World Academy of Science, Engineering and Technology
International Journal of Information and Communication Engineering
Vol:6, No:6, 2012

Designing a Framework for Network Security Protection

Eric P. Jiang

***Abstract*—**As the Internet continues to grow at a rapid pace as the primary medium for communications and commerce and as telecommunication networks and systems continue to expand their global reach, digital information has become the most popular and important information resource and our dependence upon the underlying cyber infrastructure has been increasing significantly. Unfortunately, as our dependency has grown, so has the threat to the cyber infrastructure from spammers, attackers and criminal enterprises. In this paper, we propose a new machine learning based network intrusion detection framework for cyber security. The detection process of the framework consists of two stages: model construction and intrusion detection. In the model construction stage, a semi-supervised machine learning algorithm is applied to a collected set of network audit data to generate a profile of normal network behavior and in the intrusion detection stage, input network events are analyzed and compared with the patterns gathered in the profile, and some of them are then flagged as anomalies should these events are sufficiently far from the expected normal behavior. The proposed framework is particularly applicable to the situations where there is only a small amount of labeled network training data available, which is very typical in real world network environments.

***Keywords*—**classification, data analysis and mining, network intrusion detection, semi-supervised learning.

I. Introduction

WITH the rapid growth of the Internet and other telecommunication networks and information systems, digital information has become the most valuable asset of many organizations, and our dependency upon the underlying cyber infrastructure has been increasing significantly. Unfortunately, as our dependency has grown so has the threat to the cyber infrastructure from spammers, attackers and criminal enterprises. Cyber infrastructure incorporates a diverse array of technologies, including distributed computing systems, networks, storage and supportive software services, and provides its users and customers with access to share the computing and storage resources and to conduct various business services. The growing accessibility of information, computing and service resources and the lack of security as a core element in the initial design of the infrastructure have made networks and information systems increasingly vulnerable to continuous and innovative intrusions and attacks.

It is very critical for us to protect nearly every aspect of cyber infrastructure [1]. Network breaches such as worms, viruses and spam cost the global economy billions of dollars every year in lost productivity. For instance, the disclosure of business data caused by intrusions can lead to huge financial loss through Internet transactions and other e-commerce services. All cyber intrusions and attacks have the potential for a devastating large-scale network failure, service interruption or the total unavailability of service [2].

Over the years, various network security techniques and systems have been developed and employed to help secure cyber infrastructure against intentional and potentially malicious threats. Conventional cyber security approaches are the mechanisms designed for firewall, authentication tools and network servers and are used to monitor and potentially block viruses and to protect user's private information from spyware and malware. Predominantly, they are signature based and detect known attacks by utilizing the signatures of the attacks. However, as cyber threats are dynamically and constantly evolving, the techniques for detecting known attacks are not enough to protect users and networks. Higher-level and effective methodologies are also required to detect all types of malicious network traffic and computer usage so that a more secured cyber infrastructure can be realized. One representative in this technology category is anomaly detection systems. An anomaly detection system (ADS) applies various learning algorithms to profile the normal network behavior, which enables it to be effective in finding both known and unknown intrusions and attacks. It is a dynamic monitoring entity that complements the static monitoring abilities of a firewall [3].

In this paper we propose a new network intrusion detection framework for cyber security and it integrates a semi-learning algorithm into the process of modeling normal behavior. As an intrusion detection system, the framework is based on the premise that any intrusive activity is a subset of anomalous activity, and the goal of the framework is to detect anomalous network events that behave significantly from the established normal behavior profile.

More specifically, the framework consists of two major modules: system training and system detection. In the system training module, the semi-learning algorithm is utilized to produce a profile of normal patterns in the absence of an attack. In the system detection module, new input network data are collected, analyzed and compared with the patterns in the profile, and then are possibly flagged as anomalies if the events, represented by the data, deviate sufficiently from the expected normal behavior.

