals after filtering [20], we use filter banks algorithm in our work since it is suitable for our purpose.

3.2. ARX modeling for system identification

System identification deals with the problem of identifying mathematical models of dynamical systems by using observed data from the system. In a dynamical system, its output depends both on its input as well as on its previous outputs. As we have known, ARX (AutoRegressive with eXogenous inputs) model is widely used for system identification. Let \( x(t) \) represent the regressor or predictor input and \( y(t) \) denote the output generated by the system we are trying to model. Then ARX \([p,q,r]\) can be represented by a linear difference Equation (11):

\[
y(t) = \sum_{i=1}^{p} a_i y(t-i) + \sum_{i=r}^{q} b_i x(t-i) + e(t) \quad (11)
\]

where \( a_i \) and \( b_j \) are the model parameters. Equation (11) can be transformed into Equation (12) by forwarding time-shift operator \( q \):

\[
A(q)y(t) = B(q)x(t) + e(t) \quad (12)
\]

Given an ARX model with parameters \( \theta \), we have the following Equation to predict the value of next output:

\[
\hat{y}(t|\theta) = \sum_{i=1}^{p} a_i y(t-i) + \sum_{i=r}^{q} b_i x(t-i) \quad (13)
\]

and the prediction error \( \xi(t) \) is given by:

\[
\xi(t) = y(t) - \hat{y}(t|\theta) \quad (14)
\]

In order to create the ARX model, we have to estimate the parameters \( \theta \). Let \( \xi(t, \theta) \) denote the prediction error at time \( t \) and \( Z^N \) represent the input-output set, in which \( Z^N = \{u(0), y(0), \ldots, u(N), y(N)\} \), we have normalized prediction error denoted by \( V \) as follows:

\[
V_N(\theta, Z^N) = \frac{1}{N} \sum_{t=1}^{N} \frac{1}{2} (y(t) - \phi^T(t)\theta)^2
\]

where \( \phi^T(t) = [-y(t-1) \ldots -y(t-n_y)u(t) \ldots u(t-n_u)]^T \)

The function is quadratic in \( \theta \) and we can use least-square estimate techniques to obtain the optimal value of parameters \( \theta \), which is given in the following Equation:

\[
\hat{\theta}^{LSE}_N = \arg \min V_N(\theta, Z^N) = \frac{1}{N} \sum_{t=1}^{N} (\phi(t)\phi^T(t))^{-1} \frac{1}{N} \sum_{t=1}^{N} \phi(t)y(t)
\]

Based on the above Equation, a predictive model for network signals can be established by training model parameters on collected network traffic data.

4. The proposed framework

The general architecture of our approach is illustrated in Figure 1. It consists of three components, namely feature analysis, normal traffic modeling based on wavelet decomposition and approximate autoregressive, and intrusion decision.

![Figure 1: General architecture of the proposed detection framework](image-url)
over a time interval) or existing features from public intrusion detection dataset (i.e. 41 features from KDD 1999 CUP intrusion detection dataset [8]) as the input signals. Based on these proposed features, normal daily traffic is then modeled by using the wavelet coefficient and approximate autoregressive techniques. The output for the normal traffic model is the residual that represents the deviation of current input signal from normal behavioral signals. Residuals are input to the intrusion decision engine in which an outlier detection algorithm is running and making intrusion decisions.

4.1. Feature analysis

The major goal of feature analysis is to select and extract robust network features that have the potential to discriminate anomalous behaviors from normal network activities. Since most current network intrusion detection systems use network flow data (e.g. netflow, sflow, ipfix) as their information sources, we focus on features in terms of flows. In order to define our feature vector space, we select five basic metrics to measure the entire network behaviors. Table 1 describes each metric in detail.

