# The Infuence Of Noisy Data On Skype Traffic Classification

Linhua Niu, Xiangzhan Yu and Zhimin Yin
Department of Computer Science and Technology
Harbin Institute of Technology
Harbin, China
xn12280000@126.com, yxz@hit.edu.cn, yzm0621@163.com

***Abstract*—Because of its popularity, encrypted traffic and proprietary design, there has been difficult to detect Skype from other P2P traffics. The research of Skype traffic identification focuses on collecting traffic flow feature and using machine learning method to identification. The key of machine learning method is datasets and flow feature selection. Since there is no publicly available datasets, noisy data can't be avoided. In this paper, I compare two different machine learning classification techniques, C4.5 and Neural Networks. Results show that C4.5 is better than Neural Networks when noisy data percent is low and Neural Networks is steady when noisy data percent is high.**

***Keywords-component: Skype Traffic Classification; Neural Networks; C4.5; Noisy data***

## I. INTRODUCTION

Skype was developed in 2003 based on a Peer-to-Peer (peer network) for VoIP clients. It can be almost seamless NAT and firewall traversal, and its voice quality is much better than other VoIP client software. It developed into a platform with over 600 million users and provided users with a variety of reliable services: video chat, multiplayer voice conferencing, multiplayer chat, file transfer, text chat, and network agents [1].

Skype is a typical application of P2P technology evolution to the mixture model, which combines features of both centralized and distributed. Skype uses centralized network structure at the network edge node and distributed network architecture in the super-nodes. This P2P way connection consumes a huge bandwidth and even more when they become super-nodes. While Skype's NAT traversal feature makes information security is facing great challenge..

## II. RELATED WORK

Traffic classification is an essential part of the network traffic management which allows time-sensitive packets meet their performance goals and stops malicious traffic from spreading. Most of the existing research in the literature focuses on automatic application recognition in general classification. However, the traditional methods such as based on port numbers and deep packet inspection for patterns are not useful. In [2] Philippe BIONDI et al. using reverse engineering analysis of Skype protocol implementation details, pointing out that the contents of all communications are encrypted. One of the most successful approaches to classify the network traffic is using Machine Learning techniques. There has been a few research papers describing these techniques applied to different types of traffic and these methods are proven to be the most effective for classifying network traffic. Haffner et al. employed AdaBoost, Hidden Markov, Naive Bayesian and Maximum Entropy models to classify network traffic into different applications [3]. In [4], Bonfiglio et al. use Naive Bayes approach and Pearsons Chi-Square test (packet inspection) together to reach an accuracy of close to 80% in detecting Skype traffic. In [5], Riyad Alshammari et al. find the C4.5 based approach performs much better than other algorithms on the identification of Skype traffic. In addition, they describe various techniques to fine tune the parameters and features to improve the performance. In [6], Duffy Angevine et al. compared the AdaBoost and C4.5 on Skype traffic identification of effects, the results show that both high accuracy rate and low false positive rate, but C4.5 more effective in identifying Skype traffic with the accuracy rate of 94%. Mohammad Jalali et al. compare Naïve Bayesian and Neural Networks in the identification of Skype traffic effects [7]. Neural Networks is more accurate than Naïve Bayesian in the classification of Skype traffic, but Neural Networks takes long time training, which is not suitable for online traffic identification.

## III. CLASSIFICATION MODEL

In this section, I describe the various components for classification. We define a network flow to be a sequence of packets with the same 5-tupple (source and destination IP addresses, source and destination port numbers and protocol number). Then we talk about the most important part of flow feature selection. Finally, I briefly describe the two classification algorithms.

### *A. Feature Selection*

Choosing good flow features is the most challenging and crucial part of designing a classifier. In [6], Duffy Angevine et al. use Duration, Number of bytes or packets in both forward and backward direction per second, Number of ACK packets, Protocol ID, Minimum or Maximum Packet length observed to classify internet traffic. But some features like Duration, Number of ACK and Protocol ID are not useful to classify traffic when we deal with TCP and UDP traffic separately. Therefore we use the feature selection in [7] and discard the feature of Flow Duration, a total of 16 selected flow features.

• Client flow - packets inter-arrival time (mean, variance, max, min)

• Server flow - packets inter-arrival time (mean, variance, max, min)

• Client flow - packets size (mean, variance, max, min)

• Server flow - packets size (mean, variance, max, min)

The reason for these choices is that Skype traffic has longer flow duration compared to most of network traffic. Also, because of the use of codecs for audio calls and time sensitivity of packets in Skype, these features can help distinguish Skype traffic.

