Evaluating Application–Layer Classification Using a Machine Learning Technique Over Different High Speed Networks

Sven Ubik, Petr Žejdl
CESNET – Czech academic network operator
Prague, Czech Republic
Email: {ubik,zejdlp}@cesnet.cz

***Abstract*—Application–layer classification is needed in many monitoring applications. Classification based on machine learning offers an alternative method to methods based on port or payload based techniques. It is based on statistical features computed from network flows. Several works investigated the efficiency of machine learning techniques and found algorithms suitable for network classification. A classifier based on machine learning is built by learning from a training data set that consists of data from known application traces. In this paper, we evaluate the efficiency of application-layer classification based on C4.5 machine learning algorithm used for classification network flows from different high speed networks, such as 100 Mbit, 1 Gbit and 10 Gbit networks. We find a significant decrease in the classification efficiency when classifier built for one network is used to classify other network. We recommend to build classifier from data collected from all available networks for best results. However, if different networks are not available, good results can be obtained from data traces to the commodity Internet.**

Keywords-Traffic Classification; Network Flows; Machine Learning; High speed networks; Passive network monitoring.

I. Introduction

Packet classification is needed in many monitoring applications. For example, to reduce or distribute traffic to multiple processes, or to compute statistics about specific hosts for traffic accounting. In classification, packets are marked as belonging to classes, which are then treated separately, for statistics or performance reasons. Routers classify packets to determine which class they belong to. Proper packet classification is the base for traffic shaping and traffic limiting. These are important for Internet service providers to fairly distribute the bandwidth among the users.

Port-based and deep-packet-inspection-based classification (a type of classification that is based on protocol knowledge) is still widely used. However, it is less effective or completely ineffective if tunneling or encryption is in place, an application uses dynamic port allocation, non-standard port numbers or anonymisation, or if payload removal is used. Several applications, particularly P2P applications, hide the traffic behind the ports of different applications or use dynamic port allocation to avoid detection, making the port based classification ineffective [1].

The protocol knowledge is represented by a protocol decoding automata or payload patterns and signatures, often hardcoded into the classification library for performance reasons. Maintaining such a library can be difficult. Adding a new protocol or changing an existing one requires library rewriting. Adding a proprietary or unknown protocol also requires additional work such as protocol reverse engineering. Snort [2] is a popular example of signature based classification system used for intrusion detection. It uses a database filled with signatures of known applications.

Classification based on machine learning algorithms offers an alternative method of application classification [3], [4], [5], [6]. Instead of using protocol knowledge, it is based on statistical features such as flow duration, number of packets, packet length, inter-arrival times, etc. A classifier based on machine learning is built by learning from a training set that consists of known application data. The built classifier (called model) can decide whether the input data belongs to the same class it learnt or is unknown. Any captured data can be used as a training set, therefore, it is also possible to identify a proprietary or unknown protocol.

This paper is organized as follows: Section II discusses the related work. Section III describes the dataset used in the classification experiments. Section IV presents the experimental results. Section V concludes our work and outlines future work.

II. Related work

Several papers have investigated the possibilities of using machine learning algorithms for application classification. The efficiency and performance of different machine learning techniques based on flow data are compared in [5], [3], [4]. The authors of [7], [8] investigate the performance of machine learning algorithms for classifying encrypted network traffic. In [9], the authors test the machine learning algorithms for classifying Skype communication.

The results indicated that C4.5 [10] machine learning algorithm outperforms others. It is possible to achieve very good classification accuracy with a detection rate of about 98% and a false positive rate of less than 0.5% even for encrypted traffic.

Various machine learning algorithms are well tested; however, we are missing the answer for the question: What

happen, when classifier built for one network is used to classify a different network? This is a real life situation, when a user uses some classification box ready for classifying one network and moves the box to the different network.

Machine learning algorithms are sensitive to the quality of test data used in their learning process; using them in the different environment may change classification accuracy.

In this paper, we try to answer that question by evaluating the efficiency of application-layer classification based on C4.5 machine learning algorithm. We test several classifiers on five different networks with different speeds.

III. Data preparation

This section describes the preparation of data used in out experiments.

