Designing a Hybrid Approach with Computational Analysis and Visual Analytics to Detect Network Intrusions

Dong Hyun Jeong  
Dept. of Computer Science  
University of the District of Columbia  
Washington, D.C. 20008–1122  
Email: djeong@udc.edu

Bong-Keun Jeong  
Dept. of Computer Information Systems  
Metropolitan State University of Denver  
Denver, CO 80217  
Email: bjeong@msudenver.edu

Soo-Yeon Ji  
Dept. of Computer Science  
Bowie State University  
Bowie, Maryland 20715–3319  
Email: sji@bowiestate.edu

Abstract—As network traffic has become more prevalent and complex, understanding network traffic patterns is vital to design innovative detection methods. Researchers have proposed various intrusion detection techniques by integrating different machine learning algorithms, but these techniques commonly share a high false-positive rate problem. In this study, we propose a hybrid approach of integrating computational analysis with visual analytics to detect network intrusions. For the computational analysis, both Multi-Resolution Analysis (MRA) and Principal Component Analysis (PCA) are applied to analyze network traffic data. First, Discrete Wavelet Transform (DWT) is utilized as the MRA approach to extract features from the network traffic data. After determining statistically significant features from a statistical validation, PCA is applied to transform the extracted features for identifying principal components in eigenspace. Lastly, a visual analytics tool is designed to help the user conduct an interactive visual analysis on detected network intrusions by initiating interactive visual validation and verification. We found that our approach is useful to detect the network intrusions as well as to understand their patterns.

I. INTRODUCTION

The Internet provides many useful services like news, email, storage, and social network. However, this technology has a lot of security related issues because the use of Internet services often requires to inadvertently put personal information in online storages. While people may assume that their personal information is well protected, recent online attacks indicate that it is not 100% secure. Many organizations and businesses invest significant amounts of budget in information security and data privacy to protect their computing infrastructures. They also organize computer security incident response teams to protect data storages, guarantee data confidentiality, and prevent external intrusions. Since attacks are major threats to the integrity of user data, it is important to build a trust when providing services to customers. Consequently, designing efficient intrusion detection techniques is extremely crucial. To address this challenge, researchers have proposed numerous intrusion detection techniques. A traditionally known intrusion detection system discovers threats by analyzing network packets at the network layer, and directly compare them to known patterns. But, detecting network intrusions by referencing known attack patterns has a limitation to identify unknown (or new) intrusions. For this reason, analyzing network patterns without previous knowledge about intrusions is considered. However, this approach also has an issue of high false alarm rate (i.e. high false-positives).

In this paper, we emphasize the importance of integrating both computational analysis and visual analytics to detect network intrusions. For the computational analysis, both Multi-Resolution Analysis (MRA) and Principal Component Analysis (PCA) are used. MRA [1] is an approach that focuses on representing different scales or frequency components in signals by utilizing signal processing techniques. Wavelet analysis is one of the commonly used techniques for MRA. It analyzes localized variations of power within network traffic data. Since the wavelet analysis decomposes data into time-frequency space, it is possible to determine dominant modes of variability [2]. In the past, researchers applied the wavelet analysis to detect anomalies. However, our approach is different because we apply the wavelet analysis to determine and extract informative features from the data. There are various wavelet analysis methods such as Continuous Wavelet Transform (CWT), Fast Wavelet Transform (FWT), Discrete Wavelet Transform (DWT), and Stationary Wavelet Transform (SWT). In our study, Discrete Wavelet Transform (DWT) is chosen as it is suitable for understanding non-stationary data such as network traffic. Based on DWT, various numbers of features depending on input parameters are determined. A statistical validation is then performed to remove non-statistically significant features as well as all zero-variance features (i.e. features that have the same values). The objective of applying the statistical validation is to increase the accuracy of detecting network intrusions by removing insignificant data. Then, PCA is applied to the significant features to determine principal components. Lastly, a visual analytics tool is designed to support an interactive visual analysis to further explore intrusions. The tool utilizes the principal components to represent the determined features in 2D display space so that network activities are displayed with forming clusters which
indicate either attacks or normal. Also, several user interaction techniques are added to assist the user in conducting the interactive visual analysis.

The rest of the paper consists of six sections. First, we discuss prior research in network intrusion detections and the benefits of utilizing visualization techniques. Then, our approach of analyzing network traffic data to determine intrusions is introduced with an emphasis on the importance of designed visualization tool. In section V, we present the performance of our approach and conclude with discussions, conclusion, and future work.

