Online Classification of Network Flows

Mahbod Tavallaee, Wei Lu and Ali A. Ghorbani
Faculty of Computer Science, University of New Brunswick
Fredericton, NB E3B 5A3, Canada
{m.tavallaee, wlu, ghorbani}@unb.ca

Abstract

Online classification of network traffic is very challenging and still an issue to be solved due to the increase of new applications and traffic encryption. In this paper, we propose a hybrid mechanism for online classification of network traffic, in which we apply a signature-based method at the first level, and then we take advantage of a learning algorithm to classify the remaining unknown traffic using statistical features. Our evaluation with over 250 thousand flows collected over three consecutive hours on a large-scale ISP network shows promising results in detecting encrypted and tunneled applications compared to other existing methods.

1. Introduction

Accurate classification of network traffic has received a lot of attention due to its important roles in many subjects such as network planning, QoS provisioning, class of service mapping, to name a few. Traditionally, traffic classification relied to a large extent on the transport layer port numbers, which was an effective way in the early days of the Internet. Port numbers, however, provide very limited information nowadays due to the increase of HTTP tunnel applications, the constant emergence of new protocols and the domination of P2P networking applications. An alternative way is to examine the payload of network flows and then create signatures for each application. This approach, however, is limited by the fact that classification rules must be updated whenever an application changes, and privacy laws and encryption can completely make the payload inaccessible. Observing daily traffic on a large-scale WiFi ISP network, Fred-eZone [1], over a half year period (from Jun. 2007 to Dec. 2007), we found that even exploring the flow content examination method, there are still about 40% of network flows that cannot be classified into specific applications by state-of-the-art signature based classifiers , i.e., 40% network flows are labeled as unknown applications.

To overcome the limitations of the signature-based methods, some researchers have proposed the idea of traffic classification based on the statistical features [41, 30, 42, 9, 31, 14, 38, 15, 8, 26]. Although the proposed statistical approaches provide new capabilities such as shorter classification time and classification of encrypted packets, they are far from completed yet due to the limited number of applications they can identify (e.g. only 3 in [14] and 10 in [9]) and the rough application scopes (e.g. BLINC in [26] attempts to identify the general P2P traffic instead of the specific underlying P2P applications like eDonkey, BitTorrent).

Addressing the limitations of the aforementioned approaches, we propose a hybrid mechanism for online classification of network traffic, in which we apply a signature-based method at the first level, and then we take advantage of a learning method to classify the remaining unknown traffic based on the network statistical features. In order to choose an appropriate classifier, we selected some of the most popular classification methods implemented by Weka [6], and performed some evaluations to compare the accuracy, learning time, and classification time of the selected algorithms. The experimental result showed that Adaboost [20] and Bagging [10] outperforming other methods in the accuracy; however, they both have a long learning time even though the data set was very small and only contained one-hour traffic. As a result, we picked J48 decision tree [37] since it has a high accuracy while maintaining a reasonable learning time. In addition, decision trees are shown to have a reasonable height and needs a few comparisons to reach a leaf which is the final label, and therefore have a very short classification time. Taking advantage of J48 decision tree as the statistical classifier, we evaluated our proposed hybrid system on a real network. Our evaluation on a large-scale ISP network, IRANET [4], shows promising results compared to other methods in detecting encrypted and tunneled applications.

The rest of the paper is organized as follows. Section 2 introduces related work, in which we summarize existing traffic classification approaches in terms of two cate-

978-0-7695-3649-1/09 $25.00 © 2009 IEEE
DOI 10.1109/CNSR.2009.22

IEEE computer society

gories, namely signature-based and statistical traffic classifiers. Our proposed hybrid traffic classification scheme will be explained in Section 3. Section 4 presents the experimental evaluation of our approach and discusses the obtained results. Finally, in Section 5, we draw conclusions and discuss future work.

2. Related Work

Early common techniques for identifying network applications rely on the association of a particular port with a particular protocol [32]. Such a port number based traffic classification approach has been proved to be ineffective due to: 1) the constant emergence of new peer-to-peer networking applications that IANA does not define the corresponding port numbers [3]; 2) the dynamic port number assignment for some applications (e.g. FTP); and 3) the encapsulation of different services into a same application (e.g. chat or steaming can be encapsulated into the same HTTP protocol). To overcome this issue, recently there have been significant contributions towards traffic classification. Having done a thorough study of research in traffic classification, we have classified the existing approaches into two categories, namely signature-based and statistical. In the remaining of this section, we introduce each category with some typical examples and discuss the limitations for the existing techniques.

