Theoretical and Applied Informatics
ISSN 1896–5334
Vol. x (200x), no. x
pp. x–x

On different ways to classify Internet traffic: a short review of selected publications

PAWEŁ FOREMSKI[a]

[a]The Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences
ul. Bałtycka 5, Gliwice, Poland
pjf@iitis.pl

Received date1. Revised date2. Accepted date3.

Abstract: Traffic classification is an important tool for network management. It reveals the source of observed network traffic and has many potential applications e.g. in Quality of Service, network security and traffic visualization. In the last decade, traffic classification evolved quickly due to the raise of peer-to-peer traffic. Nowadays, researchers still find new methods in order to withstand the rapid changes of the Internet. In this paper, we review 13 publications on traffic classification and related topics that were published during 2009-2012. We show diversity in recent algorithms and we highlight possible directions for the future research on traffic classification: relevance of multi-level classification, importance of experimental validation, and the need for common traffic datasets.
Keywords: Internet, Traffic Classification, Machine Learning

1. Introduction

Internet traffic *classification*—or *identification*—is the act of matching IP packets to the application that generated them. Traffic classification is important for managing computer networks: for example, it is used for traffic shaping, policy routing, and packet filtering. From business point of view, it provides valuable marketing information via customer profiling [1], whereas scientific and government agencies employ it to identify global Internet trends [2, 3].

Given just a single IP packet it is difficult to classify it—there is no application name in the protocol headers. In the past, the service *port number* was used for discriminating the traffic class [4], but this became ineffective in the early 2000s due to peer-to-peer (P2P) traffic [5]. Another popular and *de facto* standard classification method is Deep Packet Inspection (DPI): pattern matching on full packet contents. Despite being

accurate, it is computationally expensive and brings privacy concerns. Moreover, traffic encryption makes DPI increasingly irrelevant [6].

Instead, novel classifiers investigate groups of packets—in order to find distinguishing features of entire application protocols. Usually, a *flow* of packets is statistically summarized [7] (e.g. by average packet size and inter-packet arrival time) and the resultant *feature vector* is classified using Machine Learning (ML) [8] (e.g. Neural Network or Support Vector Machine). Such methods are largely resistant to misuse of the port number and to encryption: the overall behavior of a particular protocol or host is examined instead of seeking for a strict match in a single packet.

Latest methods tackle the problem of classification from many perspectives: counting packets [9], analyzing the DNS context [10], adopting *multi-classification* [11], and more. Our "Multilevel Traffic Classification" project (MuTriCs) [12] develops an algorithm that combines different methods to increase classification completeness and accuracy.

The aim of this work is to discuss diversity in classification methods. We also share our findings on the quality of traffic classification papers. For the review, we selected publications that: (a) present differentiated methods, (b) were published recently (2009-2012), and (c) are interesting in our opinion.

Comparing with existing surveys—namely [13], [14], and [3]—our paper focuses on different time span. We review newer works that were not mentioned in these studies: they represent novel developments in traffic classification (e.g. [9–11, 15]). Moreover, our paper gives the reader a quick insight into the methods for extracting traffic features (summarized in Table 3). We show that combining these different methods into one system can be an interesting avenue for future research on traffic classification.

We assume basic knowledge of the reader on traffic classification. For a general introduction, we refer to the works cited in the next section: particularly, [13] presents required background on traffic classification and ML.

The paper is organized as follows: in section 2., we reference related surveys and analysis papers; in section 3., we give the review; in section 4., we discuss our findings and finally we conclude in section 5.. This paper reviews 13 papers, but an accompanying web site [16] also offers an extended comparison of 21 works in a tabular form.

2. Related works

In an widely cited and comprehensive survey of traffic classification using ML [13], Nguyen et al. review works published during 2004-2007. The authors claim that ML was used for the first time for classifying traffic in 1994 [17], and that it was the starting point for much of the further work. However, many works fundamental to the state of the art appeared about a decade later, e.g. [6, 18–22].