II. Previous Work

In the early network intrusion detection literature, statistical analysis was broadly applied to detect intrusions by performing a statistical comparison of current network events to a pre-determined set of baseline criteria. However, the statistical analysis has a limitation of identifying new types of attacks. Since the popularity of the Internet produces a massive amount of network traffic patterns as well as unknown attacks, designing an innovative intrusion detection analysis is considered as a major research area in cybersecurity. To accomplish this challenge, researchers proposed two intrusion detection techniques: signature-based detection techniques and behavior-based or anomaly-based detection techniques [3].

Signature-based techniques (also known as knowledge-based or misused-based techniques) are based on known attack signatures or system vulnerabilities. Since signature-based techniques analyze current network traffic patterns by comparing them with the attack signatures, they are quite accurate for detecting known attacks, but less effective against unknown attacks. Therefore, a constant update on attack signatures should be performed which may require considerable resources and overhead. Behavior-based or anomaly-based techniques perform an analysis of detecting intrusions via deviations from normal or expected traffic behaviors. Although the two types (anomaly vs. behavior) maintain considerable overlap and are often regarded as identical in literature, it is important to note that there is a slight difference between them. Anomaly-based techniques create a normal profile by training current network traffic, and then use the profile to detect deviations. On the other hand, behavior-based techniques do not necessarily compare against a baseline profile. Although these techniques can be effective to detect network intrusions in real-time, there are significant limitations. Signature-based techniques cannot guarantee to detect unknown network intrusions, and behavior (or anomaly)-based techniques show a significant performance issue under heavy traffic or sudden traffic bursts [4]. To address the limitations, researchers have studied on designing new techniques which integrate different Machine Learning (ML) techniques [5], [6].

In 1990’s, Cannady [7] emphasized the usefulness of Artificial Neural Networks (ANNs) for intrusion detection. However, ANNs are not used widely since the accuracy to detect intrusions is closely dependent on datasets and methods used in training. Furthermore, they do not provide a detailed reason about detected intrusions. Amor et al. [8] utilized Naïve Bayes (NB) for detecting intrusions. As a simplified Bayesian probability model, NB classifier operates based on the likelihood that one attribute does not affect others. It is faster than Decision Tree (DT) for learning and classifying, but there is no significant performance difference between the two. As the number of studies on designing new intrusion detection techniques by adapting ML algorithms increased in early 2000, Nguyen and Armitage [9] surveyed various network traffic classification algorithms appeared in the period between 2004 and early 2007. They found that different ML algorithms such as AutoClass, Expectation Maximisation (EM), DT, and NB demonstrate high accuracy. However, most approaches are uniquely designed to define their classification models by evaluating different test datasets. Therefore, the models are less efficient to analyze different datasets and network circumstances. To overcome this limitation, researchers have continuously sought and adopted various new ML algorithms. Wang [10] showed the effectiveness of logistic regression (LR) modeling to detect multi-attack types. Albayati and Issac [11] compared the performance of detecting intrusions among NB, Random Tree Classifier (rTree), and Random Forest Classifier (rForest), and found that rForest is superior to others with maintaining low false alarm rate.

Many researchers utilized Support Vector Machine (SVM) to conduct intrusion detection analysis. SVM is well-suited for classifying data by finding the hyperplane that maximizes the margin among all intrusion classes [12], [13], [14], [15], [16]. It simply classifies the input data by using a set of support vectors representing data patterns. However, SVM classification depends mainly on the used kernel types and parameter settings [16]. It also requires longer training time than other classification algorithms. To address this limitation, Khan et al. [12] proposed an approach of integrating hierarchical clustering analysis. Genetic Algorithm (GA) has been used for various purposes in intrusion detection such as optimization, automatic model generation, and classification [17], [18]. GA is a search algorithm that utilizes the mechanics of natural selection and genetics. It is often used to generate detection rules or select appropriate features from the input data. However, the classification accuracy using GA is slightly lower than tree algorithms such as J4.8 and Classification and Regression Trees (CART) [19]. CART is an algorithm that generates a set of rules by splitting data into each child node. Since CART predicts continuous dependent variables (regression) and categorical predictor variables (classification) by building a tree, it is used to perform a classification [20], [21]. It also supports dealing with multiple data types and missing values. Despite the fact that classical PCA has high sensitivity to outliers [22], PCA is often used to extract significant features from network traffic data. In summary, numerous studies have been conducted to design an effective network intrusion detection technique. However, it is important to note that one algorithm or approach cannot detect all existing or unknown attacks precisely due to the existence of anonymity in network traffic patterns.
In the visualization community, researchers apply visualization techniques to address limitations in traditional network intrusion detection analysis. Due to the importance of analyzing large complex network traffic data, the community provides network traffic data as a part of visualization challenge to motivate people to analyze real-world network traffic data visually. For example, the VAST 2012 challenge [23] provided a financial institution’s data for researchers to gain an opportunity to understand the health of global corporate network visually by identifying anomalies or problems. Shiravii et al. [24] conducted a comprehensive review of existing network security visualization systems by classifying them into different five use-case classes: host/server monitoring, internal/external monitoring, port activity, attack patterns, and routing behavior. Although numerous network security visualization systems have been designed, most systems focus only on addressing the issue of how to represent collected log data or network events. To better understand network traffic patterns and detect network intrusions more precisely, visualization techniques should be integrated with computational and machine learning approaches. In the following sections, a detailed explanation is provided on how we combine both computational approaches and visualization techniques in intrusion detection.