### *B. Classification Algorithms*

Specific to the impact of noisy data, we choose the two effects machine learning methods C4.5 and Neural Networks. C4.5 is the best classification algorithm in [5] and [6] which is a representative method of Skype traffic identification. The Neural Networks is more stable than the Naive Bayes Skype for recognition in [7]. Theoretically, Neural Networks is not sensitive to noisy data, and C4.5 is sensitive to noisy data. We verified the sensitivity to noisy data of two algorithms by experiment.

#### *1) C4.5*

C4.5 is a decision tree based classification algorithm. A decision tree is a hierarchical data structure for implementing a divide-and-conquer strategy. It is an efficient non-parametric method that can be used both for classification and regression. In non-parametric models, the input space is divided into local regions defined by a distance metric. In a decision tree, the local region is identified in a sequence of recursive splits in smaller number of steps. A decision tree is composed of internal decision nodes and terminal leaves. Each node m implements a test function $f_m(x)$ with discrete outcomes labeling the branches. This process starts at the root and is repeated until a leaf node is hit. The value of a leaf constitutes the output. In the case of a decision tree for classification, the goodness of a split is quantified by an impurity measure. A split is pure if for all branches, for all instances choosing a branch belongs to the same class after the split. One possible function to measure impurity is entropy. If the split is not pure, then the instances should be split to decrease impurity, and there are multiple possible attributes on which a split can be done. Indeed, this is locally optimal, hence has no guarantee on finding the smallest decision tree. In this case, the total impurity after the split can be measured by equation 2. A more detailed explanation of the algorithm can be found in [8].

#### *2) Neural Networks*

An Artificial Neural Networks (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network.

## IV. EXPERIMENT AND RESULTS

In this section, I present the experiment steps and results obtained by running the classification algorithms on real data traffic.

### *A. Dataset*

Since Skype is a proprietary protocol, there is no one standard public data sets. Most of Skype flow detection algorithm based on Internet-based flow detection. Harbin Institute of Technology Network and Information Security Laboratory provided us with experimental environment. This dataset consists of network traffic from the campus network during a 40-hour period in April 2013, a total of 10G data. The dataset distribution is in table I.

TABLE I. DATASET DISTRIBUTION

| Protocol | Classify | Flow number |
|---|---|---|
| TCP | Skype | 19445 |
| TCP | Non-Skype | 23322 |
| UDP | Skype | 8534 |
| UDP | Non-Skype | 10015 |

### *B. Experiment*

For the sake of the accuracy, we test TCP and UDP dataset separately. Firstly, we have chosen 20000 TCP flows (with 10000 Skype flows) randomly as TCP testing dataset for our experiment. Then we chose 20000 TCP flows (with 10000 Skype flows) as TCP training dataset 0. And we added on the basis of 5%, 10%, 15%, 20%, 25% and 30% of the noise data as TCP training dataset 1-6. Experiment is divided into seven times, respectively, each training C4.5 and neural networks with TCP training dataset 0-6, using the same TCP testing dataset to test the recognition effect. Even more, we use the same treatment for UDP flow. But differently, we have chosen 10000 UDP flows (with 5000 Skype flows) randomly as UDP testing dataset for our experiment. And we added on the basis of 5%, 10%, 15%, 20%, 25% and 30% of the noise data as UDP training dataset 1-6. Experiment is also divided into seven times, respectively, each training C4.5 and neural networks with UDP training dataset 0-6, using the same TCP testing dataset to test the recognition effect.

### *C. Result*

In this section we analyze the results from out experiment. The precision of C4.5 and Neural networks on

TCP flow is showed in Figure 1. And the precision of C4.5 and Neural networks on TCP flow is showed in Figure 2.

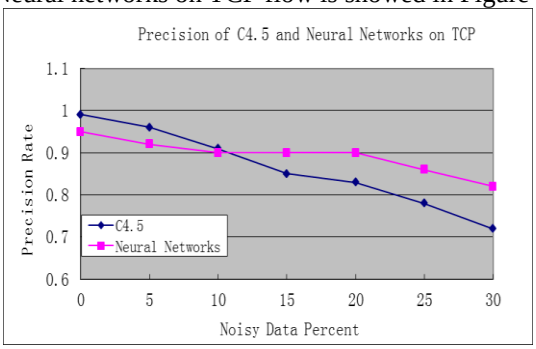


Figure 1. Example of a ONE-COLUMN figure caption.