A survey by Callado et al. [14] divides traffic analysis into *packet-* and *flow-based*, and references several traffic classification papers published during 2004-2007. Four algorithms are compared in terms of completeness and accuracy: BLINC [6], Bayesian [19], "On The Fly" [22], and Payload Analysis [23]. The authors conclude with recommendations for traffic classification and pose eight research questions.

A paper by M. Zhang et al. [3] and its accompanying website [24] present a list of 68 traffic classification papers published during 1994-2009 together with a catalog of 86 datasets used in these works. The authors propose a structured taxonomy of traffic classification and use it to answer the question on the global share of P2P traffic—basing on the results found in the reviewed papers.

Kim et al. in [8] give an insightful comparison of three general approaches to traffic classification: *ports-based*, *host-behavior-based*, and *flow-features-based*. The authors evaluate these methods on a strong, few-terabyte dataset collected at diverse geographical locations. Their five key findings were: 1) port number can still constitute a relevant feature; 2) behavior-based classification can be ineffective on backbone links and 3) it may exhibit low byte accuracy; 4) backbone traffic classification needs unidirectional TCP flow features; 5) their classifier based on Support Vector Machine (SVM) outperformed other ML algorithms and produced robust results once it was trained with a representative, unbiased training set.

In a recent study, Dainotti et al. [7] anticipate future directions in traffic classification. The authors show the evolution and current state of the field, and draw attention to the taxonomy of *flow objects* and *traffic classes*. Four challenges are discussed: 1) lack of common, representative traffic datasets labelled with ground truth; 2) inadequacy of current methods to the three trends in network protocols: encapsulation, encryption, and multi-channel communication; 3) poor scalability of algorithms to high-bandwidth links; 4) lack of standard procedures and benchmarks for method evaluation. The authors argue for further research on multi-classifier systems and for development of open-source traffic classification tools.

3. Review of selected papers

In this section, we review selected works related to traffic classification. We put our findings into four categories: 1) traffic classification, 2) detection of a particular protocol, 3) obtaining ground truth data, and 4) related.

3.1. Traffic classification

In this category, we collect papers that describe algorithms for identifying any network protocol, or at least a few protocols (e.g. group of P2P-TV protocols). For instance,

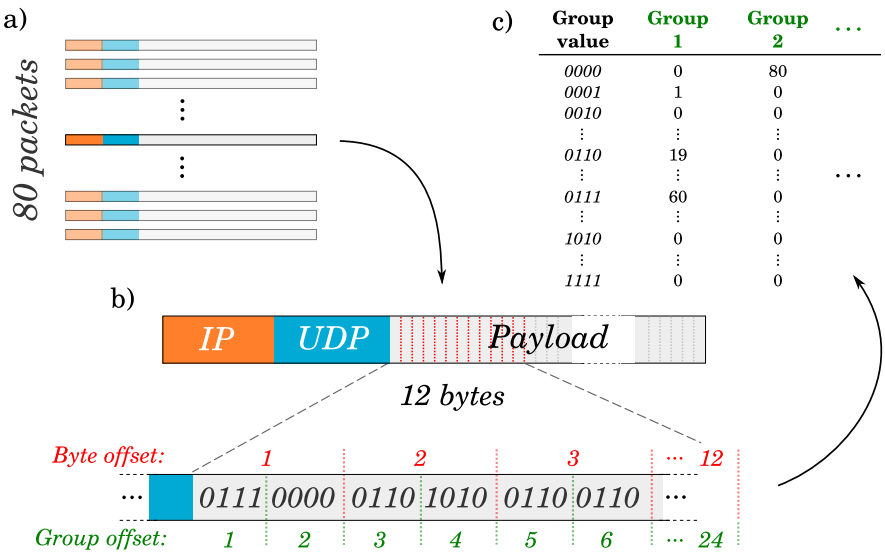

Fig. 1: Feature extraction in the KISS algorithm: for each packet in an 80-packet window (a), the first 12 bytes of UDP payload are divided into 24 groups of 4 bits each (b). Number of occurrences of distinct values in given group is counted for the whole packet window (c).

such algorithms can be deployed on a router to provide statistics on the traffic passing through it.

