An Experiment on Naïve Bayes Classifier Detecting Network Traffic Anomalies

K. Limthong[1] **and T. Tawsook**[2,C]
[1] *Department of Informatics, School of Multidisciplinary Sciences, Graduate University for Advanced Studies (Sokendai), 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan*
[2] *Department of Computer Engineering, School of Engineering, Bangkok University, 9/1 Phahonyothin Rd., Klong Luang, Pathumthani, 12120, Thailand*
[C] **E-mail:** thidarat.t@bu.ac.th; **Tel:** (+66) 0-2902-0299 ext. 2620; **Fax:** (+66) 0-2516-8554

ABSTRACT

One of the crucial responsibilities for network administrators is detecting anomalous network traffic, which might produces some serious problems such as network traffic congestion or security issues. If network administrators had an effective tool for discovering many types of anomalies, they would immediately prevent and solve the network problems caused by such anomalies. In this paper, we intensively performed experiments on nine features used by naïve Bayes classifier, which takes a short computation time to detect volume-based anomalies and does not need a lot of storage to keep training parameters, in order to detect various types of anomalies in network traffic at the edge of internetwork. The results of our experiment would help network administrators to select the feasible and practical features for different kinds of anomalies, namely the denial of service attack, IP scanning, port scanning and amplified attack.

Keywords: naïve Bayes classifier, anomaly detection, network traffic analysis, machine learning

1. INTRODUCTION

Detecting anomalous traffic which might cause network problems is a vital responsibility of network administrators. The volume-based anomalies, such as viruses, worms, denial of service (DoS) attacks, scanning, and spamming, are unusual incidents that may lead network into serious traffic congestion or security issues. If network administrators had an effective tool to discover anomalous traffic, they would immediately prevent and solve the network problems caused by such anomalies. The serveral techniques detecting traffic anomalies can be classified into two major approaches: the signature based and anomaly based detection [1].

The signature based system monitors packets and compares packets with predetermined attack patterns known as signatures. This technique is simple and efficient processing of the audit data. The false positive rate of this system can also be low. However, comparing packets with the signatures is a time consuming task and has limited predictive abilities. The signature based detection cannot detect novel anomalies which not be defined in signatures, so administrators have to update the system signatures regularly. The opposite of signature based techniques is the anomaly based detection. This techniques automatically learns the behaviour of network traffic and has the possibility of detecting novel anomalies. There are many research studies related to anomaly based detection [2].

Researchers have been applying many anomaly based techniques to task of anomaly detection. For instance, the studies of [3] combined signal processing, the wavelet transform, with statistical techniques to detect network traffic anomalies at the edge of internetwork. The other technique have been focused recently are the machine learning methods. Several of machine learning

algorithms have been employed to detect anomalies in network traffic, such as the naïve Bayes, k-nearest neighbor, and support vector machine.

In our experiment, we attempted to explore the impact of feature and interval selection for the naïve Bayes classifier in task of anomaly detection. Morover, we drew a comparison of among nine favorite features used in numerous studies. The results of this experiment would help network administrators to choose the feasible and practical features for the different kinds of anomalies, namely the denial of service attack, IP scanning, port scanning and amplified attack.

2. MATERIALS AND METHODS

We acquired three-month of anomaly-free network data traces from an edge router in a campus network at the Kasetsart University, Thailand. We chose 39 days of data traces to train the naïve Bayes classifier in the training phase and 16 days for combining them with several types of anomalies. The selected anomalies are from the Lincoln Laboratory at the Massachusetts Institute of Technology [4]. These anomalies were provided for researchers who would like to compare and evaluate the efficiency of their own anomaly detection method.

In the training phase, we trained the classifier using only the normal category (called one class training) to compute the probability density function (pdf) for each interval. In addition, we calculated the discriminant function of every interval based on five values: 2σ, 2.5σ, 3σ, 3.5σ, and 4σ. In this case, σ is the standard deviation of pdf for the normal category of each interval. For example, we assume that the discriminant function equals $P(2\sigma)$. In the testing phase, if the probability of feature value x, P(x), is equal to or greater than $P(2\sigma)$, we classify that interval as the normal category. On the other hand, if the P(x) is lower than $P(2\sigma)$, we classify that interval as the anomaly category.