E. P. Jiang is a Professor of Computer Science & Mathematics with the University of San Diego, San Diego, California 92110, USA (e-mail: jiang@ sandiego.edu).

II. Design of the framework

A. Overview

There is a host of technological challenges in developing a detection system that is capable of accurately identifying malicious network intrusive events. One of the key challenges is that the large quantity of network data with high-dimensional features can be very difficult to analyze and model. Another challenge lies in reducing intrusion false alarm rate. In general there are a lot more network data of normal activity available for system learning than those of anomalous activity. This unbalanced distribution on network training data could lead to a biased detection system that learns towards more to the network normal behavior, resulting a high false alarm rate and hindering detection accuracy. These problems need to be carefully considered and addressed when designing a detection framework, selecting adequate and efficient machine learning algorithms and applying them in data processing and pattern discovering.

Machine learning technology plays key roles in building the normal profile in anomaly detection systems. In many real-world network environments, large amounts of unlabeled audit data are abundantly available, while labeled network data (in particular the data that represent anomalous behavior) are usually limited in supply. Successful supervised machine learning methods[1] generally require a sufficient amount of training data, and when they are applied to anomaly detection systems, it represents a requirement of labeled network data for both normal and anomalous behaviors. In addition, since patterns of normal activity of a network can evolve with the changing network environments or services, the difference between the training and actual data can lead to a high misclassification rate of normal network activity.

One possible solution for addressing these problems is to use unsupervised methods[2] that take unlabeled data as input and aim to group similar data and discover malicious patterns even without having prior knowledge about training data labels [4]. As with supervised learning methods, unsupervised anomaly detection solutions have their own drawbacks. They heavily rely on the assumption that, when being projected into a high dimensional space, all unlabeled training data from normal and anomalous classes are similar in their respective identity groups and are significantly different between the classes. Unfortunately, this assumption may or may not be strictly held in practice. Moreover, due to the algorithmic nature of unsupervised learning and the fact that among the training data, there is only a very small proportion belongs to the anomalous activity class, unsupervised methods can have a low intrusion detection rate and their performance can also be training data dependent.

In this paper, we present a new semi-supervised learning framework for network anomaly detection. In the last few years, there has been surging interest in developing semi-supervised learning models, which can be considered a hybrid approach of supervised and unsupervised models and are capable of discovering patterns from both labeled training samples and additional pertinent unlabeled data. The semi-supervised learning paradigm has been successfully applied in document classification and many other areas [5].

We believe that semi-supervised learning is adequately applicable to network intrusion detection problems where labeled anomalous events (and perhaps also labeled normal events) are usually very limited in quantity while unlabeled pertinent event data are abundantly available. Specifically for the proposed intrusion detection framework and given a small set of labeled event samples and a large set of unlabeled event data, we first use a cluster analysis method[3] on the labeled samples to formulate some initial clusters or groups of data instances, and then apply the formulated clusters to classify the available unlabeled data. We run this process in an iterative fashion to refine the clusters until a stable profile of normal network profile is generated, which is comprised of some of the cluster representatives.

In comparison to unsupervised approaches, using both labeled and unlabeled data in this semi-supervised learning framework should help enhance accuracy of the profile and subsequently improve its intrusion detection rate. Furthermore, in order to fully utilize the predictive values of labeled data and to adequately adjust the influence of unlabeled data in learning, a weighting scheme is applied in the formation of cluster representatives, which aims to place more weights on labeled data than unlabeled. Once the framework has been trained, we deploy it to input network data and use it to perform detection of possible intrusive events. Since the framework detects anomalies based on the profile of normal activity, it is capable of detecting previously unknown intrusions and attacks. The clustering based semi-supervised learning approach for the proposed framework represents a key difference from other network intrusion detection systems.