1) KISS: Stochastic Packet Inspection Classifier for UDP Traffic: The work by A. Finamore et al. [15] published in 2010 (extends the original 2009 paper [25]) presents a payload inspection classifier for UDP traffic. The authors exploit the fact that protocols running over UDP must implement an application-specific header at the beginning of the packet payload, due to stateless nature of UDP communication.

For each 80-packet window in a given flow, the KISS algorithm counts occurrences of distinct 4-bit *groups* in the first 12 bytes of the packet payload; see Fig. 1 for an illustration. For each of 24 groups, a χ^2-like test is used in order to measure the distance between distribution of observed values and the uniform distribution, according to Equation 1:

$$X_i = \sum_{v=0000_2}^{1111_2} \frac{(O_v^i - E)^2}{E}, \tag{1}$$

where: X_i is the distance for group offset i, v is the value, O_v^i is the number of observed occurrences for value v on offset i, and E is the expected value ($E = \frac{80}{2^4} = 5$). The symbols 0000_2 and 1111_2 represent binary numbers: 0 and 15 in decimal system, respectively.

Thus, a characterization of randomness in the application header is obtained, in form of a 24-element feature vector. This vector is used in an SVM decision process, i.e. it is used for training and classification in a typical manner.

The authors evaluated the algorithm on a ca. 100GB dataset of real and testbed network traffic, obtaining respectively 99.6% and <1% of True Positives and False Positives, on average.

2) K-Dimensional Trees for Continuous Traffic Classification: In an interesting work published in 2010 by V. Carela-Español et al. [26], the authors revisit the idea by L. Bernaille et al. [22] of early traffic classification by analyzing the size and direction of the first few packets of a TCP connection.

However, in this new work the authors apply the K-dimensional trees algorithm [27] instead, which resulted in relatively small times for training and classification. The proposed system operates in real-time and can be continuously retrained. A preliminary evaluation was performed, using a ca. 1TB dataset of 12 types of real network traffic.

3) Abacus: Accurate behavioral classification of P2P-TV traffic: In 2011, P. Bermolen et al. [9] published an exhaustive work on a classifying P2P-TV traffic, preliminarily introduced in [28].

The authors present a method that counts the number of packets received by a given host from each of its peers. Histogram of packet counts received in a 5-second window is used as a feature vector for an SVM classification algorithm.

Bermolen et al. present an excellent experimental analysis of performance, portability, and parameter sensitivity. The authors evaluated the system on a ca. 26GB dataset of testbed P2P-TV traffic (SopCast, TVAnts, PPLive, and Joost) and on a ca. 4GB of real "background" traffic: they report 95% of True Positives and less than 0.1% of False Positives in the worst case—for packets, bytes, and peers.

4) TCP Traffic Classification Using Markov Models: In a work published in 2010 by G. Münz et al. [29], a lightweight method for classification of TCP flows using observable Markov chains [30] is presented. The discretized packet length, direction, and position within the flow are mapped to a state. For each application of interest, a Markovian model is generated in the training stage. During classification, the a-posteriori probability of observed packets is calculated for each model, and the maximum value is chosen.

The authors performed experimental validation on a small dataset and compared the results to the well-established work by L. Bernaille et al. [22]; however, these two methods are inherently different. The Markov chain method yielded better stability of the results, with similar average precision and recall values. The authors extended their method in [31] by introducing a special "end of connection" Markov state, which improved the accuracy (validated on a larger dataset).