A. Captured Tracefiles

CESNET [11] is a Czech academic network operator which operates the Czech National Research and Educational Network (NREN) with connections to several international networks. We used data traces collected from four different networks with different speeds. Three of them were captured at the border links of the CESNET network. The fourth is publicly available WITS [12] network tracefile captured at the border of the University of Waikato network.

Table I summarizes network tracefiles used in this paper. Network description follows:

- ASM-IX – Amsterdam Internet Exchange [13].
- GEANT2 – European network for research and educational community [14].
- Telia – TeliaSonera, our supplier of commodity Internet connectivity [15].
- Waikato – University of Waikato network [12].

The ASM-IX, GEANT2 and Telia tracefiles were captured in February, 2010. Tracefile from Waikato University was captured in July, 2005.

B. Application classes

The classified application's tracefiles are used as learning data for the classification algorithm. Properly classified tracefiles imply the classification quality (accuracy). Getting classified tracefiles is not an easy task. Several options exist:

- Using publicly available tracefiles.
- Creating synthetic tracefiles
- Collecting tracefiles from a network and classifying them

Every option has its pros and cons. Quality of publicly available tracefiles cannot be guaranteed. They are anonymised and without payload disabling any further classification. Creating synthetic tracefiles is possible, however, it is difficult to properly model the network environment. Accuracy of a classifier built with this kind of data can be reduced. Gathering tracefiles from a real network seems to be the best option, but it is difficult to find useful application flows in the captured traffic.

We are using tracefiles gathered from a real networks and we classify application flows based on several well-known port numbers defined by IANA [16]. Table II shows selected port numbers and its application flows.

TABLE I
Tracefile characteristics

Tracefile	Network speed	Captured packets	Duration
ASM-IX	1 Gb/s	114,766,862	33 minutes
GEANT2	10 Gb/s	120,000,683	51 minutes
Telia	10 Gb/s	251,624,735	35 minutes
Waikato	100 Mb/s	33,978,089	12 hours

TABLE II
Application classes based on well-known port numbers

Port number	Application Flow	Description
22	SSH	The Secure Shell
25	SMTP	Simple Mail Transfer
53	DNS	Domain Name Server
80	HTTP	World Wide Web HTTP
123	NTP	Network Time Protocol

We assume that majority of the flows belong to the expected applications. It can be expected, that some of the flows are from different applications due to various reasons (e.g., tunneling). These flows will be misclassified and reduce classification accuracy; therefore, the results represent a lower bound. This is the same approach as used in [5].

C. Learning data

Table III shows the flow distributions of chosen application traces (port numbers) in the tracefiles. Selected flows cover about 80% for Waikato traces (captured in 2005). GEANT2 and AMS-IX captured in 2010 cover roughly 40% and Telia (commodity Internet traffic) account for 25%. The results suggest significant difference in flow distribution between the educational networks and commodity Internet traffic and may also suggest the change of traffic distribution in the time period from 2005 to 2010.

Because captures tracefiles contain up to several million flows, we used stratified sampling to sample 4,000 flows randomly per class and trace. We use 5 application classes composing 20,000 flows per trace together. However, for some of the classes, there were fewer available flows in the tracefile. We also created traffic MIX consisting of all sampled flows from all available traces. Table IV summarizes the number of flows in the traces.

TABLE III
FLOW DISTRIBUTION PER TRACEFILE

	Network trace			
Class	ASM-IX	GEANT2	Telia	Waikato
SSH	0.02	0.03	0.4	0.5
SMTP	0.3	0.7	0.3	2.4
DNS	33.6	27.8	20.0	46.0
HTTP	6.3	10.6	3.1	20.3
NTP	1.3	1.1	1.4	11.2
Total	41.5	40.2	25.3	80.5

TABLE IV
SAMPLED FLOW CHARACTERISTICS

Tracefile	Flows
ASM-IX	15,321
GEANT2	13,759
Telia	20,000
Waikato	20,000
MIX	69,080

D. Classification Features Selection

Each training or testing data set consists of packet flows that are represented by a set of features (flow attributes). There is a large number of available attributes and their information value varies. Based on observation made in [5], [9], [7], [8], we selected the following features to characterize an individual flow and maintain the similar classification conditions:

- Protocol
- Flow duration
- Number of transferred packets
- Number of transferred bytes
- Packet length distribution
- Inter–arrival time distribution

We used Netmate [17] (Network Measurement and Accounting System) tool for packet trace processing and flow creation. We consider only UDP and TCP flows with at least one packet transferred and at least one byte of payload. TCP flows are terminated by TCP connection termination or by flow timeout. UDP flows are terminated only by flow timeout. We set flow timeout to 60 seconds for fast classification response. The use of longer flow timeouts is possible; however, there is longer classification response and [5] shown that in most cases the accuracy is unaffected by using short timeouts (e.g., 60 seconds).