B. Detailed Description

We outline the structure as well as major components of the framework in Fig. 1. The framework consists of two modules: normal behavior profile builder and anomalous activity detector. The profile builder is responsible for collecting and preprocessing accumulated network training data that reflect both normal behavior and anomalous activities, and for applying a semi-supervised learning algorithm on the training data to build an accurate network normal behavior profile. Therefore, this module contains the top four components in the diagram shown in Fig.1.

[1] Supervised machine learning methods infer a function from labeled training data. Each labeled training example is a pair consisting of an input object and a desired output label or value. A supervised learning method analyzes the training data and produces an inferring function that should predict the correct output value for any valid input object.

[2] Unsupervised machine learning methods intend to find hidden structure or patterns from unlabeled training data. This is different from supervised learning and all training data do not carry any output labels or values.

[3] Cluster analysis is a a popular machine learning method for partitioning data objects into meaningful clusters so that objects within a cluster have similar characteristics but are dissimilar to objects in other separate clusters.

For the data collection component, various data capturing tools such as Libpscap (Linux) or Winpcap (Windows) are used to capture and gather data traveling over networks. Once the data are collected, they are preprocessed through some procedures for data reduction, feature selection and scaling, where a portion of the data records and (less important) data features are eliminated from learning while a number of other features are scaled to some more reasonable value ranges to help facilitate model building. In this step, some of the collected network data may be also manually examined by the system security administrator and are labeled as normal or anomalous events to formulate a small set of labeled training data. After the step of data preprocessing, a semi-supervised learning algorithm is applied to the labeled training data and additional collected unlabeled network data to create a number of representatives of normal and anomalous activity. These representatives or patterns in the data are then used to form the normal behavior profile of the network. The second module of the framework, anomalous activity detector, is responsible for monitoring and determining if any new input network events are suspicious as intrusions or attacks. In this part, the corresponding event records are collected, and transformed into the same data feature space, which is constructed by the profile module, to be compared in real-time with the normal activity patterns in the profile, and then may be labeled as anomalies if they do not conform to the expected normal behavior.

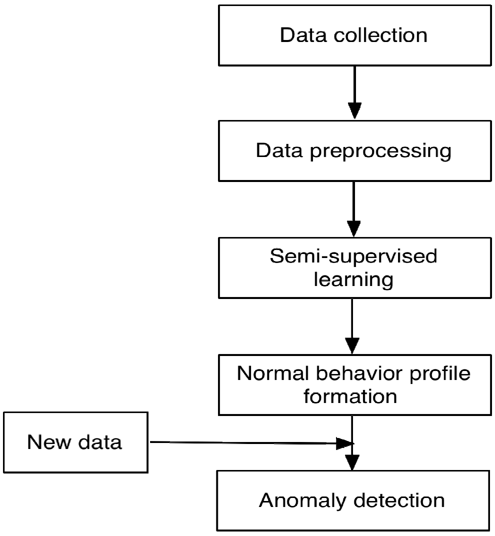

Fig. 1 Structure and major components of the semi-supervised network intrusion detection framework

There are some system thresholds used in the framework to determine if a network event is anomalous enough to warrant a security alert. These decision parameters can be further calibrated or fine-tuned by user specified criteria such as system false alarm cost and security tolerance level.

III. Semi-Supervised Learning algorithm

In this section, we describe a clustering based semi-supervised learning algorithm, which is the core component of the framework. We first introduce the classical k-means algorithm and then extend it, together with the Expectation-Maximization iteration process, to a semi-supervised learning algorithm.

