Combining Heterogeneous Classifiers for Network Intrusion Detection

Ali Borji

School of Cognitive Sciences,
Institute for Studies in Theoretical Physics and Mathematics,
Niavaran Bldg. P.O.Box 19395-5746, Tehran, IRAN
borji@ipm.ir

Abstract. Extensive use of computer networks and online electronic data and high demand for security has called for reliable intrusion detection systems. A repertoire of different classifiers has been proposed for this problem over last decade. In this paper we propose a combining classification approach for intrusion detection. Outputs of four base classifiers ANN, SVM, *k*NN and decision trees are fused using three combination strategies: majority voting, Bayesian averaging and a belief measure. Our results support the superiority of the proposed approach compared with single classifiers for the problem of intrusion detection.

Keywords: Intrusion Detection, Combined Classifiers, PCA, Misuse Detection, Anomaly Detection.

1 Introduction

With the rapid development in the technology based on Internet, new application domains in computer network have emerged. As networks grow in both importance and size, there is an increasing need for effective security monitors such as network intrusion detection systems to prevent illicit accesses. Intrusion detection systems provide a layer of defense which oversees network traffic to identify suspicious activity or patterns that may suggest potentially hostile traffics.

One promise for network intrusion detection is the abnormal access pattern that is generated by scans. Sources that attempt to access an unusual number of uncommon or non-existent destinations, or propagate an irregular number of failed connections are often deemed suspicious [1].

An intrusion detection system (IDS) attempts to detect attacks by monitoring and controlling the network behavior. While many existing IDSs require manual definitions of normal and abnormal behavior (intrusion signatures), recent work has shown that it is possible to identify abnormalities automatically using machine learning or data mining techniques. These works analyze network or system activity logs to generate models or rules, which the IDS can use to detect intrusions that can potentially compromise the system reliability.

Numerous approaches based on soft computing techniques such as artificial neural networks and fuzzy inference systems are proposed in the literature for the purpose of

I. Cervesato (Ed.): ASIAN 2007, LNCS 4846, pp. 254 – 260, 2007.

intrusion detection. In [2] two hierarchical neural network frameworks, serial hierarchical IDS (SHIDS) and parallel hierarchical IDS (PHIDS), are proposed. BPL and RBF are two important learning algorithms used in these neural networks. Authors have shown that BPL has a slightly better performance than RBF in the case of misuse detection, while the RBF takes less training time. On the other hand RBF shows a better performance in the case of anomaly detection. In [3], authors proposed ANNs and support vector machine (SVM) algorithms for ID with frequency-based encoding method. In the chosen DARPA data set, they used 250 attacks and 41,426 normal sessions. The percentage of detection rate (%DR) they archived were between 43.6% and 100% while percentage of false positive rate (%FPR) varied from 0.27% to 8.53% using different thresholds. An in depth review of several anomaly detection techniques for identification of different network intrusions are brought in [4].

In [5], authors have proposed an experimental framework for comparative analysis of both supervised and unsupervised learning techniques including C.45, multi-layer perceptron (MLP), K-nearest neighbor (KNN), etc. The best result they attained was 95% DR and 1% FPR using C.45 algorithm.

In [6] a set of fuzzy rules are generated that can distinguish anomalous connections using only normal samples. Their approach uses genetic algorithms to evolve a set of rules. In [7], SVM was used as an analysis engine which does some preprocessing on the input data. Fuzzy logic is then used as a decision making engine.

It is well known that principal component analysis (PCA) is the most popular feature reduction and data compression method. It has also been applied to the domain of ID [8]. In [9], neural network principal component analysis (NNPCA) and nonlinear component analysis (NLCA) are proposed to reduce the dimension of network traffic patterns. Their approach is based on comparing information of the compressed data with that of the original data. In [10], PCA was used to detect selected denial-of-service and network probe attacks. The authors analyzed the loading values of the various feature vector components with respect to the principal components.

In [11], an ensemble method for intrusion detection is used. They have considered two types of classifiers; ANN and SVM. Another ensemble method is proposed in [12]. In their method, each member of the ensemble is trained on a distinct feature representation of patterns and then the results of the ensemble members are combined. In this paper we propose a new combining classifier approach to intrusion detection by considering a set of heterogeneous classifiers. Four different base classifiers perform classification over an input pattern. Results are then combined using three combing methodologies.