2.1. Signature-Based Traffic Classifier

An alternative to traditional port number based application classification is to inspect the content of payload and seek the deterministic character strings for modeling the applications. In [22], Gummadi et al. develop a signature model for KaZaA workload characterization through analyzing a 200-day trace of over 20 terabytes of Kazaa P2P traffic collected on a campus network. In [40], Sen et al. analyze the application layer protocols and then generate the signatures of a few P2P applications. Although the protocol semantic analysis improves the accuracy of signatures, it makes the real-time analysis of the backbone traffic impossible since the underlying assumption is that every packet is being inspected. In their consequent work [39], Sen et al. examine available specification and packet-level traffic traces for constructing application layer signatures, and then based on these signatures, P2P traffic are filtered and tracked on high-speed network links. Evaluation results show that their approach obtain less than 5% false positives and false negatives.

Not only obtained from unencrypted traditional applications, signatures can also be extracted from the encrypted traffic. Ehlert et al. examine the hexadecimal patterns in the Skype packet traces during the initial communication setup phase [17]. Moreover, in [8], Bernaille et al. characterize the application signatures during the early stage of protocol handshakes for mixed network traffic that are carried over an encrypted SSL connection. Although looking for a pattern through inspecting the packet payload content is an effective approach even for encrypted traffic, it usually fails to classify application types for those applications (e.g. Gnutella) with variable-length packets in their protocol handshakes. Other typical examples of using payload content signatures for traffic classification include [33, 16, 23, 25]. Based on the payload signatures, the application classifier can obtain an extreme high accuracy. However, the biggest limitation is that all the above mentioned approaches focus on identifying only one single application (e.g. KaZaA in [22] or Skype in [17]) or one application group (e.g. Chat traffic identification in [16] or P2P traffic identification in [39]).

2.2. Statistical Traffic Classifier

The usage of statistical properties for network traffic classification or at least traffic behavioral modeling is not new. The early studies on the subject can be traced back to the seminal report by Paxson et al. [34, 35], in which some statistical variables (e.g. packet length, inter-arrival times and flow duration) have been proved to be suitable to express the behavior of a few protocols. With the increase of new appeared network applications, the problem now has become to associate a given flow, characterized by a set of statistics, to a specific application. As a result machine learning techniques can naturally achieve such a classification task through their training and learning capabilities. In [41], Williams et al. conduct a preliminary performance comparison of 5 machine learning algorithms for practical IP flow classification. Given the same features and flow trace, it was claimed that different machine learning algorithms provide very similar classification accuracy. The basic 5 features proposed in the paper include: protocol, flow duration, flow volume in bytes and packets, packet length, and inter-arrival time between packets.

In [14], Crotti et al. present a flow classification mechanism based on three simple properties of the captured IP packets: size of packets, inter-arrival time and arrival order. A new structure, called protocol fingerprints, is defined to express the three trace properties in a compact and efficient way. According to an anomaly score, the protocol fingerprints allow the measurement of "how far" an unknown flow is from the basic characteristics of each protocol. A simple classification algorithm is then applied to classify flows dynamically when packets pass through the classifier, deciding if a flow belongs to a given application layer protocol, or if it was generated by an "unknown" (i.e., non-fingerprinted) protocol. As claimed in the their evaluation, the limitation of the approach is that it can identify only 3 protocols, namely SMTP, POP3 and HTTP.

In [7], Bernaille et al. propose a technique that relies on the observation of the first five packets of a TCP connection to identify the application. It was claimed that the size of the first few packets is a good predictor of the application associated with a flow because it captures the application's negotiation phase, which is usually a pre-defined sequence of messages and distinct among applications. The result opens a range of new possibilities for online traffic classification since most classification techniques need the statistics of the entire flow to start the traffic classification (e.g. duration and number of packets in a flow), which limits their applicability for online classification. Bernaille et al. then illustrate their idea in [9] for an online traffic classification based on the first few packets of a TCP connection. Specifically their approach consists of two phases: an offline learning phase and an online classification phase. They employ three clustering methods (K-means and Gaussian Mixture Models on an Euclidean space, and Spectral clustering on Hidden Markov Models), and connections not belonging to any cluster are identified as unknown. As a result, it was claimed by the authors that the approach has the potential to detect new applications or new modes of operation of known applications.

