Unsupervised Anomaly Detection with Unlabeled Data Using Clustering

[1]Witcha Chimphlee, [2]Prof.Dr.Abdul Hanan Abdullah, [3]Associate Prof Dr.Mohd Noor Md Sap
Faculty of Computer Science and Information Systems
University Technology of Malaysia,
81310 Skudai, Johor, Malaysia
Tel: (607)-5536535, Fax: (607) 5565044
Email: [1]witcha_chi@yahoo.com, [2]hanan@fsksm.utm.my, [3]mohdnoor@fsksm.utm.my

ABSTRACT

Intrusions pose a serious security risk in a network environment. New intrusion types, of which detection systems are unaware, are the most difficult to detect. The amount of available network audit data instances is usually large; human labeling is tedious, time-consuming, and expensive. Traditional anomaly detection algorithms require a set of purely normal data from which they train their model. We present a clustering-based intrusion detection algorithm, *unsupervised anomaly detection,* which trains on unlabeled data in order to detect new intrusions. Our method is able to detect many different types of intrusions, while maintaining a low false positive rate as verified over the Knowledge Discovery and Data Mining - KDD CUP 1999 dataset.

KEYWORDS

Computer security, Anomaly detection, Unsupervised clustering, Outliers, Unlabeled data

1. Introduction

As defined in [1], intrusion detection is "the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions, defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network". In figure 1 depicted intrusion detection taxonomy. Intrusion detection is to build a system which would automatically scan network activity and detect such intrusion attacks.

Anomaly detection has been an active field of intrusion detection research since it was originally proposed by Denning [2].

Unsupervised Anomaly Detection (UAD) algorithms have the major advantage of being able to process unlabeled data and detect intrusions that otherwise could not be detected.

The goal of data clustering, or unsupervised learning, is to discovery a "natural" grouping in a set of patterns, points, or objects, without knowledge of any class labels.

We need a technique for detecting intrusions when our training data is unlabeled, as well as for detecting new and unknown types of intrusions.

There are generally two types of attacks in network intrusion detection: the attacks that involve single connections and the attacks that involve multiple connections (bursts of connections) [3] [4]. In misuse detection, each instance in a data set is labeled as 'normal' or 'intrusion' and a learning algorithm are trained over the labeled data. A key advantage of misuse detection techniques is their high degree of accuracy in detecting known attacks and their variations. Their obvious drawback is the inability to detect attacks whose instances have not yet been observed. [4] Misuse detection can only detect attacks that are well known and for which signatures have been written.

An anomaly detection technique builds models of normal behavior, and automatically detects any deviation from it, flagging the latter as suspect. A potential drawback of these techniques is the rate of false alarms [4].

Attacks fall into four main categories [8]:

- DoS : denial-of-service, for example ping-of-death, teardrop, smurf, SYN flood, etc.,
- R2L : unauthorized access from a remote machine, for example guessing password,
- U2R : unauthorized access to local super user (root) privileges, for example, various "buffer overflow" attacks,
- PROBING: surveillance and other probing, for

example, port-scan, ping-sweep, etc. Some of the attacks (e.g. denial of service (DoS), probing) may use hundreds of network packets or connections, while on the other hand attacks like U2R and R2L typically use only one or a few connections.[3]

Denial-of-service attacks

Denial of Service (DoS) is a class of attacks where an attacker makes some computing or memory resource too busy or too full to handle legitimate requests. A denial of service attack can be argued to have a distinct set of features and patterns that manifest themselves when examine packets on the network[5]

Remote to User Attacks

A remote to user (R2L) attack is class of attacks where an attacker sends packets to a machine over a network, then exploits machine's vulnerability to illegally gain local access a user.

User to Root Attacks

User to root exploits is a class of attacks where an attacker starts out with access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system. Most common exploits in this class of attacks are regular buffer overflows, which are caused by regular programming mistakes and environment assumptions.

Probing

Probing is a class of attacks where an attacker scans a network to gather information or find known vulnerabilities. An attacker with a map of machines and services that are available on a network can use the information to look for exploits.

2. The Our approach

2.1 Unsupervised learning

Unsupervised learning methods analyze each event to determine how similar (or dissimilar) depends on the choice of similarity measures, dimension weighting. The interesting feature of clustering is the possibility to learn without knowledge of attack classes, thereby reducing training data requirement, and possibly making clustering based techniques more viable than classification-based techniques in a real world setting.