A. K-Means Clustering Algorithm

K-means [6] is a simple and well-known unsupervised clustering algorithm; it attempts to partition a given set of data objects into a user-specified number of clusters (i.e., *k*), which are represented by their respective cluster centers or centroids. Suppose we have a set of network audit records $R = \{r_1, r_2, \ldots, r_n\}$ with classes $l = \{l_N, l_A\}$, where l_N and l_A denote the class of normal activity and the class of anomalous activity, respectively. Then the *k*-means algorithm can be used to group the records into a number of clusters where the data records within a cluster are more similar to each other than records belong to different clusters. Specifically, we first determine the parameter k ($k \geq 2$), the number of clusters desired, and then select k records at random (but at least one record from each of the classes L should be selected) as the initial cluster centroids. Then, we assign each record in R to its closest centroid according to some similarity metric such as Euclidean distance and use these assigned records to form individual clusters. Once the records that belong to a cluster C_j are identified, its cluster centroid c_j is updated by

$$c_j = \frac{1}{size(C_j)} \sum_{r_i \in C_j} r_i \qquad (1)$$

The process of assigning data records to clusters and updating their respective cluster centroids are repeated until all clusters become stable or there are only very small changes in the centroids in two consecutive iterations. The *k*-means clustering algorithm is simple in concept and is also - particularly efficient in processing large data sets.

B. A Basic K-Means Based Classification Algorithm

For a given set of labeled training data such as network records, the *k*-means algorithm can be used as a classification tool. With some adequate selection of data features and proper setting of the number of clusters as well as initial centroids, the formulated cluster centroids by *k*-means can effectively represent the content of the data and subsequently, they can be applied to previously unseen data for classification. A basic *k*-means based classification algorithm is summarized in Table I.

TABLE I
A Basic K-means Based Classification Algorithm

```
Step 1.  Select the number of clusters and initial cluster
         centroids
Step 2.  For each of labeled training data records
             Find the closest cluster of the same class label
             Assign the record to the cluster
Step 3.  For each of the formulated clusters
             Trim its outliers
             Update its cluster centroids by using (1)
```

The supervised classification approach presented in Table I can be extended into a semi-supervised algorithm that learns for classification from both labeled and unlabeled training data. This semi-supervised algorithm is used in the proposed network intrusion detection framework. A special feature of the algorithm is that it requires only a small number of labeled training samples.

C. A K-Means Based Semi-Supervised Classification Algorithm

Unsupervised learning algorithms such as clustering do not require any labeled data in training. But when any pertinent labeled samples become available, they can be integrated with unlabeled data and can generally help improve the learning process. The integration of these two types of data can be accomplished by using the Expectation-Maximization (EM) algorithm. EM is usually used to iteratively estimate the maximum likelihood of hidden parameters for problems with incomplete data [7]. If we regard the class labels of unlabeled data as unknown values, EM can then be applied to estimate these labels.

Specifically, the cluster refining process that combines *k*-means and EM can be operated on a training set that is mixed with labeled and unlabeled data. The process starts with a number of initial clusters that are constructed by only labeled samples in the training set. The corresponding computed cluster centroids are then used to classify unlabeled data in the set. These newly classified unlabeled data are blended with the originally clustered labeled samples to form a new set of expanded clusters. The centroids of these expanded clusters are then updated by using (1). Through the EM iterations, this clustering process is repeated until all the clusters are stabilized.

Furthermore, in order to deal with the situations where there are only a very limited number of labeled network training data, we can impose appropriate weights on labeled and unlabeled data to modulate their influence in cluster formation. Since, for training the proposed framework, the quantity of unlabeled data can be significantly larger than that of labeled, unlabeled data can potentially play a dominant role in computing and updating cluster centroids. In general, when the natural clusters of the combined (labeled and unlabeled) data are in correspondence with class labels, the semi-supervised learning process described above would produce the clusters that are helpful for classification.

However, when the natural clustering of the data generates cluster centroids that are not in correspondence with class labels, then these centroids would likely be destructive to classification accuracy.

