Impact of Time Interval on Naïve Bayes Classifier for Detecting Network Traffic Anomalies

Kriangkrai Limthong[*], Kensuke Fukuda[†], Yusheng Ji[†], and Shigeki Yamada[†]
[*] Graduate University for Advanced Studies (Sokendai), Tokyo 101-8430 Japan.
[†] National Institute of Informatics, Tokyo 101-8430 Japan.
{krngkr,kensuke,kei,shigeki}@nii.ac.jp

Abstract

Choosing a proper time interval value for detecting network traffic anomalies is a big problem for many network administrators because the different interval values produce the dissimilar network statistics, such as the mean, variance, and distribution. If network administrators select the right time interval value for their own anomaly detection system, it dramatically and significantly increases the accuracy of the system. We intensely studied on the efficiency and impact of time interval values for network traffic anomaly detection by using the 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. The results of this study would help network administrators to select the most practical time interval value for several types of anomalies, e.g. the denial of service attack, IP scanning, port scanning, amplified attack.

Key Words: time interval, naïve Bayes, anomaly detection, network traffic analysis

1. Introduction

Detecting anomalous traffic which might cause network problems is a vital responsibility of network administrators. 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 problems. If network administrators had an effective tool to perceive anomalous traffic, they would immediately prevent and solve the network problems caused by such anomalies. Most anomaly detection systems use packet aggregation techniques to minimize resources of network equipment. However, selecting a proper time interval value is one of the serious problems for anomaly detection systems because it produces an effect on efficiency and accuracy of the systems.

The impact of time interval values has been extensively studied in terms of network statistics, such as the mean, variance, and distribution of variables. Many anomaly detection methods often depend on such statistics, especially in supervised learning methods such as the naïve Bayes algorithm. Although this algorithm requires a small amount of training data to estimate parameters (the mean and variance), varying the time interval value also alters the value of these parameters. Therefore, many network administrators still have a question, how the time interval value affects the efficiency of anomaly detection?

In this paper, we concentrated on the effects of time interval value on the naïve Bayes classifier used for network traffic anomaly detection. We performed experiments on real network traffic acquired from a campus network. In addition, we selected five different types of test bed anomalies from DARPA in order to evaluate the efficiency and accuracy of naïve Bayes classifier by using seven time interval values. The results of our study illustrated three separate features; the number of packets, the sum of packet size, and the number of flows, used in the naïve Bayes classification method.

We introduce the related work focused on the effects of time interval value on network traffic statistics in the next section. Section 3 explains the materials and methods used in our experiments on campus network traffic. The experimental results from our intensive study are exhibited in Section 4. In the last section, we discuss the experimental results, draw our conclusion of the experiments, and introduce future work.

2. Related Work

Packet aggregation techniques have been widely applied to the Internet traffic measurements, for instance, the report of [1] showed statistical analysis of data traffic measurements from a campus networks

has non-stationary features in the aggregate traffic stream. The study of [2] proposed a distributed packet aggregation algorithm to improve the VoIP quality over multi-hop mesh network. The packet aggregation techniques do not only apply to network traffic analysis and management purposes but also to the task of anomaly detection and classification [3].

From previous studies, the impact of time interval values on packet aggregation in task of anomaly detection have been rarely investigated. While packet sampling scheme are considered that the interval value produces an effect on network statistics and accuracy of anomaly detection [4]-[6].

The machine learning technique is one of the popular methods tasked with anomaly detection. The authors from [7] proposed genetic algorithms and decision trees for automatically classifying anomalies. The study presented in [8] suggested using the naïve Bayes algorithm is better than the neuron network algorithm. However, this paper tried to find out how the time interval value affects the efficiency of naïve Bayes classifier.

The significant contributions of this work are summarized as follows: (i) one of the initial attempts at exploring the impact of time interval values on the naïve Bayes classifier in task of anomaly detection, (ii) comparison of seven different time interval values on five dissimilar types of anomalies, (iii) comparison of three common features used by the naïve Bayes classifier to detect anomalies in real network traffic.

3. Materials and Methods

In the following subsections, we are going to describe the data sets, procedure, classifier, and evaluation metrics in our experiment.