In [30], McGregor et al. present a machine learning based methodology to break the traffic trace into different trace clusters in which each cluster has different trace characteristics. Typical clusters include bulk transfer, single and multiple transactions and interactive trace. For the clustering, they applied Expectation-Maximization (EM) algorithm with a set of statistical features including packet size, inter-arrival statistics, byte counts, connection duration, the number of transitions between transaction mode and bulk transfer mode, and the time spent in bulk transfer and in transaction mode. The evaluation results basically group traffic into different application types, like bulk transfer, small transactions, etc. However, further work is necessary in order to obtain the more specific applications groups.

Other typical examples of statistical traffic classifiers can be found in [42, 15, 29, 19, 28, 18, 38, 31]. Although all these techniques show their capability for traffic classification to some extent, the number of applications they can identify is very limited. In addition, the definition of application classes is very rough and is not precise enough to obtain the fine-grained applications. An exception work is conducted by Erman at al. [19], in which they employed a semi-supervised learning technique to classify over 29 applications.

3. Hybrid Traffic Classification Scheme

As explained in the previous section, to overcome the shortcomings of port-based traffic classification, researchers have proposed new approaches based on either the content of the payload or the statistical features such as connection duration. Since the statistical features are very general and restricted, statistical classifiers can only identify general categories of applications (e.g. Web, P2P, Mail). In contrast, signature-based classifiers are able to detect a wide range of applications providing the specific signatures. Besides, They are completely accurate and generate almost no false positives. These advantages have made the signature-based traffic classifiers very popular. This approach, however, is limited by the fact that classification rules must be updated whenever an application changes, and privacy laws and encryption can completely make the payload inaccessible.

Conducting a thorough analysis of state-of-the-art signature based classifiers, we observed that depending on the type of network (e.g. Universities, ISPs, Enterprises) between 20% to 40% of the traffic is still unknown. This unknown traffic mainly consists of new applications, variation of old applications or encrypted traffic. To overcome this issue, we have proposed a hybrid traffic classification method using signature-based approach in the first level and then applying a learning approach to classify the unknown portion based on the network statistical features.

3.1. Statistical Traffic Classifier

As the first step to have an effective statistical traffic classifier, we should extract robust network features that have the potential to distinguish among heterogeneous applications. Since most current network management systems use network flow data (e.g. netflow, sflow, ipfix) as their information sources, we focus on features generated based on these flows. The name and description of the applied features are listed in Table 1. All the selected features have two important characteristics: 1) They have shown to be effective in distinguishing different applications. 2) They can be calculated in real time, and impose no delay to the classifier.

In order to process the packets, classify them into flows and extract the features, we used a commercial network security management tool, QRadar [5]. Flows generated by QRadar, qflows, are defined by source IP, destination IP, source port, destination port, and protocol. Each flow will be terminated by a timeout which is set to 60 seconds in our experiments. Except for that, TCP flows can also be terminated upon proper connection teardown.

Having specified the applied network features, we focus on the selection of a high-performance classifier. Taking advantage of a signature-based classifier in our hybrid sys-

Table 1. Applied flow-based features

SrcIP	source IP address
DstIP	destination IP address
SrcPort	source port number
DstPort	destination port number
ProtocolName	the name of the protocol used in transport layer
ConnectionDuration	the period of time in milliseconds during which the connection is alive
SrcBytes	the total number of bytes sent from the source to the destination
DstBytes	the total number of bytes sent from the destination to the source
SrcPackets	the total number of packets sent from the source to the destination
DstPackets	the total number of packets sent from the destination to the source
SrcBytes/DstBytes	the ratio of "SrcBytes" to "DstBytes"
SrcPackets/DstPackets	the ratio of "SrcPackets" to "DstPackets"
SrcBytes/SrcPackets	the ratio of "SrcBytes" to "SrcPackets"
DstBytes/DstPackets	the ratio of "DstBytes" to "DstPackets"

tem, we always have access to an up-to-date training set to be used by the classifier. As a result, we limited ourselves to supervised algorithms since they usually have a better performance compared to unsupervised algorithms. In order to choose an appropriate classifier, we selected some of the most popular classification methods implemented by Weka [6] and performed some evaluations to compare the accuracy, learning time, and classification time. In the following, we briefly explain the algorithms we used in our experiments.