Table 1: List of metrics

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FlowCount</td>
<td>A flow consists of a group of packets going from a specific source to a specific destination over a time period.</td>
</tr>
<tr>
<td>PacketCount</td>
<td>The average number of packets in a flow over a time interval.</td>
</tr>
<tr>
<td>ByteCount</td>
<td>The average number of bytes in a flow over a time interval.</td>
</tr>
<tr>
<td>PacketSize</td>
<td>The average number of bytes per packet over a time interval.</td>
</tr>
<tr>
<td>FlowBehavior</td>
<td>Ratio of FlowCount to PacketSize.</td>
</tr>
</tbody>
</table>

Based on the five metrics listed in Table 1, we define in Table 2 a set of features to describe entire traffic behaviors on networks.

Empirical observations with the real network traffic and 1999 DARPA network traffic show that the network traffic can be characterized and discriminated through these features.

4.2. Normal network traffic modeling with wavelet and ARX

Modeling the normal network traffic consists of two phases, namely wavelet decomposition and generation of auto regressive model. During wavelet decomposition, the original signals are transformed into a set of wavelet approximation coefficients that represent an approximate summary of the signal, since details have been removed during filtering. Next, in order to estimate ARX parameters and generate ARX prediction model, we use the wavelet coefficients from one part of training data as inputs and wavelet coefficients from the other part of training data as the model fitting data. The ARX fitting process is used to estimate the optimal parameters based on least square errors. The whole procedure for modeling the normal network traffic is illustrated in Figure 2.

Table 2: List of features

<table>
<thead>
<tr>
<th>Features</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1 )</td>
<td>Number of TCP Flows per Minute</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>Number of UDP Flows per Minute</td>
</tr>
<tr>
<td>( f_3 )</td>
<td>Number of ICMP Flows per Minute</td>
</tr>
<tr>
<td>( f_4 )</td>
<td>Average Number of TCP Packets per Flow over 1 Minute</td>
</tr>
<tr>
<td>( f_5 )</td>
<td>Average Number of UDP Packets per Flow over 1 Minute</td>
</tr>
<tr>
<td>( f_6 )</td>
<td>Average Number of ICMP Packets per Flow over 1 Minute</td>
</tr>
<tr>
<td>( f_7 )</td>
<td>Average Number of Bytes per TCP Flow over 1 Minute</td>
</tr>
<tr>
<td>( f_8 )</td>
<td>Average Number of Bytes per UDP Flow over 1 Minute</td>
</tr>
<tr>
<td>( f_9 )</td>
<td>Average Number of Bytes per ICMP Flow over 1 Minute</td>
</tr>
<tr>
<td>( f_{10} )</td>
<td>Average Number of Bytes per TCP Packet over 1 Minute</td>
</tr>
<tr>
<td>( f_{11} )</td>
<td>Average Number of Bytes per UDP Packet over 1 Minute</td>
</tr>
<tr>
<td>( f_{12} )</td>
<td>Average Number of Bytes per ICMP Packet over 1 Minute</td>
</tr>
<tr>
<td>( f_{13} )</td>
<td>Ratio of Number of Flows to Bytes per Packet (TCP) over 1 Minute</td>
</tr>
<tr>
<td>( f_{14} )</td>
<td>Ratio of Number of Flows to Bytes per Packet (UDP) over 1 Minute</td>
</tr>
<tr>
<td>( f_{15} )</td>
<td>Ratio of Number of Flows to Bytes per Packet (ICMP) over 1 Minute</td>
</tr>
</tbody>
</table>

Figure 2: Procedure for modeling normal network traffic
After the prediction model for the normal network traffic is obtained, we can use it to identify anomalous signals from normal ones. When the input to the model includes only normal traffic, its output, called residuals, will be close to 0, which means the predicted value generated by the model is close to the actual input normal behaviors. Otherwise, when the input to the model includes normal traffic and anomalous traffic, the residuals will include a lot of peaks where anomalies occur. In this case, residuals are considered as a sort of mathematical transformation which tries to zeroize normal network data and amplify the anomalous data.

### 4.3. Outlier detection and intrusion decision

According to the above section, we assume that the higher the value of residuals, the more anomalous the flow is. As a result, in order to identify the peaks (or outliers) of residuals, we apply an outlier detection algorithm based on Gaussian Mixture Model (GMM) and make intrusion decisions based on the results of the outlier detection algorithm [21].