The reminder of this paper is organized as follows. Problem of intrusion detection is defined in more detail in section 2. Section 3 explains the datasets and brings the results of single classifiers. Our proposed method for classifier combination and its results are shown in section four. Finally section five, draws conclusions and summarizes the paper.

2 Intrusion Detection

Intrusion detection process is a software or hardware product that detects illicit activities, which are defined as attempts to compromise the confidentiality,

integrity, availability, or to bypass the security mechanisms of a host or network. There exist mainly two categories of intrusion detection techniques: anomaly detection and signature recognition (misuse detection). Signature recognition techniques store patterns of intrusion signatures and compare those signatures with the observed activities for a match to detect an intrusion. The misuse detection, first attempts to model specific patterns of intrusions to a system, then systematically scans the system for their occurrences. Since the knowledge of the intrusions has to be known before the modeling, this method is mostly used to detect well-known intrusions. Although many existing intrusion detection systems are based on signature recognition techniques, anomaly detection techniques are better to detect novel intrusions or new variants of known intrusions. Anomaly detection creates a profile of typical normal traffic activities or user behaviors, then it compares the deviation between the profile and the input activity with a preset threshold to decide whether the input instance is normal or not. The preset threshold can be adjusted to meet desired performance. Signature recognition techniques may be more accurate in detecting known intrusions. Also many known attacks can be easily modified to present many different signatures. Hence, signature recognition techniques and anomaly detection techniques can be used together to complement each other by monitoring the same activities and generating their own results regarding the intrusiveness of the activities. Anomaly detection addresses the problem of detecting novel intrusions. Usually, it cannot provide detailed information about the attacks. A well designed intrusion detection system should have the ability to detect both misuse and anomaly attacks.

It is important to establish the key differences between anomaly detection and misuse detection approaches. The most significant advantage of misuse detection approaches is that known attacks can be detected fairly reliably and with a low false positive rate. Since specific attack sequences are encoded into misuse detection systems, it is very easy to determine exactly which attacks, or possible attacks, the system is currently experiencing. If the log data does not contain the attack signature, no alarm is raised. As a result, the false positive rate can be reduced very close to zero.

However, the key drawback of misuse detection approaches is that they cannot detect novel attacks against systems that leave different signatures behind. Anomaly detection techniques, on the other hand, directly address the problem of detecting novel attacks against systems. This is possible because anomaly detection techniques do not scan for specific patterns, but instead compare current activities against statistical models of past behavior. Any activity sufficiently deviant from the model will be flagged as anomalous, and hence considered as a possible attack. Furthermore, anomaly detection schemes are based on actual user histories and system data to create its internal models rather than predefined patterns. Though anomaly detection approaches are powerful in that they can detect novel attacks, they have their drawbacks as well. For instance, one clear drawback of anomaly detection is its inability to identify the specific type of attack that is occurring. However, probably the most significant disadvantage of anomaly detection approaches is the high rates of false alarm.

3 Intrusion Detection Using Single Classifiers

3.1 Dataset

In the 1998 DARPA intrusion detection evaluation program, an environment was set up to acquire raw TCP/IP dump data for a network by simulating a typical U.S. Air Force LAN. The LAN was operated like a real environment, but being blasted with multiple attacks. For each TCP/IP connection, 41 various quantitative and qualitative features were extracted. Of this database a subset of 494021 data were used, of which 20% represent normal patterns. The four different categories of attack patterns are:

a. **Denial of Service (DOS) Attacks:** A denial of service attack is a class of attacks in which an attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine. Examples are Apache2, Back, Land, Mail bomb, SYN Flood, Ping of death, Process table, Smurf, Syslogd, Teardrop, Udpstorm.

b. **User to Superuser or Root Attacks (U2Su):** User to root exploits are a class of attacks in which 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. Examples are Eject, Ffbconfig, Fdformat, Loadmodule, Perl, Ps, Xterm.

c. **Remote to User Attacks (R2L):** A remote to user attack is a class of attacks in which an attacker sends packets to a machine over a network–but who does not have an account on that machine; exploits some vulnerability to gain local access as a user of that machine. Examples are Dictionary, Ftp_write, Guest, Imap, Named, Phf, Sendmail, Xlock, Xsnoop.

d. **Probing (Probe):** Probing is a class of attacks in which an attacker scans a network of computers to gather information or find known vulnerabilities. An attacker with a map of machines and services that are available on a network can use this information to look for exploits. Examples are Ipsweep, Mscan, Nmap, Saint, Satan.