Table 1. Characteristics of selected anomalies

Source	#SrcAddr	#DstAddr	#SrcPort	#DstPort	#Packet	Packet Size Min:Avg:Max (Byte)	Duration (Second)	#AvgPacket Per Second	%Anomaly
back									
Week2 Fri	1	1	1,013	1	43,724	60:1,292.31:1,514	651	67.16	0.75
Week3 Wed	1	1	999	1	43,535	60:1,297.29:1,514	1,064	40.92	1.23
ipsweep									
Week3 Wed	1	2,816	1	104	5,657	60:60.26:118	132	42.86	0.15
Week6 Thurs	5	1,779	2	105	5,279	60:67.75:118	4,575	1.15	5.30
neptune									
Week5 Thurs	2	1	26,547	1,024	205,457	60:60:60	3,143	65.37	3.64
Week6 Thurs	2	1	48,932	1,024	460,780	60:60:118	6,376	72.27	7.38
Week7 Fri	2	1	25,749	1,024	205,600	60:60:60	3,126	65.77	3.62
portsweep									
Week5 Tues	1	1	1	1,024	1,040	60:60:60	1,024	1.02	1.19
Week5 Thurs	1	1	1	1,015	1,031	60:60:60	1,015	1.02	1.17
Week6 Thurs	2	2	2	1,024	1,608	60:60:60	1,029	1.56	1.19
smurf									
Week5 Mon	7,428	1	1	1	1,931,272	14:1,066:1,066	1,868	1,033.87	2.16
Week5 Thurs	7,428	1	1	1	1,932,325	14:1,066:1,066	1,916	1,008.52	2.22
Week6 Thurs	7,428	1	1	1	1,498,073	1,066:1,066:1,066	1,747	857.51	2.02

In the testing phase, we combined five different types of anomalies listed in Table 1 with anomaly-free network traffic in order to evaluate the performance. The *back* attack is a denial of service attack against the Apache web server through port 80, where a client requests a URL containing many backslashes. The *ipsweep* attack is a surveillance sweep performing either a port sweep or ping on multiple IP addresses. The *neptune* attack is a SYN flood denial of service attack on one or more destination ports. The *portsweep* attack is a surveillance sweep through many ports to determine which services are supported on a single host. The *smurf* attack is an amplified attack using ICMP echo reply flood.

We performed the task of naïve Bayes classifier based on nine features listed in Table 2. The number of packets, the sum of packet size, and the number of flows, feature number 1-3, are common features which have been applied to several techniques in this fields. The feature number 4 and 5 are the number of source and destination addresses occurred during time interval. The feature number 6 and 7 are the number of source and destination ports appeared as time interval. The dtAddr is the difference in the number of source and destination addresses. The last feature, dtPort, also means the difference in the number between source and destination ports.

Table 2. Feature selection

Feature	Description
1.Packet	The number of packets per time interval
2.Byte	The sum of packet size per time interval
3.Flow	The number of flows per time interval
4.SrcAddr	The number of source addresses per time interval
5.DstAddr	The number of destination addresses per time interval
6.SrcPort	The number of source ports per time interval
7.DstPort	The number of destination ports per time interval
8.dtAddr	|SrcAddr – DstAddr| per time interval
9.dtPort	|SrcPort – DstPort| per time interval

To evaluate the performance of classifier, we use the precision, recall [5], and F-measure [6] on a per-interval basis. All the measures can be calculated based on the following four values: the true positive (TP; the number of anomalous intervals correctly detected), the false positive (FP; the number of normal intervals wrongly detected as the anomalous intervals), the false negative (FN; the number of anomalous intervals not detected), and the true negative (TN; the number of normal intervals correctly detected). All of the parameters are defined in Table 3.