III. Approach

In this paper, a publicly available intrusion detection dataset (called NSL-KDD [25]) is used. It contains 148,517 records with 41 attributes (three nominal, six binary, and thirty-two numeric attributes). The dataset includes twenty-four attack types that can be grouped into four major attack categories: DoS, R2L, U2R, and Probe. DoS indicates denial-of-service attack attempting to disable machine or network resources unavailable from remote machines. R2L represents Remote to User attack to gain access to local user accounts by sending packets to a computing machine over the network. Probe is a method of gathering network information to find known vulnerabilities in machines. User to Root Attack (U2R) indicates that an attacker accesses normal user accounts by exploring the system as an administrator. Since the size of data for U2R attack is small (only 119 records), U2R is excluded in the analysis.

MRA utilizes signal processing techniques, so it is capable to discover hidden patterns from the network traffic data. Our approach utilizes DWT to extract features from the data in different resolution levels (γ). DWT is a broadly known time-frequency analysis method. It uses two basis functions such as wavelet function and scaling function [26]. The functions are applied to transform input data into a set of approximation coefficients and detail coefficients. Since a small localized wave in a time domain is analyzed, any sudden or rapid changes in the data can be identified. Thus, DWT is especially useful for analyzing non-stationary signal data (e.g. internet traffic data) [27].

Sliding window analysis is a common approach when examining large network traffic data [28], [29]. The sliding window analysis uses two main parameters: window size and step size. The sliding window size (α) is used to extract feature vectors for analyzing anomalies in the network traffic data. If the size of the window is small, large feature vectors are generated. The step size (β) is considered as a tunable parameter that has a direct impact on identifying the anomalies. Since it indicates the distance between successive windows, if the step size increases, fewer windows are required to analyze the data. When applying DWT to network traffic data, an appropriate wavelet function should be chosen that is closely matched to variations in the input data. Also, choosing appropriate attributes to apply DWT to the network traffic data is critical [30], [31]. A common approach to determine the size of sliding window referencing time information. However, there is no optimal approach of determining the attribute values for analyzing the network traffic data. Therefore, determining optimal values for the sliding window (α), step (β), and level (γ) is important.

In network anomaly detection with DWT, various levels of decomposition are often considered [32]. Depending on the decomposition level (γ), different levels of detail coefficients (detail level 1 ∼ n) and approximate coefficients can be measured. We performed an empirical study with MATLAB software to determine the optimal values for detecting network intrusion more precisely. In DWT, there are various wavelet families proposed by researchers such as Daubechies, Coiflets, Symlets, Discrete Meyer, Biorthogonal, and among others. To identify the best possible wavelet family for network intrusion detection, we evaluate all available wavelet families.