Methods for unsupervised anomaly detection do not assume that the data is labeled or somehow otherwise sorted according to classification [5]. Problem of unsupervised intrusion detection [6]. First, they modified the data significantly by limiting the number of attacks to 1 ~ 1.5 % of the complete training dataset so that their hypothetical assumption is true. Second, each cluster is self-labeled as attacks or normal, based purely on the number of instances in it. Finally, the idea of detecting intrusions in a new dataset using the self-labeled clusters of the training dataset seems misguided. Certain attacks, such as Denial of Service and scanning can produce large amounts of attack data. Those two cases falsify the assumption of unsupervised anomaly detection and need to be handled separately [7].

2.2 Clustering

An ideal case is to group related data (measured by a distance function) into the same cluster and unrelated data into different clusters. More efficient algorithms like K-Means and EM, which have linear cost per iteration, also need scale-up before they can be applied to very large data sets. An example of clustering is depicted in Figure 2. The input patterns are shown in Figure 2(a) and the desired clusters are shown in Figure 2(b).

3 UAD model

Unsupervised Anomaly Detection (UAD) model have 6 steps:

3.1 Dataset Description

The DARPA 99 is various intrusions simulated in a military network environment. It has 4,900,000 data instances and connection is a sequence of TCP packets to and from some IP addresses. Basic features of individual TCP connections are *duration, protocol type, number of bytes transferred, the flag indicating the normal or error status.* Content features within a connection suggested by domain knowledge such as the number of file creation operations, number of failed login attempts. Traffic features computed using a two-second time window are the number of connections to the same host, percent of connections that have "SYN" and "REJ" errors.

3.2 Normalization

Convert the data instances to a *standard form* based on the training dataset's *distribution*

$$\text{new\_instance}[j] = \frac{\text{instance}[j] - \text{avg\_vector}[j]}{\text{std\_vector}[j]}$$

$$\text{avg\_vector}[j] = 1/N \sum \text{instance}[j]$$

std_vector[j] = {1/(N-1) ∑(instance[j] – avg_vector[j]2)}1/2
vector[j] : the j th element(feature) of the vector

3.3 Metric

- Finding or constructing an appropriate metric is critical to the performance of the method
- Experiment with several weighted metrics
- In the end, use a metric with equally weighted features. Some increase in performance from weighted metrics is not a significant amount and undermine the system's generality.

3.4 Clustering

- Variant of single-linkage clustering
- Assume we have fixed a **metric M**, and a constant **cluster width W**.
- Let *dist(C,d)*, where C is a cluster and d is an instance, be the distance under the metric M, *between C's defining instance (as the centroid) and d*

1. Initialize the set of clusters, S, to the empty set
2. Obtain a *data instance (feature vector) d* from the training set
   - 2.1 If S is empty, then create a cluster with d as the defining instance, and add it to S.
   - 2.2 Otherwise, find the cluster *C* in *S*, such that for all C_1 in S, *dist(C,d) <= dist(C_1,d)*
3. If *dist(C,d)* <= W, then associate *d* with the cluster *C*. Otherwise, new cluster must be created for it : S ← S ∪ {C*n*}, *Cn* is a cluster with d as its defining instance
4. Repeat steps 2 and 3, until no instances are left in the training set

3.5 **Labeling clusters**

Some percentage N of the clusters containing *the largest number of instances* associate with them as '*normal'*.

3.6 Detection

- Convert d based on the statistical information of the training set form which the clusters were created. Let d′ be the instance after conversion.
- Find a cluster C which is closest to d′ under the metric M.
- Classify d′ according to the label of C (normal or anomalous).

4. Conclusion

Standard measures for evaluating IDSs are first, *detection rate* (i.e. how many attacks we detected correctly). second, the *false alarm rate* (i.e. how many of normal connections we incorrectly detected correctly). Third, *trade-off between detection rate and false alarm rate.* Fouth,*performance*(Processing speed+propagation+reaction). Finally, *Fault Tolerance* (resistant to attacks, recovery, resist subversion). Detection rate is computed as the radio between the number of correctly detected attacks and the total number of attacks, while false alarm (false positive) rate is computed as the ratio between the numbers of normal connections that are incorrectly misclassified as attacks [3]. Clustering is suitable for anomaly detection, since no knowledge of the attack classes is needed whilst training [7]. Standard metrics that were developed for evaluating network intrusions usually correspond to detection rate as well as false alarm rate (Table 1).