5) Early Classification of Network Traffic through Multi-classification: The work

Label	Classifier (see [33, 34])	Overall performance	Selected?
J48	J48 Decision Tree	97.2%	✓
K-NN	K-Nearest Neighbor	95.9%	✓
R-TR	Random Tree	96.3%	✓
RIP	Ripper	97.0%	✓
MLP	Multi Layer Perceptron	82.3%	✓
NBAY	Naive Bayes	43.7%	-
PL	PortLoad [35]	83.7%	✓
PORT	Port number	15.6%	-

Table 1: Stand-alone classifiers used in [11]. The "Overall performance" column presents the overall classification accuracy, as reported by the authors; the "Selected?" column indicates which classifiers were used in the final system.

Label	Combiner	Reference in [32]	Best performance
NB	Naive Bayes [36]	pp. 126	93.5%
MV	Majority Voting [37]	pp. 112	90.8%
WMV	Weighted Majority Voting [38]	pp. 123	91.0%
D-S	Dempster-Shafer [39]	pp. 175	97.0%
BKS	Behavior Knowledge Space [40]	pp. 128	97.9%
WER	Wernecke [41]	pp. 129	97.9%

Table 2: Algorithms for combining pattern classifiers, as applied in [11]. The "Best performance" column gives classification accuracy for the best selection of stand-alone classifiers working in an ensemble, as reported by the authors (see Table 1).

by A. Dainotti et al. [11] published in 2011 presents an innovative approach of multi-classification: the traffic is simultaneously processed by an ensemble of several stand-alone classifiers, and the final result is obtained using a decision combiner algorithm [32].

The authors connect eight stand-alone classifiers (see Table 1) using six state-of-the-art combiners (see Table 2). Experimental validation on a 59GB dataset of real traffic yielded the best accuracy for the BKS combiner and an ensemble of 6 classifiers: J48, K-NN, R-TR, RIP, MLP, and PL.

The authors highlight that in case we limit feature extraction to just the first few packets in a flow, their method brings significant performance improvements, comparing to the best results of stand-alone classifiers working alone: for example, in case of just the first packet being used, a 20.8% improvement. The authors chose to use the first 4 packets, obtaining the final accuracy of 98.4%; supplementary metrics were not reported.

6) CUTE: Traffic Classification Using TErms: In 2012, S.H. Yeganeh et al. published a paper [42] in which they propose a payload inspection classifier that automati-

cally finds protocol signatures.

For the training, the algorithm extracts common terms shared by flows of a given protocol: it aligns the flows and finds all common substrings of at least b bytes. Next, for each protocol, it assigns *weights* to terms, according to Equation 2:

$$W_t^p = \begin{cases} (\frac{f_p^t}{\sum_{p \in P} f_p^t})^\rho & f_t^p \geq T \\ 0 & f_t^p < T \end{cases}, \tag{2}$$

where f_t^p is the frequency of term t in protocol p, P is the set of all protocols, and W_t^p is the term weight; ρ and T are the algorithm parameters. Terms that are unique to protocol have weights close to 1, whereas common terms have weights close to 0.

During classification, for each protocol, the algorithm searches the packet payload for the learned terms, and computes the average weight. The protocol with the maximum value is chosen as the target class.

Yeganeh et al. show by means of theoretical analysis and experimental validation, that in case of pattern matching for traffic classification, occurrences of terms in network flows are more important than their relative order. In practice, this means that it is enough to use term *sets* instead of *lists*: one can identify a certain protocol by checking for occurrence of terms in any order. This makes CUTE inherently simpler and faster than similar algorithms that employ term lists, e.g. LASER [43].

The authors used two traffic traces from Tier-1 ISPs for experimental analysis, i.e. tuning the classification system and validating its accuracy. They report precision and recall metrics above 90% for almost all protocols considered in the experiment.

3.2. Single application detection

In this subsection, we put the algorithms that aim at single application or certain traffic kind. For instance, such algorithms can be deployed on a network firewall in order to block access to given service. We maintain the numbering of papers for easy referring in Table 3.

7) Tunnel Hunter: Detecting application-layer tunnels with statistical fingerprinting: In a paper published in 2009 [44], M. Dusi et al. present a reliable method for detecting HTTP and SSH tunnels.