Table 3. Sampling interval-based evaluation

Test Result	Actual Status	
	Anomaly	Normal
Anomaly	True Positive (TP)	False Positive (FP)
Normal	False Negative (FN)	True Negative (TN)

From these parameters, the precision, recall, and F-measure are calculated by using Eqs. (1)-(3), respectively:

$$precision = \frac{TP}{TP+FP}, \tag{1}$$

$$recall = \frac{TP}{TP+FN}, \tag{2}$$

$$F-measure = 2 \times \frac{precision \times recall}{precision+recall}. \tag{3}$$

In Eq. (1), the precision or positive predictive value is the percentage of detected intervals which are actually anomalies. In Eq. (2), the recall or sensitivity is the percentage of the actual anomalous intervals which are detected. Eq. (3) shows the F-measure which is the harmonic mean of the precision and recall. We used the F-measure as a single measure of the classifier performance.

3. RESULTS

We performed and extracted the features as described in section 2 for all 39 training and 16 testing data traces using the GNU Compiler Collection (GCC) and the Libpcap library. In this experiment, the classification is interval-based, so each training or testing sample represents one interval from the training or testing data traces.

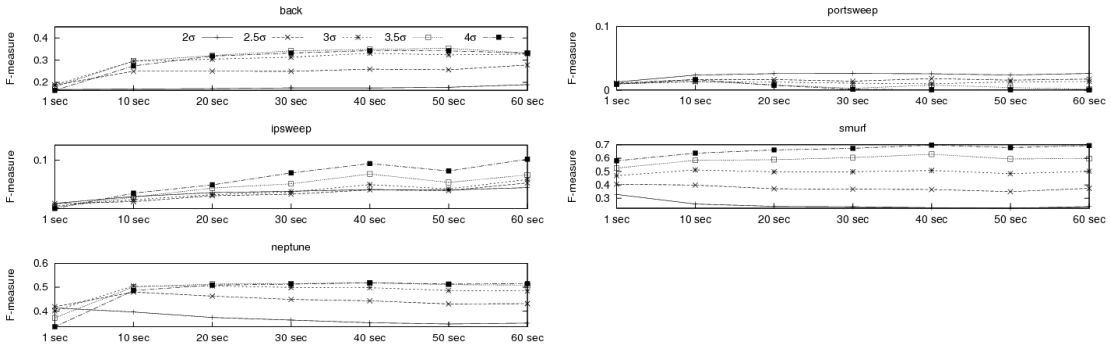

Figure 1. F-measure with number of packets as feature

The figure 1 shows an example of F-measure with the number of packet feature (f1) for each type of anomaly. The x-axis indicates the interval values or bin sizes used in our experiment. We varied the interval values with 1, 10, 20, 30, 40, 50, and 60 seconds. The different lines in the same type of anomaly mean distinctive values of discriminant function as we set, we chose the discriminant function values with 2σ, 2.5σ, 3σ, 3.5σ, and 4σ as descripted in section 2. We performed like this feature on all of the nine features to discover the best average values of F-mesurement for each type of anomaly.