J48 Decision Tree is the Weka implementation of C4.5 [37] decision tree which builds decision trees from a set of training data in the same way as ID3 [36], using the concept of information entropy. However, it is made a number of improvement over ID3: 1) handling both continuous and discrete attributes; 2) handling training data with missing attribute values; 3) handling attributes with differing costs; and 4) pruning trees after creation.

Random Forest is a classifier that consists of many decision trees and outputs the class that is the mode of the classes output by individual trees [11].

Naive Bayes is based on the Bayesian theorem strong (naive) independence assumptions [24]. This classification technique analyzes the relationship between each attribute and the class for each instance to drive a conditional probability for the relationships between the attributes and the class.

Bayesian Network (BayesNet) is a probabilistic graphical model that represents a set of variables and their probabilistic independencies. Bayesian networks are directed acyclic graphs whose nodes represent variables, and whose missing edges encode conditional independencies between the variables. There exist many different algorithms to learn a Bayesian network, among which we selected K2. This Bayes Network learning algorithm uses a hill climbing algorithm restricted by an order on the variables [13].

Naive Bayes Tree (NBTree) is a hybrid of decision tree and Naive Bayes classifiers [27]. Designed to allow the accuracy to scale up with increasingly large training sets, the NBTree model is a decision tree of nodes and branches with Naive Bayes classifiers on the leaf nodes.

Bagging is a meta-algorithm to improve machine learning of classification and regression models in terms of stability and classification accuracy. It also reduces variance and helps to avoid overfitting [10]. In our experiments we applied this meta-algorithm to C4.5 decision tree [37].

AdaBoost , also called Adaptive Boosting, is a meta-algorithm which can be used in conjunction with many other learning algorithms to improve their performance [20]. It calls a classifier repeatedly, and on each round, the weights of each incorrectly classified example are increased, so that the new classifier focuses more on those examples.

In addition to the mentioned classification algorithms, we also tried support vector machines (SVM) [12], and Tree augmented naive Bayes (TAN) [21]. However, having a very high space or time complexity, they are not efficient to be used as the statistical classifier.

3.2. signature-based Traffic Classifier

As our signature-based traffic classifier we chose a powerful traffic classification software, MeterFlow [2], because of its availability to researchers. MeterFlow returns stateful deep packet classification results in real time, which means that it evaluates each and every packet while preserving context information relative to the flow and to other packets. Unlike existing packet classification technologies, MeterFlow extensively analyzes application-specific data in real time, allowing it to determine the content and identify the application of a single packet as well as the relationships between packets.

3.3. The Hybrid Traffic Classifier

In order to achieve reliable results, the statistical classifier needs to be provided with an up-to-date training set. To this end, we have defined learning time intervals, e.g. 1 day, at the end of which the statistical classifier will be trained by the latest training set. This training set is the known flows labeled by the signature-based traffic classifier in the previous day. Figure 1 illustrates the structure of the hybrid traffic classifier. In the first interval which we do not have any training set, we only rely on the labels from MeterFlow. The flows with known labels will be used as the training set for the statistical classifier in the next time interval. During

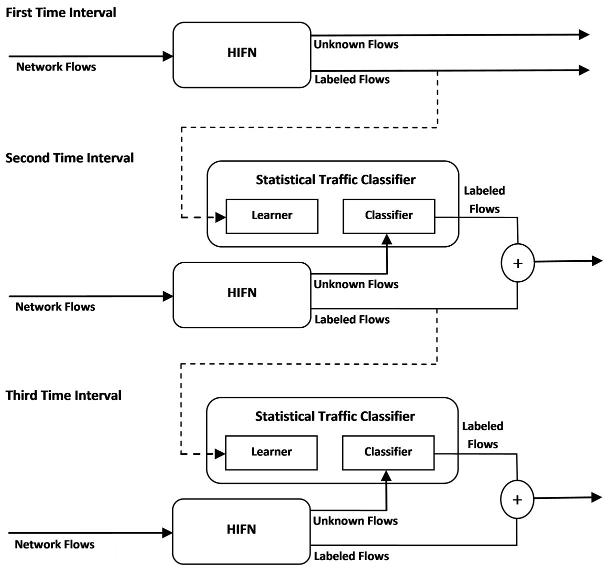

Figure 1. General structure of the hybrid traffic classifier

the second time interval, flows are given to MeterFlow to do the labeling. The unknown flows will then be given to the statistical classifier to be labeled based on the learned information in the previous time interval. Finally, the known flows from MeterFlow and the labeled flows by the statistical classifier will be mixed together and reported to the administrator as the application labels.