The intrusion decision strategy is based on the outcome of outlier detection: if no outlier data are detected, the network flows are normal; otherwise, the network flows represented by this outlier is reported as the intrusion.

### 5. Experimental evaluation

We evaluate our approach with the 1999 DARPA intrusion detection dataset. In particular, we conduct an analysis for network traffic collected on one whole day and identify the intrusions appeared on that day. Since most current existing network intrusion detection systems use network flow data (e.g. network, sflow, ipfix, etc.) as their information sources, we convert all the raw TCPDUMP packet data into flow based traffic data by using the public network traffic analysis tools (i.e. editcap, tshark). To the best of our knowledge, this is the first work to convert the full 1999 DARPA network packet logs into network flow based logs since the 1998 DARPA intrusion detection dataset has been converted into connection based dataset in 1999 (i.e. 1999 KDDCUP intrusion detection dataset). Although the 1998 and 1999 DARPA dataset was criticized in [22] due to the methodology for simulating actual network environment, they are the widely used and acceptable benchmark for the intrusion detection research.

During the evaluation, the results are summarized and analyzed in two categories, namely how many attack types are detected by each feature and all features correlation and how many attack instances are detected for each attack type. We don't use the traditional Receiver Operating Characteristic (ROC) curve to evaluate our approach and analyze the tradeoff between the false positive rates and detection rates because ROC curves are often misleading and incomplete [23]. Next, we explain the method for converting the 1999 DARPA's TCPDUMP packet logs into network flow based logs, analyze and compare the residuals for different wavelet basis functions, and discuss the intrusion detection results using four typical wavelet basis functions.

#### 5.1. Converting 1999 DARPA dataset into flows

The 1999 DARPA intrusion detection evaluation dataset has been widely used for evaluating network anomaly detection systems since it was created and extended in 1999 as a succession of the 1998 DARPA's dataset. The original 1999 DARPA's dataset is based on raw tcpdump log files and thus most of current evaluations are based on signatures in terms of packets.

Two existing tools (editcap, tshark) are used to convert the DARPA tcpdump files into flow logs. First, editcap is used to split the raw tcpdump file into different tcpdump files based on a specific time interval. In this case, we set the time interval as one minute in order to keep it the same as the time interval of flow data provided by most industrial standard. An example of using editcap is as follows:

```bash
eeditcap -A '1999-04-09 09:00:00' -B '1999-04-09 09:01:00' inside.tcpdump 1.pcap
```

Then, the tcpdump traffic data over the specific time interval is converted into flow logs by tshark through the following commands:

```bash
tshark -r 1.pcap -q -n -z conv,udp
tshark -r 1.pcap -q -n -z conv,tcp
tshark -r 1.pcap -q -n -z ip,icmp
```

Finally, the format of the generated DARPA flow logs is as follows:

```python
{ timestamp, local src IP : src port, remote dst IP : dst port, incoming number of packets (remote → local), incoming number of bytes, outgoing (local → remote) number of packets, outgoing number of bytes, total number of packets, total number of bytes, protocol }
```

#### 5.2. Analysis for residuals

The purpose for analyzing the residuals is to support our assumption in Section 4.3, that is the higher the value of residuals, the more anomalous the flow is. The traffic data we analyze was colleted on Monday, Week 5 (Apr 5, 1999). It includes not only normal behaviors, but also a large number attacking activities. As an example, we selected the first feature (f1) and compared its residuals generated by four different wavelet basis functions, namely Daubechies1, Coiflets1, Symlets2 and Discrete Meyer. Figure 3 illustrates the original network behaviors...
characterized by the selected feature (i.e. Number of TCP Flows per Minute) over one day. Figures 4, 5, 6, and 7 illustrate the network behaviors characterized by residuals over the four different wavelet basis functions.

By comparing the results showed in Figures 3 and 4, 5, 6, 7, we conclude that the peaks of residuals identify exactly the location where attacks happen even they are generated by different wavelet basis functions.