After identifying informative features with DWT, a statistical validation is performed to determine statistically significant features (p < 0.05). If the decomposition level is increased, the chance of having zero-variance features is also increased. Therefore, the statistical validation is useful for removing such unimportant features. Then, PCA is applied to determine principal components of the data. PCA computation is broadly used in feature extraction and exploratory data analysis in network intrusion detection [33], [34]. PCA performs eigenvalue decomposition to determine the variances and coefficients of the data by finding eigenvectors and eigenvalues. First, covariance is measured to determine how much the dimensions vary from the mean with respect to each other. Then, the eigenvectors and eigenvalues are calculated. The eigenvector with the highest eigenvalue is the most dominant principle component in the data, indicating the most significant relationship among the data dimensions. For this reason, PCA is often considered as a dimension reduction method for representing high-dimensional data into a lower dimensional space with the dominant principal components. To determine the principal components, we use Singular Value Decomposition (SVD) [35] because it is good for finding eigenvectors and eigenvalues in non-square matrix such as network traffic data. When representing data into 2D or 3D display space (i.e. coordinate system) with the principal components, confidence interval (θ) should be considered because it indicates the error between original and projected data. For instance, for mapping a high-dimensional data into a lower dimensional
Our visualization tool is designed by following a coordinated multi-views (CMV) framework [36] to support data analysis from different perspectives. It consists of two views - (a) Projection view and (b) Data view (Figure 1). The two views are tightly integrated such that a user's interactions with one view are immediately reflected in all other views. In the Projection view, computed PCA results of the network traffic data are displayed. By default, the 1st and 2nd principal components are used to display the data. A parallel coordinates visualization technique (simply called parallel-coordinates [37]) is used to represent the MRA result (Figure 1b). As the tool is designed to help the user conduct an interactive visual analysis, basic navigation techniques and user interaction techniques are added. For the navigation techniques, semantic zooming and panning are supported in the Projection view, with which the user can review the represented network traffic data items in a closer look (see Figure 1c ∼ 1f). The user interaction techniques include selection and manipulation. If the user is interested in understanding the further detail of data item(s), she is allowed to select the item(s) in the Projection view. Figure 1 shows an example when an R2L attack is selected. The red arrow indicates the selected data item in the Projection view. In the Data view, the corresponding information of the selected item is highlighted in black with representing all attribute values (Figure 1b). The manipulation technique is designed to allow the user to eliminate data item(s). This technique is useful when the user conducts a study of identifying the effectiveness of the data item(s) to PCA computation. For example, if the user is interested in exploring the relationship among attacks, this technique can be used to remove normal network activities.

Our visualization tool also supports identifying statistically similar data items. In the tool, four similarity measurements are supported such as Cosine similarity, Euclidean distance, extended Jaccard coefficient, and Pearson correlation coefficient. Once the user selects data item(s) in the Projection view, a similarity measurement can be performed to find statistically similar data items ($p < 0.05$). This feature is useful when the user conducts a study of comparing relevant data items to previously known (or determined) outliers. For example, if the user finds a possible network anomaly, she may want to validate it by conducting a statistical analysis to identify similar network events. If no similar network events are found, the anomaly can be considered as a possible network anomaly. Lastly, applying an agglomerative hierarchical clustering method with various distance measures is supported.
sliding window size. However, since it is not the primary objective of this study, we leave this for our future work.

When applying DWT with different wavelet functions and levels, different amounts of data records and attributes are generated. For instance, when applying Daubechies 3 (i.e. db3) with \( \alpha = 150, \beta = 50, \) and \( \gamma = 3 \), total of 2,958 data records with 703 attributes are generated. Out of 703 attributes, 418 (59.5%) are selected as statistically significant attributes \((p < 0.01)\). However, when using Discrete Meyer with \( \alpha = 50, \beta = 20, \) and \( \gamma = 3 \), total of 7,417 data records with 3,478 attributes are generated. And, 2,992 attributes (86%) are determined as significant features \((p < 0.01)\). The significant features are considered as dominant features that can be used to detect intrusions. As discussed in Section III, if the size of the window \((\alpha)\) is small, large feature vectors are generated. The sliding window size can be adjusted to speed-up the computation process. However, it may compromise the overall accuracy of detecting intrusions if improperly set. It is also important to note that if the step size \((\beta)\) is small, more windows are required to analyze the data.

With the original network traffic data, it is difficult to determine normal vs. attack activities. As we discussed, MRA determines significant features from the data. However, it is still not clear what DWT wavelet families produce more significant features to detect intrusions. Therefore, we tested most wavelet families under the same experimental condition (i.e. \( \alpha = 150, \beta = 50, \) and \( \gamma = 3 \)). Figure 3a shows a PCA projection of the original data. As can be seen, different network activities appear in all over the place which makes it difficult to separate them clearly. On the other hand, when applying different wavelet families, a somewhat clear separation between normal and attack activities is appeared by forming unique clusters (see Figure 3b ~ 3i). Biorthogonal (Figure 3f and 3g) shows that some Probe attacks appear in the DoS attack cluster. Discrete Meyer (Figure 3h) and Symlets (Figure 3i) are suitable for separating DoS and Probe attacks from R2L attack. But, R2L attack is appeared near to the normal network activity cluster (see the regions in left-bottom in Figure 3h and 3i). Another interesting observation is that the results of Coiflets 1 and Daubechies 3 are similar to each other. The reason might be because Coiflets is constructed from Daubechies with a high number of vanishing moments in the scaling and the wavelet functions [39].