4. Experiments

4.1. Applied Statistical Classifier

In order to choose the most efficient classifier with respect to the accuracy and time complexity, we prepared a data set of 3-hour traffic from a large-scale ISP network, IRANET [4]. IRANET is a large-scale ISP providing Internet service to Iranian academic institutes. Table 2 summarizes the workload of the IRANET network over an hour. Having extracted the corresponding flows using QRadar, we fed the flows into MeterFlow to get the application labels. We then filtered out the unknown flows which were about 35% of total flows, and copied the rest (known flows) to an ARFF file to be used as a labeled data set by Weka. Table 3 illustrates general information of the prepared data set.

Having prepared the data set, we evaluated the performance of the algorithms described in Section 3.1 using 10-fold cross-validation. Figures 2 and 3 illustrates the accuracy and learning time of the applied methods, respectively. As it is shown in Figure 2, Adaboost and Bagging outperforming other methods in the accuracy; however, they both have a long learning time even though the data set is very small and only contains one-hour traffic. As a result, J48 seems to be the best choice since it has a high accuracy while maintaining a reasonable learning time.

Table 2. Workload of the IRANET network over an hour

Src. IPs	Dst. IPs	Flows	Packets	Bytes
17,456	11,585	84,082	2,184,236	1,074,802,236

Table 3. General information of the prepared data set

Number of features	14
Number of samples	161,079
Number of applications (labels)	89

4.2. Experimental Results

Although the decision tree based classifier achieves a high classification accuracy during the experimental evaluation with cross-validation, the results might be misleading due to the fact that in the hybrid detector the classifier should deal with real unknown flows. As a result, we conducted an online evaluation of the hybrid traffic classifier on a large-scale ISP network, IRANET [4]. In our experiments, we considered the learning time interval to be 1 hour, and evaluated the hybrid classifier during 3 hours. Table 4 illustrates the detailed information of the three hours of evaluation. As it is shown in Figure 1, in the first time interval, the hybrid classifier only relies on the result of MeterFlow. However, in the second and third time intervals, the portion of the flows which is remained unknown by MeterFlow are fed to the statistical classifier to be labeled based on the information learned in the previous time interval. As it is mentioned earlier, signature-based traffic classifiers such as MeterFlow are shown to be accurate since they label a flow only if it exactly matches the provided signatures. However, having no information about the unknown flows, we experienced some difficulties to evaluate the statistical classifier results. To this end, we randomly selected 100 flows from the second and third time intervals, and then using expert knowledge we compared real application names with the labels generated by the statistical classifier. Out of 100 flows, only 2 flows were incorrectly labeled. Having an in-depth analysis of these two flows, we found that they both belonged to the Skype application. However, since there was no flows labeled as Skype in the training set, the classifier forced them to be labeled as Yahoo application.