Figure 3: Number of TCP flows per minute

Figure 4: Residuals for number of TCP flows per minute by Daubechies1

Figure 5: Residuals for number of TCP flows per minute by Coiflets1

Figure 6: Residuals for number of TCP flows per minute by Symlets2

Figure 7: Residuals for number of TCP flows per minute by Discrete Meyer

5.3. Experimental settings and intrusion detection results

We have known that the 1999 DARPA data includes 5 weeks data and we use notation "w1d1" to represent data on Monday of First Week. During the training phase, in order to generate the external regressor we create the input signal by averaging and smoothing the first 7 days of data (w1d1, w1d2, w1d3, w1d4, w1d5, w3d1 and w3d2). Based on this new generated signal we get wavelet approximation coefficients, which act as the external regressor input into the ARX model. Then, we get another test signal by averaging and smoothing the remaining 3 days of normal data (w3d3, w3d4 and w3d5) and use this test signal to fit the ARX model. An ARX [5 5 0] model was fitted to the data using the least squares error method. The parameter outlier refers to the minimum mixing proportion in the outlier detection algorithm, and we set it as 0.00001 during the evaluation.

We evaluate our approach with Monday, Week5 from the 1999 DARPA flow logs. There are 20 attack types on that day, namely, apache2, arpppoison, crashis, dict, dosnake, fbbconfig, guesstelnet, imap, ipswipe, loadmodule, l s, ncftp, neptune, pod, portsweep, selfping, smtp, syslogd, udp storm, warezclient. See [9] for detail description of each attack. The evaluation results are analyzed in two categories, namely how many attack types are detected by each feature and all features correlation, and how many attack instances are detected for each attack type.
Table 3: Number of attack types detected for each feature by different wavelet basis functions

<table>
<thead>
<tr>
<th>Features</th>
<th>Daubechies1</th>
<th>Coiflets1</th>
<th>Symlets2</th>
<th>Discrete Meyer</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1</td>
<td>apache2, neptune</td>
<td>apache2, neptune</td>
<td>apache2, neptune</td>
<td>apache2, neptune</td>
</tr>
<tr>
<td>f2</td>
<td>pod</td>
<td>null</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td>f3</td>
<td>smurf, apache2</td>
<td>udpstorm, selfping</td>
<td>udpstorm, selfping</td>
<td>udpstorm, selfping</td>
</tr>
<tr>
<td>f4</td>
<td>udpstorm, selfping</td>
<td>udpstorm, selfping</td>
<td>udpstorm, selfping</td>
<td>udpstorm, selfping</td>
</tr>
<tr>
<td>f5</td>
<td>apache2, dict</td>
<td>arppoison</td>
<td>arppoison</td>
<td>arppoison</td>
</tr>
<tr>
<td>f6</td>
<td>udpstorm, selfping</td>
<td>udpstorm, selfping</td>
<td>udpstorm, selfping</td>
<td>udpstorm, selfping</td>
</tr>
<tr>
<td>f7</td>
<td>apache2, dict</td>
<td>arppoison</td>
<td>arppoison</td>
<td>arppoison</td>
</tr>
<tr>
<td>f8</td>
<td>udpstorm, selfping</td>
<td>udpstorm, selfping</td>
<td>udpstorm, selfping</td>
<td>udpstorm, selfping</td>
</tr>
<tr>
<td>f9</td>
<td>apache2</td>
<td>apache2</td>
<td>apache2</td>
<td>apache2</td>
</tr>
<tr>
<td>f10</td>
<td>udpstorm</td>
<td>null</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td>f11</td>
<td>udpstorm</td>
<td>null</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td>f12</td>
<td>udpstorm</td>
<td>null</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td>f13</td>
<td>udpstorm</td>
<td>null</td>
<td>null</td>
<td>null</td>
</tr>
<tr>
<td>f14</td>
<td>udpstorm</td>
<td>null</td>
<td>null</td>
<td>null</td>
</tr>
</tbody>
</table>

Table 4: Number of attack instances detected for each attack type by different basis functions