3.1 Data Sets

We acquired three-month data traces of anomaly-free network from an edge router in a campus network at the Kasetsart University, Thailand. This center is for college students, educators, and researchers so that they can ascertain advantageous information for their studies from the Internet. There are around 1,300 users per day, and the service time is between 8:00 and 24:00 on workdays. Users cannot change or install any software in the computer client, and administrators provided appropriate software for all ordinary users. Moreover, administrators regularly update virus signatures of anti-virus software which are installed on all clients. At the end of every day, all clients reverse all software into the initial state so we can guarantee that all clients are clean.

We chose 39 days of clean 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 [9], [10]. These anomalies were provided for researchers who would like to compare and evaluate the efficiency of their own anomaly detection method.

We selected five different types of anomalies with the characteristics listed in Table 1. 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.

There are two reasons why we selected the network data traces from different sources. First, although the data trace from MIT consists of both normal and anomaly traffic, we do need the real and clean network traffic to train classifier because the accurate decision of naïve Bayes classifier depends on training data. The second reason is the selected anomalies are test bed data that anyone can use for evaluating their own anomaly detection methods.

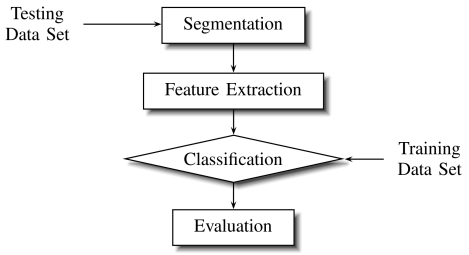

Fig. 1. Experiment flowchart

3.2 Procedure

We divided the procedure into two phases: the training phase and the testing phase. In the training phase, we fed 39 days of anomaly-free data traces into the naïve Bayes classifier. Afterward, we combined the other 16 days of anomaly-free data traces with selected anomalies and used them to evaluate the performance of classifier in the testing phase.

The flowchart of our experiment is shown in Fig. 1. The role of the segmentation step is to group

Table 1. Characteristics of Selected Anomalies

Source	#SrcAddr	#DstAddr	#SrcPort	#DstPort	#Packet	Packet Size Min:Avg:Max (Byte)	Occurrence (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

packets from network traffic into different interval values. In this study, we varied in the time interval by using seven values: 1, 10, 20, 30, 40, 50, and 60 seconds. The feature extraction has the main function of extracting the key network features for each interval from the network traffic. We conducted the experiments on three network features, namely the number of packets, the sum of packet size, and the number of flows. In the classification step, we applied the naïve Bayes classifier to distinguish between normal and anomalous traffic, we will explain this in more detail in the next subsection. The last step is an evaluation of anomaly detection, and we also described the method and metrics for evaluating the performance of classifier in the evaluation subsection.

3.3 Naïve Bayes Classifier

The naïve Bayes classifier technique is based on Bayes' theorem, which can be simplified in the Bayes' formula as

$$P(\omega|x) = \frac{p(x|\omega)P(\omega)}{p(x)}, \qquad (1)$$

so we can express Eq. (1) in plain English by saying that

$$posterior = \frac{likelihood \times prior}{evidence}. \qquad (2)$$

$P(\omega|x)$ is the *posterior* probability of category ω given that feature value x has been measured. $p(x|\omega)$ is the *likelihood* of category ω with respect to feature value x. $P(\omega)$ is the *prior* probability of category ω, and $p(x)$ is the *evidence* that can be viewed as a scale factor and guarantees the posterior probabilities sum to one.

In the training phase, we trained the classifier by using only the normal category (called one class training) to compute the probability density function (pdf) for each interval. The main reason that we use one class training is we would like to discriminate anomalies from normal traffic. Thus we decide that the packet or behavior which not conform to normal traffic is an anomaly. In addition, we calculated the discriminant function of every interval for naïve Bayes leaning algorithm 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 2σ. In the testing phase, if the probability of feature value x, $P(x)$, is equal to or greater than 2σ, 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 2. Time Interval-Based Evaluation

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

3.4 Evaluation

To evaluate the performance of classifier, we use the precision, recall [11], and F-measure [12] 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 2. From these parameters, the precision, recall, and F-measure are calculated by using Eqs. (3)-(5), respectively:

$$precision = \frac{TP}{TP + FP}, \quad (3)$$

$$recall = \frac{TP}{TP + FN}, \quad (4)$$

$$F\text{-}measure = 2 \times \frac{precision \times recall}{precision + recall}. \quad (5)$$

In Eq. (3), the precision or positive predictive value is the percentage of detected intervals which are actually anomalies. In Eq. (4), the recall or sensitivity is the percentage of the actual anomalous intervals which are detected. Eq. (5) 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.