Huang et al. [31] found that Coiflet and Mexican Hat wavelets are good for detecting anomalies when using a five-minute, sixty-sample window. However, we identified that Daubechies 3 is the best-suited wavelet for detecting network intrusions since it creates well-separated clusters (see Figure 3c). Huang’s experiment was well formatted, but they used only the first and second coefficients based on the assumption that these two coefficients have sufficient information. As they commented, any larger coefficients might contain too sparse information. However, from our empirical study, we found that the coefficient \((\gamma = 3)\) is better for extracting significant information to detect intrusions.

Table I shows the results when the hierarchical clustering method is applied to three different datasets: wavelet features (3,478 attributes), two principal components (2 attributes), and three principal components (3 attributes). The results are generated with the DWT features \((\alpha = 150, \beta = 50, \) and \( \gamma = 3 \)). Solid connected lines indicate the clustering results. Since the wavelet features are the statistically validated DWT features, good clustering results are generated. However, we found that a large number of attributes often require a significantly large computational time which is not appropriate for the real-time intrusion detection. To support the real-time intrusion detection, PCA is considered because it has a benefit of reducing the total number of attributes (i.e. dimensions). As mentioned above, the confidence interval \((\theta)\) is often considered to determine major PCA components. However, minor PCA components (i.e. after removing the components that have higher eigenvalues) are also essential for revealing anomalies in network traffic data [40]. Because of this reason, allowing the user to select different PCA components is necessary. Our visualization tool has the feature of selecting the best possible PCA components to detect intrusions.

From the hierarchical clustering results, clustering accuracy is measured as 0.86 ± 0.16 (for using all wavelet features), 0.64 ± 0.04 (for using two principal components), and 0.83 ± 0.18 (for using three principal components). When comparing the results of the two vs. three principal components, the three principal components produced a better clustering result. Moreover, the three principal components and the wavelet features showed similar clustering accuracy. We also found that...
TABLE I: Results of the detected $k$ clusters with different distance metric (Euclidean distance ($L^2$), Chebyshev distance ($L^\infty$), City-block distance ($L^1$) and Pearson correlation coefficient ($R^2$)). The cluster are represented as solid connected lines.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Wavelet Features</th>
<th>Two Principal Components</th>
<th>Three Principal Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L^2$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L^\infty$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L^1$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R^2$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

the three principal components outperform the other two when Chebyshev distance ($L^\infty$) is applied. Interestingly, Chebyshev distance ($L^\infty$) was not good for creating clusters with the wavelet features. Moreover, Pearson correlation coefficient ($R^2$) did not work well for creating clusters with the two and three principal components.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we introduced a new approach to analyze network traffic data for intrusion detections. Although various approaches have been proposed by incorporating machine learning algorithms, most methods still suffer from detecting unknown attacks. To address the limitation, we proposed a hybrid approach that integrates computational analysis and visual analytics. The computational analysis is used to extract significant features from the data. For visual analytics, an interactive visualization tool is designed to display the analyzed network traffic features and provide user interaction techniques to support an interactive visual analysis on the visually represented data items. To determine best suitable parameters for applying DWT on the network traffic data, an empirical study was conducted. To show the effectiveness of our approach, the hierarchical cluster method was applied to identify clusters. Although the results indicate that our approach has a strength, it is still important to conduct a formal evaluation study to determine the effectiveness of all possible input and output parameters for DWT and PCA. For future works, we plan to extend our study to test all possible input parameters and measure sensitivity and specificity in detecting network intrusions. Also, a comparative study with known intrusion detection models will be conducted to determine the benefits and limitations of our approach. Since our approach requires less amount of computation time for detecting intru-
sions, a real-time intrusion detection system will be designed and tested in a real network environment.

ACKNOWLEDGMENT

This work was supported by the U.S. Army Research Office (ARO) - Grant No. W911NF-13-1-0143.

REFERENCES