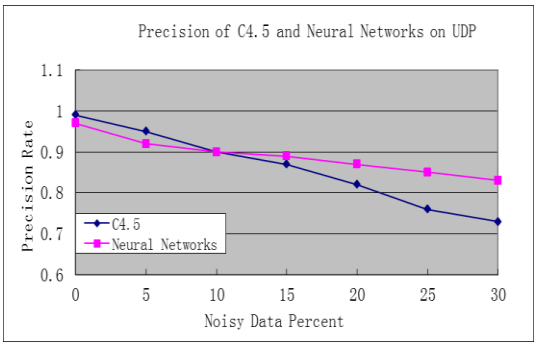


Figure 2. Precision of C4.5 and Neural networks on UDP

From figure 1, we can see that Neural Networks has a large advantage over C4.5 in percision rate on TCP when the noisy data percent is high. The advantages of the Neural Networks is its stability. C4.5 is sensitive to noise data, the percision rate drops quickly when the noisy data percent increases. From figure 2, we can also see this phenomenon on UDP. The percision rate line of the two algorithms across when the noise data percent reaches about 10%. C4.5 is better than Neural Networks when the noise data percent is less than 10% while Neural Networks than C4.5 when the noise data percent is greater than 10%.

TABLE II. EXECUTION TIMES FOR C4.5 AND NEURAL NETWORKS IN SECONDS ON TCP

| Training Dataset | C4.5 | Neural Networks |
|---|---|---|
| TCP dataset 0 | 3.68 | 127.25 |
| TCP dataset 1 | 3.2 | 133.09 |
| TCP dataset 2 | 2.75 | 143.57 |
| TCP dataset 3 | 2.88 | 143.16 |
| TCP dataset 4 | 2.93 | 141.87 |
| TCP dataset 5 | 3.16 | 137.71 |
| TCP dataset 6 | 3.23 | 134.14 |

TABLE III. EXECUTION TIMES FOR C4.5 AND NEURAL NETWORKS IN SECONDS ON UDP

| Training Dataset | C4.5 | Neural Networks |
|---|---|---|
| UDP dataset 0 | 1.84 | 62.42 |
| UDP dataset 1 | 2.02 | 64.59 |
| UDP dataset 2 | 1.91 | 58.31 |
| UDP dataset 3 | 2.28 | 65.8 |
| UDP dataset 4 | 2.03 | 60.21 |
| UDP dataset 5 | 2.16 | 61.76 |
| UDP dataset 6 | 2.35 | 66.46 |

From table II and table III, we can see that the execution time for Neural Networks can be over 2 minites while the execution time for C4.5 only in 3 seconds. Neural Networks takes a long time to execute and may not be effective for online traffic classification.

## V. CONCLUSIONS

In this paper, we have employed C4.5 and Neural Networks both supervised learning algorithms for classifying Skype on both TCP and UDP. Result so far show that both approaches perform with a very high detection rate and a low false positive rate when the feature set is employed. These results also indicate that the features selected to represent the traffic seem to be sufficient as well.

In summary, in this work, we have shown that it is possible to detect Skype TCP and UDP traffic using flow features. Indeed more analysis on the noisy data influence to the two algorithms. In this case, we have the worst performance of 82% detection rate achieved for both TCP and UDP based Skype classification by Neural Networks.

However, Neural Networks takes a long time to execute and may not be effective for online traffic classification. Therefore, C4.5 would be more beneficial in real-time traffic classifiers when the noise data percent is less than 10%. If the training dataset has too many noisy data (greater than 10%), Neural Networks is an effective method on detecting Skype traffic.

## ACKNOWLEDGMENT

This research was partially supported by the National Basic Research Program of China (973 Program) under grant No. 2011CB302605, the National High Technology Research and Development Program of China (863 Program) under grant No. 2011AA010705, the National Science Foundation of China (NSF) under grants No. 61100188 and No. 61173144, the National Key Technology R&D Program of China under grant No. 2012BAH37B01. Thanks to the lab to provide me with a good learning environment, thanks to my supervisor's support and help.

## REFERENCES

[1] Skype web site, http://www.skype.com

[2] P. Biondi, F. Desclaux, “Silver Needle in the Skype.” Black Hat Europe’06, Amsterdam, the Netherlands, Mar. 2006.

[3] Haffner P., Sen S., Spatscheck O., Wang D., “ACAS: Automated Construction of Application Signatures”, Proceedings of the ACM SIGCOMM, pp.197-202, 2005.

[4] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, “Revealing skype traffic: when randomness plays with you”, ACM SIGCOMM Computer Communication Review, vol. 37, 2007, p. 48.

[5] Riyad Alshammari and A. Nur Zincir-Heywood “Machine Learning Based Encrypted Traffic Classification:Identifying SSH and Skype” Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium.

[6] Duffy Angevine and A. Nur Zincir-Heywood. “A preliminary investigation of skype traffic classification using a minimalist feature set”. In ARES, pages 1075¨C1079. IEEE Computer Society, 2008.

[7] Mohammad Jalali "Skype Traffic Classification: Naive Bayes or Neural Networks". Report submitted April 2010.

[8] Alpaydin, Ethem, “Introduction to Machine Learning”, MIT Press, October 2004.