The algorithm is trained with legitimate (non-tunneled) HTTP and SSH traffic. Each flow is characterized by a signature consisting of packet size, inter-arrival time, and arrival order. During classification, a flow "anomaly score" is computed by comparing the flow signature to fingerprints of legitimate traffic. If the value is above a certain level, the flow is considered as carrying tunneled traffic. The authors claim nearly 100% completeness and accuracy (verified experimentally).

8) Skype-Hunter: A real-time system for the detection and classification of Skype traffic: The paper by D. Adami et al. published in 2012 [45] introduces a novel method for identification of the Skype protocol.

The authors present a detailed, packet-level analysis of the Skype traffic and propose a relevant detection algorithm that combines signature-based and statistical procedures. The method is experimentally validated on several datasets—compared to standard statistical classifiers and to a state-of-the-art Skype classifier [46], it yielded better performance results.

3.3. Obtaining ground truth data

Below we describe the papers on datasets for verifying the accuracy of classification methods.

In a typical scenario, an author of a new method will work on a trace of network traffic while developing the algorithm. The traffic composing the trace needs to be representative for the scope of interest of a particular research effort. The dataset should also indicate the real application that generated each flow in the dataset, so the researcher is able to compare the results of the algorithm with the right answer: this information is called *ground truth*.

9) GT: picking up the truth from the ground for Internet traffic: In a 2009 paper published by F. Gringoli et al. [47], the authors present a distributed system for capturing Internet traffic in a computer network. The system keeps the names of applications that generated the traffic.

A special software agent “gt" is installed on each machine taking part in the experiment. The agent periodically queries the operating system for a list of opened network sockets and the names of applications that own them. For each socket, it stores a piece of information with current time-stamp, local and remote IP address and port number, transport protocol, and application name. At the same time, a standard packet sniffer is run on the gateway router, so that all the traffic coming from and into the local network is captured.

Finally, a post-processing tool “ipclass" is run. The tool connects the socket information collected by gt with the traffic captured on the router. As the result, a traffic trace file annotated with ground truth is produced. The authors validated the method on a 218GB dataset. For the completeness metric, they report more than 99% of bytes and 95% of flows.

10) Quantifying the accuracy of the ground truth associated with Internet traffic traces: In 2011 M. Dusi et al. [48] published a paper that compares their gt tool [47] to traditional port- and DPI-based ground truth establishment methods.

Basing on evaluation on a ca. 200GB dataset, the authors claim that—depending on the protocols composing a trace—ground truth information can be incorrect for up to

91% bytes for port-based and 26% for DPI-based methods. The authors speculate that the error one might commit while applying these well-established methods to publicly available anonymized traces is significant, especially for modern traffic like Streaming, Skype, or P2P.

11) Tracedump: A Novel Single Application IP Packet Sniffer: A paper by P. Foremski published in 2012 [49] introduces a packet sniffer that captures traffic of a single Linux process only. This solves the problem of ground truth accuracy, as the application name is immediately known.

The author explains implementation of a single-process packet sniffer and provides an architectural view on the proposed solution. The “tracedump" utility captures all application traffic in real-time, including DNS traffic. A short evaluation on BitTorrent traffic is presented.

The “tracedump" tool can run a computer program in a fully controlled manner—for instance, Graphical User Interface (GUI) testing tools can be applied to create a kind of specialized traffic generator (preliminary results available at [50]).

3.4. Related works

In the last subsection, we present works that analyze IP traffic and are similar to traffic classification.

12) Taking a Peek at Bandwidth Usage on Encrypted Links: In a 2011 paper [51], M. Dusi et al. present a simple regression-tree-based algorithm that monitors the amount of data that protocols transmit over encrypted tunnels (incl. IPSec).

During the training phase, both the cipher- and plain-text transmissions are visible to the algorithm; the plain-text is used for ground truth information. As traffic features, the authors employed probability mass function of packet sizes, and statistics related to changes in packet direction. During the operation phase, the algorithm extracts flow features each few seconds, and applies a regression tree algorithm in order to give estimates on the traffic carried within the tunnel.