3.2 Single Classifier Recognition

In our experiments, we performed 5-class classification. The (training and testing) data set contains 11982 randomly generated points from the five classes, with the number of data from each class proportional to its size, except that the smallest class is completely included. The normal data belongs to class 1, probe belongs to class 2, denial of service belongs to class 3, user to super user belongs to class 4, remote to local belongs to class 5. A number of 6890 points of the total data set (11982) was randomly selected for testing and the rest for the train. Smaller number of training patterns than test patterns is because the intrusion detection method must learn from few learning samples the characteristics of the intrusions.

3.2.1 ANN

The set of 5092 training data is divided into five classes: normal, probe, denial of service attacks, user to super user and remote to local attacks, where the attack is a

collection of 22 different types of instances that belong to the four classes described in section 2, and the other is the normal data. In our study we used two hidden layers with 20 and 30 neurons respectively and the networks were trained using standard back propagation algorithm.

3.2.2 SVM

The same training test (5092) used for training the neural networks and the same testing test (6890) used for testing the neural networks were used to validate the performance of SVM. Because SVMs are only capable of binary classifications, we will need to employ five SVMs, for the 5-clas classification problem in intrusion detection, respectively. We partition the data into the two classes of "Normal" and "Rest" (Probe, DoS, U2Su, R2L) patterns, where the Rest is the collection of four classes of attack instances in the data set. The objective is to separate normal and attack patterns. We repeat this process for all classes. Training is done using the RBF (radial bias function) kernel option.

3.2.3 Decision Trees

The decision tree is constructed during the learning phase, it is then used to predict the classes of new instances. Most of the decision trees algorithms use a top down strategy, i.e from the root to the leaves. Two main processes are necessary to use the decision tree: the building process and the classification process. The same dataset as ANN and SVM were used for building and verifying decision trees. C4.5 algorithm with normalized information gain was used in tree building.

3.2.4 *k*NN

In *k*NN, an input pattern is classified by a majority vote of its neighbors, with the pattern being assigned the class most common amongst its k nearest neighbors. Training patterns are saved in memory. Then in classification a majority vote determines the class label of a test pattern. In our experiments we used Euclidean distance to find the nearest neighbors. Using a cross-validation experiment we found k=3 the most sailable value for k. Results showing the performances of four single classifiers discussed above is summarized in table one.

Table 1. Intrusion detction performacne using four heterogenius classifiers

Classification / *Method* / *Performance*	ANN	SVM	Decision Tree	kNN (k=3)
Detection Rate (DR)	98.45%	99.5%	95.5%	88.9%
False Positive Rate (FPR)	3.57%	2.9%	1.2%	4.1%

4 Combing Classifiers for Intrusion Detection

The ensemble method proposed for solving the Intrusion Detection problem can be illustrated as follows. First each trained classifiers over the same training set is used independently to perform attack detection. Then the evidences are combined in order

to produce the final decision. The approach based on classifier combination may also attain effective attack detection as the combination of multiple evidences usually exhibits higher accuracies, i.e. lower false positives, than individual decisions. In addition, the generalization capabilities of pattern recognition algorithms allow for the detection of novel attacks that is not provided by rule-based signatures.

In order to illustrate combination approach, we used three simple fusion techniques: the majority voting rule, the average rule and the "belief" function. These fusion techniques compute the final decision from the set of decisions of an ensemble made up of K classifiers. The "majority voting rule "assigns a given input pattern to the majority class among the K outputs of the classifiers combined. The "average rule" assigns a given input pattern to the class with the maximum average posterior probability, the average being computed among the K classifiers (this rule can be applied if classifiers provide estimates of posterior probabilities, like multi-layer perceptron neural networks). The third fusion rule is based on the computation of a "belief" value for each data class given the set of outputs of the K classifiers. Belief values are based on estimates of the probabilities that a pattern assigned to a given data class actually belongs to that class or to other classes. These probabilities can be easily computed from the confusion matrix on the training set. The classification is then performed by assigning the input pattern to the data class with the maximum "belief" value. For more details about the above combination methods the reader is referred to [13].