<table>
<thead>
<tr>
<th>Attack type</th>
<th>W5D1</th>
<th>Number of attack instances for each attack type</th>
<th>Detected number of attack instances for each attack type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Daubechies1</td>
</tr>
<tr>
<td>apache2-dos</td>
<td>30</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>arppoison-probe</td>
<td>15</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>dict-r2l</td>
<td>17</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>ncfp-r2l</td>
<td>8</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>pod-dos</td>
<td>3</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>portsweep-probe</td>
<td>4</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>selfping-dos</td>
<td>2</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>smurf-dos</td>
<td>2</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>udpstorm-dos</td>
<td>16</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>crashis-dos</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3 illustrates the number of attack types detected by each feature using four different wavelet basis functions. We found that the wavelet basis function is sensitive to features. That is, one basis function operating well for one feature might have bad results for the other features. For example, Coiflets1 is better than Symlets2 in terms of $f_1$, but worse than it in terms of $f_{13}$. In order to achieve an optimal solution, we have to use different wavelet basis functions for different features. Table 4 illustrates the number of attack instances detected for each attack type by different wavelet basis functions. Since attacks always lasted couple of minutes in DARPA, we consider all traffic appeared over the attacking period are anomalous behaviors. Thus, even only one attack instance is identified during the attacking period, we still can say the approach identify this attack type successfully. According to Table 4, 10 attack types are detected by Daubechies1, Coiflets1 and Symlets2 over total 20 attack types on that day, and in particular Daubechies1 detected 7 DoS attacks over total 10 DoS attacks on that day, while Coiflets1 and Symlets2 detected 6 DoS attacks. Generally speaking, we conclude that Daubechies1 basis function achieves the best results compared to other three wavelet basis functions.

As we discussed before, we don’t use ROC curves to evaluate our approach. Moreover, we don’t calculate the traditional detection performance metric FPR (false positive rate) during the evaluation. The main reason is that the residuals of an attack behavior have an impact on the following successive normal traffic. As a result, residuals of an attack behavior will be mixed into the normal traffic and identifying this kind of behaviors is blurred. Ignoring these blurring behaviors during the evaluation will generate a large number of false alarms. A possible solution to this issue is that we may define an attack decaying period $t_{\text{decay}}$ which starts from the exact time point $t_{\text{attack}}$ when attacks happen. When we find an attack at $t_{\text{attack}}$, we consider all following traffic behaviors over $[t_{\text{attack}}, t_{\text{attack}} + t_{\text{decay}}]$ as intrusions.

6. Conclusions and future work

In this paper, we present a novel network anomaly detection approach based on wavelet transformation. The
input signal is a 15-dimensional feature vector, which is defined to characterize the behaviour of the network flows. Normal daily traffic prediction model is established, in which wavelet coefficients play an important role. The outputs of this traffic prediction model are called residuals that measure the difference between normal and anomalous activities. The empirical observations show that the peaks of residuals always stand for the location where attacks occur. As a result, an outlier detection algorithm based on GMM is implemented in order to detect peaks from a set of residuals. Decisions are made based on the results of the proposed outlier detection algorithm.

A comparative study between four typical wavelet basis functions on detecting network intrusions with the 1999 DARPA dataset shows that the Daubechies wavelet families have the slightly better performance than other three wavelet families, namely Coiflets, Symlets and Discrete Meyer. Moreover, in order to improve the detection performance, we need to define more features that can characterize network behaviors and apply a more adaptive tuning algorithm to find the outliers of residuals.

In the near future, we will test our approach and do a comparison using a real traffic data. As criticized by McHugh in [20], DARPA dataset being used is orthogonal in terms of the simulated normal events and the attack events. It is imperative now to include accurate models of normal behaviours to be able to distinguish abnormal behaviours. A challenge for this work is to construct a new and pure normal traffic dataset, which not only simulates the network traffic in reality, but also guarantees that no attacks are included in this dataset. Moreover, in order to conduct a truer test, we will test all wavelet order for each wavelet family since current evaluation only considers one order from each wavelet family and most likely higher order of one wavelet family will perform equally to a certain order of the other wavelet family.

References