The authors evaluated their method on a ca. 50GB dataset and reported an acceptable accuracy: the performance depends on the differences in the networks used for training and testing.

13) DNS to the Rescue: Discerning Content and Services in a Tangled Web: In 2012, I. Bermudez et al. published a paper on inferring Internet traffic by analyzing its DNS context [10]. The work introduces “DN-Hunter", a system that tags traffic flows with their associated domain name, based on the fact that each new flow is anticipated by a DNS query.

The system consists of two modules: a flow sniffer, which reconstructs traffic flows, and a DNS resolver, which maintains mapping between clients, domains, and servers. The authors verified that flow tagging can be accomplished in most cases and could not

be replaced by making a reverse DNS lookup or inspecting TLS certificates—this would fail in 94% or 86%, respectively. The key property of this novel method is that it can identify traffic before the actual flow starts.

Using capabilities of DN-Hunter, the authors provide a detailed analysis of Content Delivery Networks (CDNs) in 5 datasets of total 64 million flows, covering thousands of ISP customers in US and Europe. Analysis of real traffic revealed domains handled by hundreds of servers that change with time. The authors discovered a diurnal pattern of more machines during late evenings; a similar phenomenon was noticed for CDNs and their domains. For an 18-day observation period about 100,000 new domains emerged each day, which reflects the rapid growth of the Internet.

DN-Hunter can map distribution of particular content across CDNs—the authors found that LinkedIn was hosted by Edgecast (59% of flows), Akamai (17%), CDNetworks (3%), and on own servers (22%). The system can also reveal the domains of a specific CDN: top three domains provided by Amazon EC2 in Europe were cloudfront.net (20%), playfish.com (16%), and sharethis.com (5%). Finally, DN-Hunter can tell the most popular services delivered on a given IP port number—for port 25 the authors observed service tags of “smtp", “mail", “mxN”, and several others. Interestingly, they also identified several BitTorrent trackers running on the Google Appspot service.

4. Discussion

1. **There are many ways to classify the traffic.** Each work reviewed in sections 3.1. and 3.2. presents a different approach to classification: analysis of packet count, length, payload, etc.—see Table 3 for a summary. We speculate that each modern Internet protocol exhibits so many phenomena that it has plenty of observable traffic characteristics that can reveal its generating application. Moreover, A. Dainotti et al. in [11] (sect. 3.1.) proved that it is possible to combine multiple different classifiers into one system that unveils high performance.

   We argue that:
   (a) there are many traffic features yet to be found (anticipated e.g. by [9,10,15]);
   (b) traffic classification algorithms can be combined so they complement each other (e.g. [15] for UDP and [31] for TCP traffic);
   (c) there is much room for improvement in the design of traffic classifiers that analyze several kinds of traffic features at the same time, i.e. multi-level traffic classifiers (e.g. [6, 11]).
2. **Classification methods need thorough validation.** New services appear rapidly on the Internet, and the application protocols get more sophisticated [7], hence modeling of new kinds of traffic gets harder. For instance, at the time of this writing, there is no adequate traffic model for the SPDY [54] protocol, introduced by

Google and deployed for its popular “Gmail" service. Consequently, robust traffic classification methods need thorough *experimental* validation, as purely theoretical approach is insufficient. A certain sign of a high-quality paper is a detailed section on validation, employing an up-to-date traffic trace.

We give our recommendations for validating classification methods:

(a) usage of large, representative, and geographically diverse datasets with relevant amounts of background traffic (e.g. [8, 15]);

(b) presentation of the results in terms of well-established and *complementary* performance metrics—e.g. the recall metric together with precision, or True Positives together with False Positives (e.g. [15, 31]);

(c) analysis of parameter sensitivity of the algorithm (e.g. [9, 42]).