Results of combining classifiers to recognize intrusions using three combination approaches are shown in table 2.

Table 2. Intrusion detection performance using combination of four disticnt classifiers

Combination Method / *Performance*	Majority	Bayesian Average	Belief
Detection Rate (DR)	99.18%	99.33%	99.68%
False Positive Rate (FPR)	1.20%	1.03%	0.87%

5 Conclusions

Our results show the effectiveness of classifier combination in providing more reliable results, as the final decision depends on the agreement among distinct classifiers. In particular better results have been obtained by the fusion rule based on the "belief" function paradigm because it takes into account the different discriminative power provided by the considered feature sets. Other combination schemes should be devised to further improve the presented figures. In addition, more extensive testing is required to compare IDSs based on pattern recognition tools with traditional IDSs. With respect to the capability of ensemble learning approaches of providing a better trade-off between generalization capabilities and false alarm rate, it can be concluded that combination reduces the overall error rate, but may also reduce the generalization capabilities. This aspect should be further investigated in order to deploy effective IDSs based on pattern recognition.

References

1. Roesch, M.: Snort: Lightweight intrusion detection for networks. In: Proceedings of the 13th Conference on Systems Administration (LISA 1999), pp. 229–238 (1999)
2. Zhang, C., Jiang, J., Kamel, M.: Intrusion detection using hierarchical neural networks. Pattern Analysis and Machine Intelligence Research Group, Department of Electrical and Computer Engineering, University of Waterloo, Canada (2004)
3. Wun-Hua, C., Sheng-Hsun, H., Hwang-Pin, S.: Application of SVM and ANN for intrusion detection. Comput. Oper. Res. 32(10), 2617–2634 (2005)
4. Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., Srivastava, J.: A comparative study of anomaly detection schemes in network intrusion detection. In: Proceedings of the Third SIAM Conference on Data Mining (2003)
5. Pavel, L., Patrick, D., Christin, S., Rieck, K.: Learning Intrusion Detection: Supervised or Unsupervised. In: Roli, F., Vitulano, S. (eds.) ICIAP 2005. LNCS, vol. 3617, pp. 50–57. Springer, Heidelberg (2005)
6. Gómez, J., González, F., Dasgupta, D.: An immuno-fuzzy approach to anomaly detection. Fuzzy Systems. In: FUZZ 2003. 12th IEEE International Conference on Fuzzy Systems, vol. 2, pp. 1219–1224 (2003)
7. Yao, J., Zhao, S., Saxton, L.: A study on fuzzy intrusion detection. Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2005. In: Dasarathy, B.V.(ed.) Proceedings of the SPIE, vol. 5812, pp.23–30 (2005)
8. Oja.: Principal components, minor components, and linear neural networks. Neural Networks 5(6), 927–935 (1972)
9. Kuchimanchi, G.K., Phoha, V.V., Balagami, K.S., Gaddam, S.R.: Dimension reduction using feature extraction methods for Real-time misuse detection systems. In: Proceedings of the 2004 IEEE Workshop on Information Assurance and Security, West Point, NY, pp. 195–202 (2004)
10. Labib, K., Vemuri, V.R.: Detecting and visualizing denial-of-service and network probe attacks using principal component analysis. In: Third Conference on Security and Network Architectures, La Londe, France (2004)
11. Mukkamala, S., Sung, A.H., Abraham, A.: Intrusion Detection Using Ensemble of Soft Computing Paradigms. Journal of Network and Computer Applications 28, 167–182 (2005)
12. Didaci, L., Giacinto, G., Roli, F.: Ensemble Learning for Intrusion Detection in Computer Networks. In: Workshop su apprendimento automatico: metodi ed applicazioni (2006)
13. Xu, L., Krzyzak, A., Suen, C.Y.: Methods for combining multiple classifiers and their applications to handwriting recognition. IEEE Trans. Systems, Man and Cybernetics 22, 418–435 (1992)