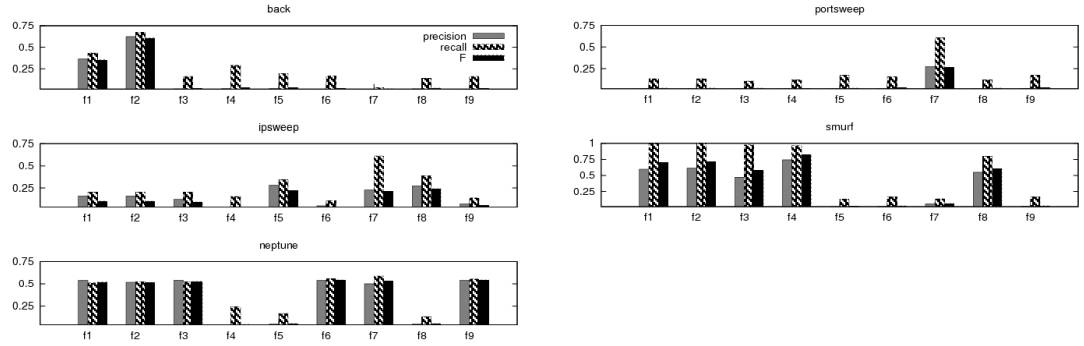

Figure 2. Precision, Recall, and F-measurement for each feature

The best average values of precision, recall, and F-measurement for each feature are depicted in figure 2. The horizontal axis shows in different feature from the number of packets feature (f1) to the delta port feature (f9) as listed in Table 2. The highest F-measure of the back attack belongs to the sum of packet size feature (f2) with 0.6071 value. For the ipsweep attack, the feature number 8, dtAddr, contain the top F-measure with 0.2414 value. The second and third are 0.2197 and 0.2113 value for feature number 5 and 7 respectively. Most of features on neptune attack produced F-measure values similarly except feature number 4, 5 and 9. However, the best value is 0.5443 by the number of source ports feature (f6). For the portsweep attack, the number of destination port (f7) is outstanding compared with other features. The maximum F-measure value of portsweep attack is 0.2653. The last one, the feature number 4 gains the highest F-measure value of smurf attack 0.8225 and following by feature number 2, 1, 8, and 3 respectively.

4. DISCUSSION AND CONCLUSION

We intensively studied of a machine learning method on network traffic anomaly detection using the naïve Bayes classifier. In our experiment, we acquired anomaly-free traffic traces from the edge router of a campus network. The data traces were separated into a training phase to train classifier, and a testing phase to combine them with various types of anomalies. The five distinct types of anomalies were chosen from the common test bed for anomaly detection system. In the testing phase, we varied the time intervals and discriminant function parameters of classifier. We evaluated the efficiency of classifier by F-measure value for each anomaly type. We also looked at nine different features to compare the efficiency of classifier.

The experimental results show that the performance of the naïve Bayes classifier increases for some cases when we use longer interval values. For example, in case of the neptune attack, the F-measure of feature number 1 at 10 second interval value is more preferable than that at 1 second as shown in Figure 1. However, the F-measure values around 20-60 seconds do not seem different from those at 10 second interval values. Moreover, we found that the parameters of discriminant function also have an effect on the F-measure when we varied the interval values. The other thing that affects the performance of classifier is the feature selection as the results shown in Figure 2.

In summary, the network administrator who would like to apply the naïve Bayes classifier on their own system should take the combination of interval values, discriminant fuction, and feature selection into consideration. These factors can engender the performance effects in the anomaly detection system. In future work, we plan to use other classifiers, such as the k-nearest neighbor or the support vector machine, using the same data set in order to evaluate their performance.

ACKNOWLEDGMENTS

We gratefully acknowledge the funding from the Faculty Members Development Scholarship Program of Bangkok University, Thailand. The authors would like to thank all of the anonymous reviewers for their excellent suggestions that have greatly improved the quality of this paper.

REFERENCES

1. Varun Chandola, Arindam Banerjee, and Vipin Kumar, “Anomaly detection: A survey”, *ACM Comput. Surv*, 2009, **41**(3), 1-58.
2. Animesh Patcha and Jung-Min Park, “An overview of anomaly detection techniques: Existing solutions and latest technological trends”, *Computer Networks*, 2007, **51**(12), 3448-3470.
3. Limthong, K., Watanapongse, P., and Kensuke, F., “A wavelet-based anomaly detection for outbound network traffic”, *Information and Telecommunication Technologies (APSITT), 2010 8th Asia-Pacific Symposium on*, 2010, 1-6.
4. Lippmann, R.P., Fried, D.J., Graf, I., Haines, J.W., Kendall, K.R., McClung, D., Weber, D., Webster, S.E., Wyschogrod, D., Cunningham, R.K., and Zissman, M.A., “Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation”, *DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings*, 2000, **2**, 12-26.
5. Jesse Davis and Mark Goadrich, “The relationship between Precision-Recall and ROC curves”, *Proceedings of the 23rd international conference on Machine learning (ICML '06)*, ACM, New York, USA, 2006, 233-240.
6. C. J. V. Rijsbergen, *Information Retrieval*, 2nd ed, Butterworth-Heinemall, Newton, MA, USA, 1979.