3. **The problem of common traffic datasets is still unsolved.** Several respected scientists demanded publication of common, packet-level traffic datasets labeled with ground truth: e.g. [55] in 2007 and [7] more recently. This would enable systematic and fair comparison of classification methods, but the problem still remains largely unsolved. Some authors published their datasets, but none of them satisfies all of the postulated requirements[1]. Others, like CAIDA [57] or MAWI [58] publish datasets without ground truth and packet payload, which limits their usability.

   However, authors of the studies referenced in section 3.3. made ground truth data collection simpler and more comprehensible. Particularly, the “gt" [47] software agent seems to be a candidate for the standard ground truth tool for current and future research on Internet traffic.

5. Conclusions

In this paper, we reviewed 13 significant papers on traffic classification and related matters, published during 2009-2012. We presented the review in 4 categories: general traffic classification (sect. 3.1.), single protocol detection (sect. 3.2.), the ground truth problem (sect. 3.3.), and related works (sect. 3.4.). We showed diversity in methods for characterizing modern IP traffic and discussed a few important issues, giving our recommendations. We also presented a succinct “review of reviews" in traffic classification in section 2..

It is almost a decade since first major publications on traffic classification appeared [13], but the authors of the reviewed papers proved that it is still possible to find new algorithms [10, 15], or significantly improve the existing ones [26]. In order to classify an IP flow, one can choose to either focus on a specific traffic feature (packet counts

[1] A short list of published datasets can be found at [56]

[9], lengths [26], payload characteristics [15, 42], etc.), use many features at once (e.g. [31,45]), or combine several approaches in a multi-classifier system ([11] in sect. 3.1.). Especially for the latter technique we speculate a vast space for improvement.

Classification methods need to be verified on real IP traffic. The problem of obtaining adequate traces labeled with ground truth (introduced in sect. 3.3.) is still largely unsolved. This limits systematic and fair comparison of existing methods: there are no “reference benchmarks" in traffic classification. Besides, the authors of [48] suggest that there may be a significant error in self-made traffic traces anyway. Two utilities—“gt" [47] and “tracedump" [49]—can be applied to assure the accuracy of ground truth data.

Let us conclude with an observation that we are able to tell things apart if we can see the differences among them, i.e. the more one can see, the more has he the power to discriminate. Our paper showed diversity in methods for classifying IP traffic—in our opinion, an interesting direction for future research on traffic classification.

Acknowledgments

This work was funded by the Polish National Science Centre, under research grant nr 2011/01/N/ST6/07202—project “Multilevel Traffic Classification” [12].

The author would like to thank Michele Pagano and Christian Callegari of University of Pisa for their continuous help and support, and Michał Romaszewski of IITiS PAN Gliwice for proofreading early manuscripts of this paper.

References

[1] M. Pietrzyk, L. Plissonneau, G. Urvoy-Keller, and T. En-Najjary, “On profiling residential customers,” *Traffic Monitoring and Analysis*, pp. 1 – 14, 2011.

[2] “CAIDA: The Cooperative Association for Internet Data Analysis.” Available from: http://www.caida.org/ [27 March 2013].

[3] M. Zhang, W. John, K. Claffy, and N. Brownlee, “State of the art in traffic classification: A research review,” in *PAM Student Workshop*, 2009.

[4] “IANA Service Name and Transport Protocol Port Number Registry.” Available from: http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml [27 March 2013].

[5] T. Karagiannis, A. Broido, N. Brownlee, K. C. Claffy, and M. Faloutsos, “Is p2p dying or just hiding?,” in *Global Telecommunications Conference, 2004. GLOBECOM’04. IEEE*, vol. 3, pp. 1532 – 1538, IEEE, 2004.

[6] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel traffic classification in the dark," in *ACM SIGCOMM Computer Communication Review*, vol. 35, pp. 229 – 240, ACM, 2005.

[7] A. Dainotti, A. Pescapè, and K. C. Claffy, "Issues and future directions in traffic classification," *Network, IEEE*, vol. 26, no. 1, pp. 35 – 40, 2012.