Table1.Standard metrics for evaluations of intrusions (attacks)

Confusion matrix (Standard metrics)		Predicted connection label	
		Normal	Intrusion (Attacks)
Actual connection label	Normal	True Negative(TN)	False Alarm (FP)
	Intrusions (Attacks)	False Negative(FN)	Correctly detected attacks (TP)

From Table 1, recall, precision and F-value may be defines as follows :

Precision = TP / (TP + FP)
Recall = TP / (TP + FN)

$$F - value = \frac{(1+\beta^2).\text{Recall}.\text{Precision}}{\beta^2.\text{Recall}+\text{Precision}}$$

Where β corresponds to relative importance of precision vs. recall and it is usually set to 1 [9].

ROC Curves (in Figure 3.) is a trade-off between detection rate and false alarm rate. It is plot for different false alarm rates. Ideal system should have 100% detection rate with 0% false alarm

References

[1] R.Bace and P.Mell. "Intrusion Detection Systems". *NIST Special Publications SP 800-31.*November. 2001.
[2] D.E. Denning: An intrusion detection model. IEEE Transactions on Software Engineering, SE-13: 222-232, 1987.
[3] Aleksandar Lazarevic, Aysel Ozgur, Levent

Ertoz, Jaideep Srivastava, and Vipin Kumar. A comparative study of anomaly detection schemes in network intrusion detection. In SIAM International Conference on Data Mining (2003).

[4] Dokas, P., Ertoz, L., Kumar, V., Lazarevic, A., Srivastava, J., Tan, P.: Data Mining for Network Intrusion Detection, Proc. NSF Workshop on Next Generation Data Mining, Baltimore, MD, November 2002.

[5] Leonid Portnoy, Eleazar Eskin and Salvatore J. Stolfo. ``Intrusion detection with unlabeled data using clustering" Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001). Philadelphia, PA: November 5-8, 2001.

[6] Shi Zong, Taghi Khoshgoftaar, and Naeem Seliya, Evaluating Clustering Techniques for Network Intrusion Detection. International Journal of Reliability, Quality, and Safety Engineering. 2005. In press.

[7] K. Burbeck and S. Nadjm-Tehrani, ADWICE: Anomaly Detection with Real-time Incremental Clustering, in Proceedings of 7th International Conference on Information Security and Cryptology (ICISC 04), Springer Verlag, December 2004.

[8] Intrusion Detection Attacks Database, http://www.cs.fit.edu/~mmahoney/ids.html

[9] Paul Dokas, Eric Eilertson, Levent Ertoz, Yongdae Kim, Aleksandar Lazarevic, Jaideep Srivastava, Vipin Kumar, Pang-Ning Tan, Zhi-li Zhang: Data Mining for Network Intrusion Detection, Digital Technology Center, University of Minnesota, March 26, 2003.