4. Results

We performed and extracted the features as described in section 3.2 for all 39 training and 16 testing data traces by using the GNU Compiler Collection 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.

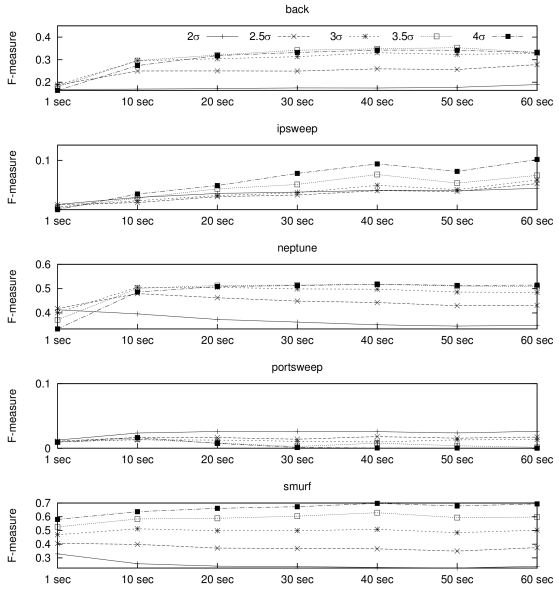

Fig 2. F-measure of naïve Bayes with Number of Packets as Feature

4.1 Feature 1: Number of Packets

Fig. 2 shows the experiment results from the number of packets as a feature of classifier. The *back* attack lines, between 3σ and 4σ lines, were close at a 1-60 sec. Interval. The lowest efficiency was in the 2σ line. The maximum F-measure for this attack was 0.3530, which belonged to the 3.5σ line at 50 sec. and that point increased from 0.1788 at 1 sec.

For the *ipsweep* attack, all the lines continued to increase when the interval values were long. The maximum F-measure value was 0.1016 at 60 sec., which belonged to the 4σ line. At 1 sec., the F-measure of the 4σ line was 0.0199 and increased to the maximum value at 60 sec.

In the *neptune* attack, the 4σ line at 40 sec., obtained the maximum F-measure value, 0.5183. The F-measure of 4σ line started at 1 sec. with 0.3340.

For the *portsweep* attack, all the lines at other interval values showed little change from 1 sec. The 2σ line at 1 sec. gained 0.0129 of F-measure and slightly increased to 0.0264 at 30 sec., where was the maximum F-measure of *portsweep* attack.

In the *smurf* attack, the 4σ line was greater than other lines of all interval values. The 2σ and 2.5σ lines declined when the interval values were long. At 1 sec., the 4σ line F-measure started from 0.5801 and increased to the maximum value, 0.6975, at 40 sec.

4.2 Feature 2: Sum of Packet Size

Fig. 3 depicts the results from the sum of packet size as a feature of classifier. For the *back* attack, the 4σ line had greater efficiency than other lines between 1 and 60 sec. The 4σ line started from 0.5250 at 1 sec. and then reached to the maximum F-measure at 60 sec. with the value 0.6071.

For the *ipsweep* attack, all of the lines in this graph seem similar to the results of number of packet feature. The maximum value of F-measure was the value 0.1006 at 60 sec., which belonged to the 4σ line. At 1 sec., the F-measure of 4σ line was 0.0198 and increased to that at 60 sec.

For the *neptune* attack, although all of the lines look close to the results of previous feature, the maximum F-measure occurred on a different line. The 3.5σ line at 40 sec. had the maximum F-measure value 0.5130. The F-measure of 3.5σ line at 1 sec. was the value 0.3581.

Even if the *portsweep* attack results showed that all lines matched the first feature figure, the values of F-measure were dissimilar. The 2σ line at 1 sec. had the F-measure value 0.0130 and it slightly increased to 0.0263 at 60 sec., which was the maximum value of this line.