Flows are bidirectional and the first packet seen by the tool determines the forward direction. The number of transferred packets, bytes and distributions are computed for both flow directions. Distribution statistic consists of minimum, mean, maximum and standard deviation statistics. Flows are determined by the source IP address and source port number, destination IP address and destination port number.

IV. EXPERIMENTAL RESULTS

All machine learning experiments were done by the WEKA – Data Mining Software in Java, version 3.6.2 [18].

A. Evaluation Method

We use the common n–fold cross validation method [18]. Here, a number of folds n is specified. The dataset is randomly reordered and then split into n folds of equal size. In each iteration, one fold is used for testing and the other $n-1$ folds are used for training the classifier. The test results are collected and averaged over all folds. This gives the cross-validation estimate of the accuracy. We use $n = 10$.

There are various approaches to determine the performance of classifiers. The most frequently used is counting the proportion of correctly predicted examples in an unseen dataset. This metric is called accuracy, which is also $1 - ErrorRate$. Both terms are used in literature. We use the following performance metrics:

- Accuracy – the number of correctly classified instances over the total number of instances.
- Precision – the number of correctly classified instances of class X over the total number of instances classified as belonging to class X.
- Recall (true positive rate) – the number of correctly classified instances of class X over the total number of instances of class X.

For the overall performance metrics of the classifier, weighted average of particular performance metrics are used.

B. Measured performance

The overall performance metrics per trace are shown in Table V. All test data traces have very high accuracies, precision and recall values in the range from 96.3% to 99.6%. These results are expected and confirm the excellent performance of C4.5 algorithm found in literature.

Now we use a classifier built for one of networks and test it with the other networks. We want to test the classifiers behavior between the particular networks. This is based on real life situation, when an user uses classifier for one network and is asked to classify a different network. Machine learning algorithms are sensitive to the quality of test data used in their learning process and using them in the different environment may change classification accuracy.

TABLE V
ACCURACY

	MODEL				
Performance	ASM-IX	GEANT2	Telia	Waikato	MIX
Accuracy	97.6	99.0	96.3	99.6	97.8
Precision	97.4	99.0	96.3	99.6	97.8
Recall	97.6	99.0	96.3	99.6	97.8

We have five traces (including mixed trace) and we test a classifier built for one trace (model) with the other traces. The results are summarized in Table VI.

TABLE VI
ACCURACIES FOR COMBINED TRACES AND CLASSIFIERS

	MODEL				
DATA	ASM-IX	GEANT2	Telia	Waikato	MIX
ASM-IX	x	87.38	87.90	88.18	98.69
GEANT2	94.00	x	92.86	94.36	99.52
Telia	74.04	64.63	x	70.15	98.51
Waikato	75.98	79.23	95.06	x	99.85
MIX	84.12	80.86	94.02	94.02	x
Average	82.04	78.03	92.46	86.68	99.14

The results show drop in accuracy for all models except model built from mixed traces. The accuracy drops to 84.80% on average with minimum in 64.63% and maximum in 95.06%. The best results are obtained with model built from mixed data trace. With MIX model we are able to achieve accuracy of 99.14% in a range from 98.51% to 99.85%.

Good result is obtained for Telia model. It is a classifier built from commodity Internet traces. The average accuracy is 92.46%.

V. CONCLUSION AND FUTURE WORK

In this paper, we evaluated the efficiency of application-layer classification based on C4.5 machine learning algorithm used for classification network flows from five high speed networks. We used captured tracefiles from 100 Mbit, 1 Gbit and two 10 Gbit networks.

Experimental results confirmed high accuracies over 96.3% for C4.5 algorithm when used for particular network data traces.