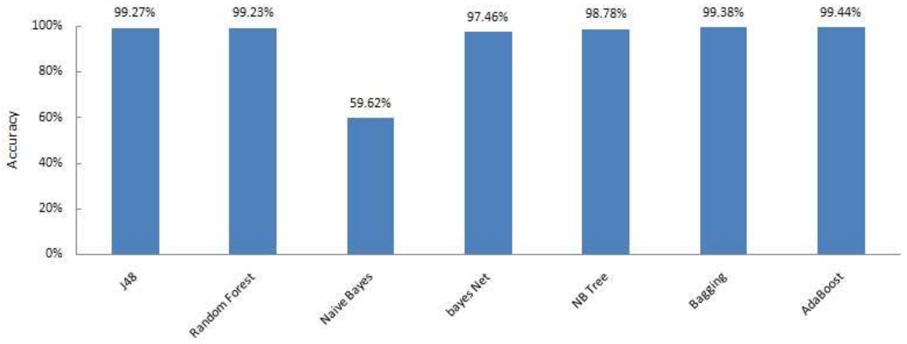

Figure 2. Accuracy of the applied classification algorithms

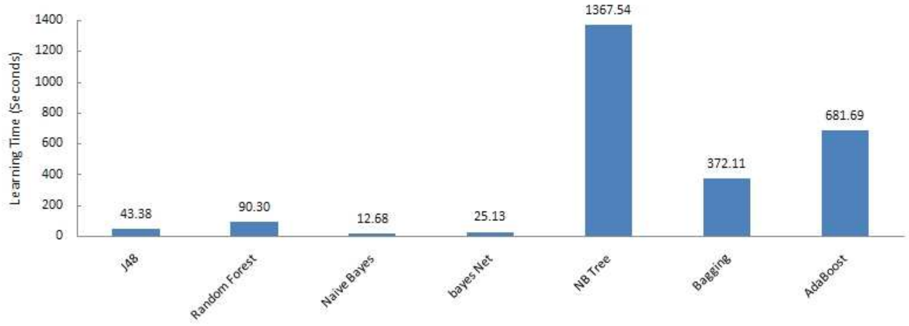

Figure 3. Learning time of the applied classification algorithms based on seconds

5. Conclusions and Future Work

In this paper, we proposed a hybrid mechanism for online classification of network traffic, in which we apply a signature-based method at the first level, and then we take advantage of a learning method to classify the remaining unknown traffic based on the network statistical features. In order to choose an appropriate classifier, we selected some of the most popular classification methods implemented by Weka, and performed some evaluations to compare the accuracy, learning time, and classification time of the selected algorithms. The experimental result showed that Adaboost and Bagging outperforming other methods in the accuracy; however, they both have a long learning time even though the data set was very small and only contained one-hour traffic. As a result, we picked J48 decision tree since it has a high accuracy while maintaining a reasonable learning time. In addition, decision trees are shown to have a reasonable height and need a few comparisons to reach a leaf which is the final label, and therefore have a very short classification time. Taking advantage of J48 decision tree as the statistical classifier, we evaluated our proposed hybrid system on a real network. Our evaluation on a large-scale ISP network, IRANET, showed promising results compared to other methods in detecting encrypted and tunneled applications.

However, one of the biggest limitations of the proposed hybrid approach is that it fails to find novel unknown applications because of applying a supervised classification method, J48. Unsupervised learning techniques have been studied recently to discover the new applications on the Internet [29]. They, however, suffer from a large number of false positives. In order to address this issue, in the immediate future we will extend our approach to automatically discover new applications. The basic idea behind this is that we assign a new class to any samples that do not have a corresponding class in the training set instead of forcing them to be classified as one of the existing labels.

Acknowledgements

The authors graciously acknowledge the funding from the Atlantic Canada Opportunity Agency (ACOA) through the Atlantic Innovation Fund (AIF) to Dr. Ghorbani.

References

[1] Fred-ezone wifi service. Available on: http://www.fred-ezone.com, September, 2008.

Table 4. Detailed information of the 3-hour traffic for the evaluation of the hybrid classifier

Time Interval	Source IPs	Destination IPs	Flows	Packets	Bytes	Applications	Unknown Flows
1	14,282	12,307	83,015	1,932,541	968,704,950	79	54362(40%)
2	15,285	12,731	82,806	2,083,392	1,027,597,142	78	53018(39%)
3	17,158	12,412	84,862	1,975,373	946,676,295	81	53699(39%)

[2] Hifn company. Available on: http://www.hifn.com, September, 2008.

[3] Iana port numbers. Available on: http://www.iana.org/assignments/port-numbers, November 2008.

[4] Iranet. Available on: http://www.iranet.ir/, October, 2008.

[5] Q1 labs network security management company. Available on: http://www.q1labs.com, September, 2008.

[6] Waikato environment for knowledge analysis (weka) version 3.5.7. Available on: http://www.cs.waikato.ac.nz/ml/weka/, June, 2008.

[7] L. Bernaille, I. Akodkenou, A. Soule, and K. Salamatian. Traffic classification on the fly. *ACM SIGCOMM Computer Communication Review*, 36(2):23–26, 2006.

[8] L. Bernaille and R. Teixeira. Early recognition of encrypted applications. In *Proceedings of the 8th Internatinoal Conference on Passive and Active Network Measurement*, pages 165–175, 2007.

[9] L. Bernaille, R. Teixeira, and K. Salamatian. Early application identification. In *Proceedings of the 2006 ACM Conference on Emerging Network Experiment and Technology (CoNEXT'06)*, 2006.

[10] L. Breiman. Bagging Predictors. *Machine Learning*, 24(2):123–140, 1996.

[11] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[12] C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[13] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

[14] M. Crotti and F. Gringoli. Traffic classification through simple statistical fingerprinting. *ACM SIGCOMM Computer Communication Review*, 37(1):5–16, 2007.