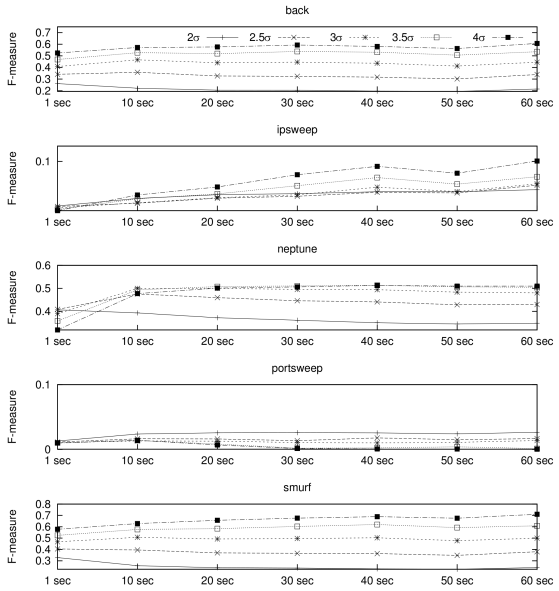

Fig 3. F-measure of naïve Bayes with Sum of Packet Size as Feature

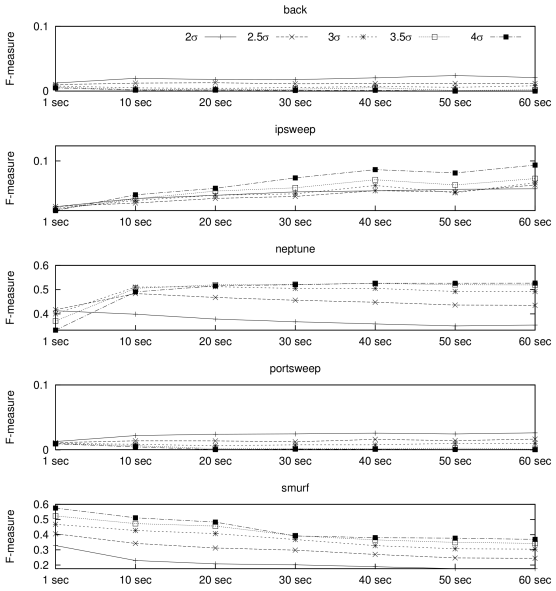

Fig 4. F-measure of naïve Bayes with Number of Flows as Feature

For the *smurf* attack, the F-measure values of 4σ line were higher than F-measure values of other lines for all interval values. The 4σ line F-measure started from 0.5779 at 1 sec. and reached the maximum value 0.7100 at 60 sec.

4.3 Feature 3: Number of Flows

The last figure, Fig. 4, illustrates the results of our experiment with the number of flows as a feature of classifier. For the *back* attack, all lines were lower than the F-measure value 0.03. The 2σ-4σ lines seemed closely for all interval values, there were much more different than both prior features. The maximum F-measure value was 0.0240, which belonged to the 2σ line at 50 sec. The 2σ line got the F-measure value 0.0124 at 1 sec.

For the *ipsweep* attack, when we took the 4σ line into consideration, the maximum value of F-measure was 0.0931 at 60 sec. The F-measure values of 4σ line started with 0.0177 at 1 sec. and then increased to 0.0931 at 60 sec.

For the *neptune* attack, the outcomes of F-measure values for all features look alike. The 4σ line at 60 sec. had the maximum F-measure value 0.5269. The F-measure value of 4σ line at 1 sec. was 0.3330.

All of the outputs from the *portsweep* attack also seem to match the three features. The 2σ line at 1 sec. had the F-measure value 0.0128 and little increased to 0.0262 at 60 sec., where was the maximum value.

For the *smurf* attack, the 4σ line was also greater than other lines at all interval values like other features. However, F-measure trend for all lines was to drop when we used longer interval values, and this was unlike previous features. The 4σ line F-measure started from 0.5756 at 1 sec., and then decreased to 0.3689 at 60 sec.

5. Discussion and Conclusion

We intensively studied into the impact of time interval values on network traffic anomaly detection by using the naïve Bayes classifier. In our experiment, we acquired anomaly-free traffic traces from an edge router of a campus network. The data traces were separated for a training phase to train classifier, and for a testing phase to combine them with selected anomalies. The five distinct types of anomalies were chosen from the common test bed for anomaly detection. In the testing phase, we varied the time interval values between 1 and 60 seconds, and evaluated the efficiency of classifier by looking at F-measure value for each interval value. We also focused on three different features to compare the efficiency of classifier.