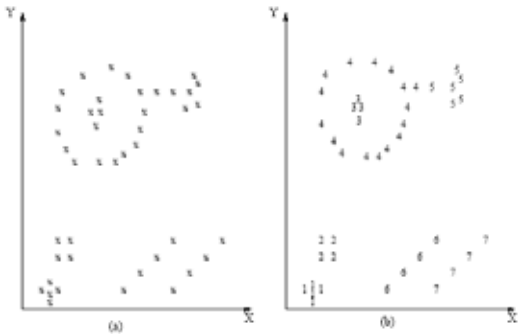

Figure 2. Data Clustering

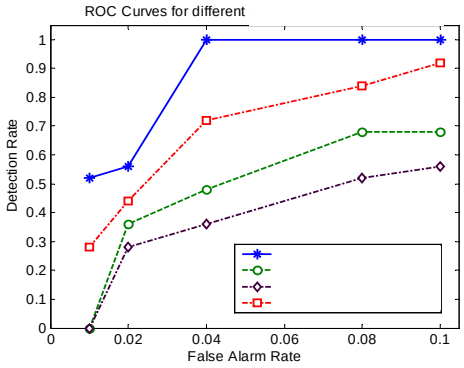

Figure 3 ROC

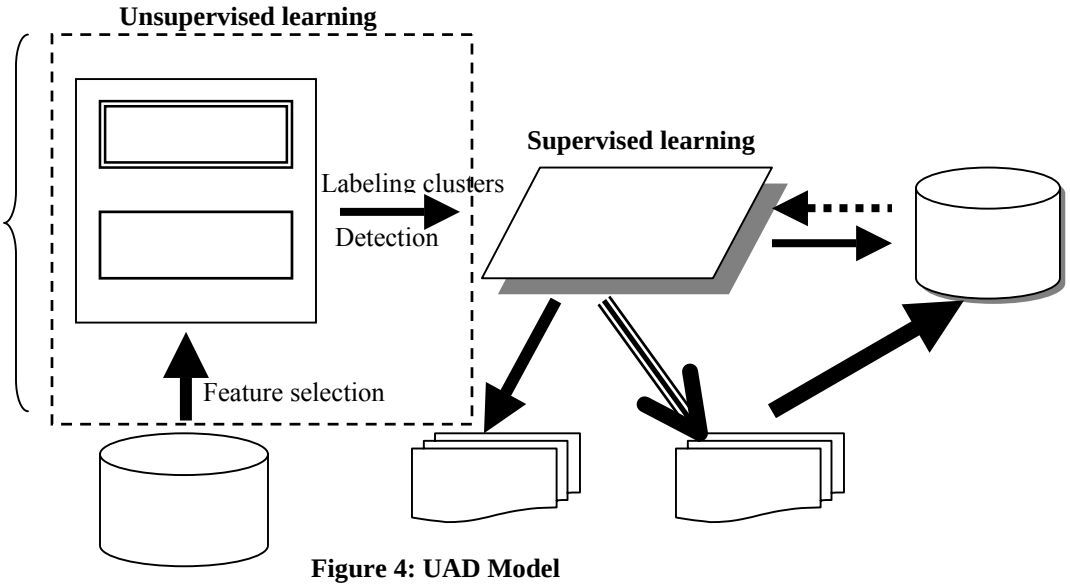

Figure 4: UAD Model

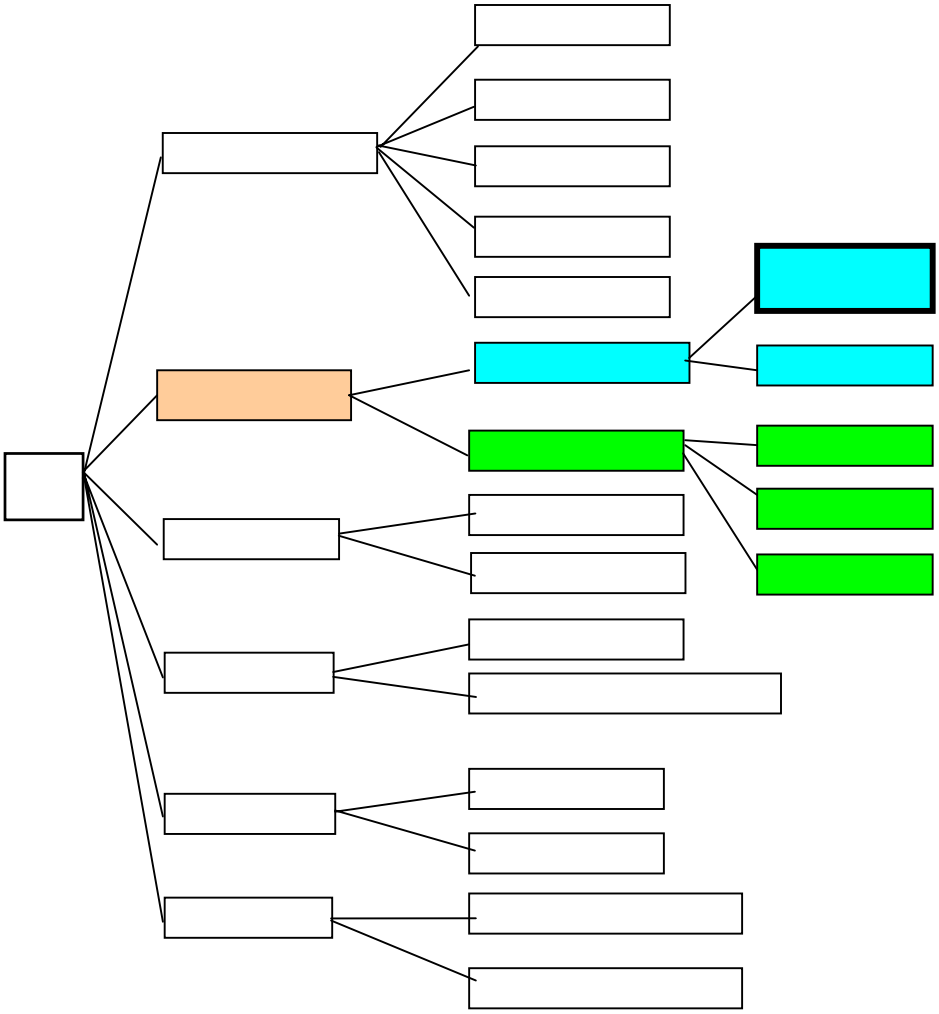

Figure 1. Intrusion Detection Taxonomy