[15] H. Dahmouni, S. Vaton, and D. Rosse. A markovian signature-based approach to IP traffic classification. In *Proceedings of the 3rd annual ACM workshop on Mining network data*, pages 29–34, 2007.

[16] C. Dewes, A. Wichmann, and A. Feldmann. An analysis of internet chat systems. In *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, pages 51–64, 2003.

[17] S. Ehlert, S. Petgang, T. Magedanz, D. Sisalem, I. Links, and G. Back. Analysis and Signature of Skype VoIP Session Traffic. In *Proceedings of the Fourth IASTED International Conference on Communications, Internet, and Information Technology*, 2006.

[18] J. Erman, M. Arlitt, and A. Mahanti. Traffic classification using clustering algorithms. In *Proceedings of the 2006 SIGCOMM workshop on Mining network data*, pages 281–286, 2006.

[19] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson. Offline/realtime traffic classification using semi-supervised learning. *Performance Evaluation*, 64(9-12):1194–1213, 2007.

[20] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Thirteenth International Conference on Machine Learning (ICML'96)*, pages 148–156, 1996.

[21] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian Network Classifiers. *Machine Learning*, 29(2):131–163, 1997.

[22] K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. *ACM SIGOPS Operating Systems Review*, 37(5):314–329, 2003.

[23] P. Haffner, S. Sen, O. Spatscheck, and D. Wang. ACAS: automated construction of application signatures. In *Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, pages 197–202, 2005.

[24] G. John and P. Langley. Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345, 1995.

[25] T. Karagiannis, A. Broido, N. Brownlee, K. Claffy, and M. Faloutsos. Is P2P dying or just hiding. In *Proceedings of IEEE Global Telecommunications Conference*, volume 3, pages 1532–1538, 2004.

[26] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: multilevel traffic classification in the dark. In *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 229–240. ACM New York, NY, USA, 2005.

[27] R. Kohavi. Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, volume 7, 1996.

[28] Z. Lai, A. Galis, M. Rio, and C. Todd. Towards Automatic Traffic Classification. In *Proceedings of the third International Conference on Networking and Services (ICNS'07)*, pages 19–28, 2007.

[29] Y. Liu, W. Li, and Y. Li. Network Traffic Classification Using K-means Clustering. *Proceedings of the 2nd International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'07)*, pages 360–365, 2007.

[30] A. McGregor, M. Hall, P. Lorier, and J. Brunskill. Flow clustering using machine learning techniques. In *Proceedings of the 5th Internatinoal Conference on Passive and Active Network Measurement*, pages 205–214, 2004.

[31] A. Moore and D. Zuev. Internet traffic classification using bayesian analysis techniques. *ACM SIGMETRICS Performance Evaluation Review*, 33(1):50–60, 2005.

[32] D. Moore, K. Keys, R. Koga, E. Lagache, and K. Claffy. The CoralReef Software Suite as a Tool for System and Network Administrators. In *Proceedings of the 15th USENIX conference on System administration*, pages 133–144, 2001.

[33] B. Park, Y. Won, M. Kim, and J. Hong. Towards Automated Application Signature Generation for Traffic Identification. pages 160–167, 2008.

[34] V. Paxson. Empirically derived analytic models of wide-area TCP connections. *IEEE/ACM Transactions on Networking (TON)*, 2(4):316–336, 1994.

[35] V. Paxson and S. Floyd. Wide-area trac: The failure of Poisson modeling. *IEEE/ACM Transactions on Networking (TON)*, 3(3):226–244, 1995.

[36] J. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[37] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[38] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield. Class-of-Service Mapping for QoS: A statistical signature-based approach to IP traffic classification.

[39] S. Sen, O. Spatscheck, and D. Wang. Accurate, scalable in-network identification of p2p traffic using application signatures. In *Proceedings of the 13th international conference on World Wide Web*, pages 512–521, 2004.

[40] S. Sen and J. Wang. Analyzing peer-to-peer traffic across large networks. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 137–150, 2002.

[41] N. Williams, S. Zander, and G. Armitage. A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. *ACM SIGCOMM Computer Communication Review*, 36(5):5–16, 2006.

[42] S. Zander, T. Nguyen, and G. Armitage. Automated Traffic Classification and Application Identification Using Machine Learning. In *Proceedings of the 30th Annual IEEE Conference on Local Computer Networks (LCN'05)*, pages 250–257, 2005.