When classifier built for one network trace is used for classification of other network, the accuracy drops to 84.80% on average. Good result is obtained with classifier built for commodity Internet traffic, with the accuracy of 92.46% on average.

The best results are obtained with mixed data trace consisting of the data collected from all tested networks. In this experiment we are able to achieve classification accuracy of 99.14% on average.

The results show the classifier sensitivity to the used network. When classifier is used for different network decrease in the efficiency should be expected. The best way to build classifier is to combine traces from different networks. When different networks are not available, good results can be obtained from traces from commodity Internet traffic.

The next research will involve investigation of similarities between networks in order to create a robust network classifier. We will also perform feature reduction to find most descriptive feature set and test a wider range of application classes, particularly applications based on dynamic ports like peer-to-peer sharing applications and Skype.

ACKNOWLEDGMENT

This task was done as part of the GN3 project and supported by the EC, Grant Agreement No. 238875.

REFERENCES

[1] T. Karagiannis, A. Broido, M. Faloutsos, and K. C. Claffy, "Transport layer identification of p2p traffic," in *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, 2004, pp. 121–134.

[2] M. Roesch, "Snort - lightweight intrusion detection for networks," in *LISA '99: Proceedings of the 13th USENIX conference on System administration*. Berkeley, CA, USA: USENIX Association, 1999, pp. 229–238.

[3] Y. Wang and S.-Z. Yu, "Supervised learning real-time traffic classifiers," *Journal of Networks*, vol. 4, no. 7, pp. 622–629, September 2009. [Online]. Available: http://www.academypublisher.com/ojs/index.php/jnw/article/view/1178

[4] T. T. T. Nguyen and G. J. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 1-4, pp. 56–76, 2008.

[5] N. Williams, S. Zander, and G. Armitage, "Evaluating machine learning algorithms for automated network application identification," Swinburne University of Technology, Melbourne, Australia, CAIA Technical Report 060410B, March 2006. [Online]. Available: http://caia.swin.edu.au/reports/060410B/CAIA-TR-060410B.pdf

[6] ——, "A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 5, pp. 5–16, 2006.

[7] R. Alshammari and A. N. Zincir-Heywood, "A preliminary performance comparison of two feature sets for encrypted traffic classification," in *Proc. International Workshop on Computational Intelligence in Security for Information Systems CISIS 2008*. Springer Publishing Company, Incorporated, October 2008, pp. 203–210.

[8] ——, "Investigating two different approaches for encrypted traffic classification," in *Proc. Sixth Annual Conference on Privacy, Security and Trust, PST 2008*. Fredericton, New Brunswick, Canada: IEEE, 2008, pp. 156–166.

[9] D. Angevine and A. N. Zincir-Heywood, "A preliminary investigation of skype traffic classification using a minimalist feature set," in *Proc. Third International Conference on Availability, Reliability and Security ARES'08*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 1075–1079.

[10] R. Kohavi and J. R. Quinlan, "Data mining tasks and methods: Classification: decision-tree discovery," in *Handbook of data mining and knowledge discovery*. New York, NY, USA: Oxford University Press, Inc., June 2002, pp. 267–276.

[11] CESNET, Czech Academic Network Operator. (Last accessed: April, 2010). [Online]. Available: http://www.ces.net/

[12] WITS: Waikato Internet Traffic Storage. (Last accessed: April, 2010). [Online]. Available: http://www.wand.net.nz/wits/

[13] AMS-IX – Amsterdam Internet Exchange. (Last accessed: April, 2010). [Online]. Available: http://www.ams-ix.net/

[14] GEANT2 – European network for research and educational community. (Last accessed: April, 2010). [Online]. Available: http://www.geant2.net/

[15] TeliaSonera, Network service company. (Last accessed: April, 2010). [Online]. Available: http://www.telia.net/

[16] IANA – Internet Assigned Numbers Authority (IANA), Well Known Port Numbers. (Last accessed: April, 2010). [Online]. Available: http://www.iana.org/assignments/port-numbers{pokusny}

[17] *NetMate Software*, network measurement and accounting system. (Last accessed: April, 2010). [Online]. Available: http://www.ip-measurement.org/

[18] Weka Software, Version 3-6-2, Data Mining Software in Java. The University of Waikato. (Last accessed: April, 2010). [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/