Note that when (1) is used for updating cluster centroids, labeled and labeled data are not treated differently in terms of their contributions towards centroid computation. This approach may not be adequate for the following two reasons. First, in comparison to labeled training samples, there are many order of magnitude more unlabeled data used in training. The overwhelming quantity of unlabeled data may incline to produce undesirable clusters. Secondly, since unlabeled data do not have class labels, they should generally carry less predictive values in determining cluster centroids. Therefore, we suggest using an improved weighted formula in place of (1) for updating cluster centroids. Assume, for a cluster C_j, L_j and U_j are its labeled and unlabeled data sets, respectively, and β is a weighting parameter with $0 \leq \beta \leq 1$, the corresponding centroid is updated by

$$c_j = \frac{1}{size(L_j) + \beta \cdot size(U_j)} \left(\sum_{r_i \in L_j} r_i + \beta \sum_{r_k \in U_j} r_k \right) \tag{2}$$

Equation (2) can be considered an extension of (1). When β takes a small value that is close to zero, then the centroid c_j is updated primarily using the labeled samples. In the extreme case that β is set to zero, this entire classification process shall reduce to the basic (supervised) *k*-means based algorithm shown in Table I. On the other hand, when the parameter β takes a large value away from zero, then (2) indicates that the unlabeled data play a certain role in the computation of centroids, and in the case that β takes the value of one, each unlabeled record shall have the same weight as labeled training records and the process effectively becomes the traditional *k*-means algorithm with both labeled and unlabeled data being used in training. A *k*-means based semi-supervised classification algorithm that incorporates all aforementioned strategies is summarized in Table II. This algorithm serves as the core component in our proposed network intrusion detection framework.

TABLE II
A K-MEANS BASED SEMI-SUPERVISED CLASSIFICATION ALGORITHM

```
Step 1.  Select the number of clusters and initial cluster
         centroids (based on only labeled data)
Step 2.  For each of labeled training records
             Find the closest cluster
             Assign the record to the cluster
Step 3.  For each of formulated clusters
             Trim its outliers
             Update its cluster centroids by using (1)
Step 4.  For each of unlabeled training records
             Find the closest cluster
             Assign the record to the cluster
Step 5.  Update cluster centroids by using (2)
Step 6.  If clusters are stabilized, then stop; otherwise
         repeat Step 4 – Step 5
```

IV. CONCLUSION

In the paper we have proposed a new network intrusion detection framework for cyber security. It is based on a semi-supervised machine learning method, which combines the well-known *k*-means and EM algorithms, can learn for building a profile of normal network behavior and subsequently for detecting various network intrusions. The proposed framework has a unique feature - requiring only a small set of labeled training data and therefore it is particularly applicable to the situations where the vast majority of available network training data are unlabeled.

As future work, we plan to implement the framework according to the design and methodology presented in this paper. In addition, we plan to conduct various experiments and perform an extensive empirical analysis of the framework with several popularly used network security datasets such as KDD-CUP network intrusion data [8].

REFERENCES

[1] Homeland Security Council of USA, "National strategy for homeland security," 2007.

[2] T. N. Saadawi, and L. H. Jordan, *Cyber Infrastructure Protection*. Strategic Studies Institute, US Army War College, 2011.

[3] A. Patcha and J. Park, "An overview of anomaly detection technologies: exisiting solutions and latest technological data," *Computer Networks,* vol. 51(12), 2007, pp. 3448–3470.\

[4] E. Eskin, et.al. *A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data*. Application of Data Mining in Computer Security (eds. S. Jajodia and B. Dordrecht), Kluwer, 2002, ch. 4.

[5] E. Jiang. *Automatic Text Classification from Labeled and Unlabeled Data.* A chapter to be appears in Intelligent Data Analysis for Real-Life Applications: Theory and Practice (eds. R. Magdalena, et. al.), IGI Global Publishing, 2012.

[6] J, MacQueen, "Some methods for classification and analysis of multivariate observations," in *1967 Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press,* pp. 281–297.

[7] A. Dempster, N. Laird and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Statistical Society, Series B, 39,* pp. 1–38, 1977.

[8] KDD Cup, *The International Knowledge Discovery and Data Mining Tools Competition KDD-CUP*. http://kdd.ics.uci.edu/datasets/kddcup99, 1999.