The experimental results showed that the performance of the naïve Bayes classifier increased for some of the features when we used longer interval values. For example, in case of the *neptune* attack, F-measure values of all three features at 10 second were more preferable than that at 1 second. However, the F-measure values around 20-60 seconds do not seem

different from those at 10 second interval value. Another example was in the *back* attack, the F-measure of the number of packet feature at 10 second was better than that at 1 second.

We found that the parameters of discriminant function also have an effect on the F-measure when we varied the time interval values. Most of the discriminant functions with the 3.5σ and 4σ value increased when we used interval with longer values. Except for that in the *smurf* attack with the flow feature, no matter what parameters were set, the F-measure decreased when we set the time interval to a longer value.

The last thing that affected the performance of classifier was the feature selection. If we take the *smurf* attack into consideration, the F-measure values of both the number of packets and the sum of packet size features increased between 3σ and 4σ. Whereas, the F-measure values of the number of flow feature fell down for all discriminant function parameters.

In summary, the time interval value had an effect on the naïve Bayes classifier for network traffic anomaly detection. The performance of the naïve Bayes classifier improved when the time interval value was changed, but it also produced some of the worst performance in several cases. Unfortunately, the time interval value was not only a metric that improved or reduced the efficiency of anomaly detection. We should also take the combination of other parameters with the time interval value into consideration, such as the parameters of discriminant function and the feature selection. In some situations, varying the time interval value produced an outstanding performance, but when we switched the features or some of the parameters of discriminant function, it might worsen the performance.

In future work, we plan to apply other classifiers, such as the *k*-nearest neighbor or the support vector machine, on the same data set in order to evaluate their performance.

6. Acknowledgements

We would like to 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.

7. References

[1] C. You and K. Chandra, "Time series models for internet data traffic," in Local Computer Networks, 1999. LCN '99. Conference on, Oct. 1999, pp. 164-171.

[2] K. Kim, S. Ganguly, R. Izmailov, and S. Hong, "On packet aggregation mechanisms for improving voip quality in mesh networks," in Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd, vol. 2, May 2006, pp. 891-895.

[3] G. Shen, D. Chen, and Z. Qin, "Anomaly detection based on aggregated network behavior metrics," in Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on, 2007, pp. 2210-2213.

[4] J. Mai, A. Sridharan, C.-N. Chuah, H. Zang, and T. Ye, "Impact of packet sampling on portscan detection," Selected Areas in Communications, IEEE Journal on, vol. 24, no. 12, pp. 2285-2298, 2006.

[5] G. Androulidakis, V. Chatzigiannakis, and S. Papavassiliou, "Network anomaly detection and classification via opportunistic sampling," Network, IEEE, vol. 23, no. 1, pp. 6-12, 2009.

[6] X. He, W. Yang, and Q. Wang, "An adaptive traffic sampling method for anomaly detection," International Conference on Internet Computing in Science and Engineering, vol. 0, pp. 142-146, 2009.

[7] C. Sinclair, L. Pierce, and S. Matzner, "An application of machine learning to network intrusion detection," in Computer Security Applications Conference, 1999. (ACSAC '99) Proceedings. 15th Annual, 1999, pp. 371-377.

[8] M. P and M. R. Patra, "Network intrusion detection using naive bayes," 2007.

[9] R. Lippmann, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyschogrod, R. Cunningham, and M. Zissman, "Evaluating intrusion detection systems: the 1998 darpa off-line intrusion detection evaluation," vol. 2, 2000, pp. 12-26 vol.2.

[10] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 darpa off-line intrusion detection evaluation," Computer Networks, vol. 34, no. 4, pp. 579-595, 2000, recent Advances in Intrusion Detection Systems.

[11] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in ICML '06: Proceedings of the 23rd international conference on Machine learning. New York, NY, USA: ACM, 2006, pp. 233-240.

[12] C. J. V. Rijsbergen, Information Retrieval. Newton, MA, USA: Butterworth-Heinemann, 1979.