[8] H. Kim, K. C. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, "Internet traffic classification demystified: Myths, caveats, and the best practices," in *Proceedings of the 2008 ACM CoNEXT conference*, p. 11, ACM, 2008.

[9] P. Bermolen, M. Mellia, M. Meo, D. Rossi, and S. Valenti, "Abacus: Accurate behavioral classification of P2P-TV traffic," *Computer Networks*, vol. 55, no. 6, pp. 1394 – 1411, 2011.

[10] I. Bermudez, M. Mellia, M. M. Munafò, R. Keralapura, and A. Nucci, "DNS to the Rescue: Discerning Content and Services in a Tangled Web," in *Proceedings of the 12th ACM SIGCOMM Conference on Internet Measurement*, vol. 1101, p. 12, 2012.

[11] A. Dainotti, A. Pescapé, and C. Sansone, "Early classification of network traffic through multi-classification," *Traffic Monitoring and Analysis*, pp. 122 – 135, 2011.

[12] "MuTriCs: Multilevel Traffic Classification." Available from: http://mutrics.iitis.pl/ [27 March 2013].

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

[14] A. Callado, C. Kamienski, G. Szabó, B. Gero, J. Kelner, S. Fernandes, and D. Sadok, "A survey on internet traffic identification," *Communications Surveys & Tutorials, IEEE*, vol. 11, no. 3, pp. 37 – 52, 2009.

[15] A. Finamore, M. Mellia, M. Meo, and D. Rossi, "KISS: Stochastic packet inspection classifier for udp traffic," *Networking, IEEE/ACM Transactions on*, vol. 18, no. 5, pp. 1505 – 1515, 2010.

[16] "MuTriCs: Literature review." Available from: http://mutrics.iitis.pl/literature-review [27 March 2013].

[17] J. Frank, "Artificial intelligence and intrusion detection: Current and future directions," in *Proceedings of the 17th National Computer Security Conference*, October 1994.

[18] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, "Class-of-service mapping for QoS: A statistical signature-based approach to IP traffic classification," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pp. 135 – 148, ACM, 2004.

[19] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, pp. 50 – 60, ACM, 2005.

[20] S. Zander, T. Nguyen, and G. Armitage, "Automated traffic classification and application identification using machine learning," in *Local Computer Networks, 2005. 30Th Anniversary. The IEEE Conference on*, pp. 250 – 257, IEEE, 2005.

[21] K. Xu, Z.-L. Zhang, and S. Bhattacharyya, "Profiling internet backbone traffic: Behavior models and applications," in *ACM SIGCOMM Computer Communication Review*, vol. 35, pp. 169 – 180, ACM, 2005.

[22] L. Bernaille, R. Teixeira, and K. Salamatian, "Early application identification," in *Proceedings of the 2006 ACM CoNEXT conference*, p. 6, ACM, 2006.

[23] G. Szabó, I. Szabó, and D. Orincsay, "Accurate traffic classification," in *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, pp. 1–8, IEEE, 2007.

[24] "Internet Traffic Classification." Available from: http://www.caida.org/research/traffic-analysis/classification-overview/ [27 March 2013].

[25] A. Finamore, M. Mellia, M. Meo, and D. Rossi, "KISS: Stochastic packet inspection," *Traffic Monitoring and Analysis*, pp. 117 – 125, 2009.

[26] V. Carela-Español, P. Barlet-Ros, M. Solé-Simó, A. Dainotti, W. de Donato, and A. Pescapé, "K-dimensional trees for continuous traffic classification," *Traffic Monitoring and Analysis*, pp. 141 – 154, 2010.

[27] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software (TOMS)*, vol. 3, no. 3, pp. 209–226, 1977.

[28] S. Valenti, D. Rossi, M. Meo, M. Mellia, and P. Bermolen, "Accurate, fine-grained classification of P2P-TV applications by simply counting packets," *Traffic Monitoring and Analysis*, pp. 84 – 92, 2009.

[29] G. Münz, H. Dai, L. Braun, and G. Carle, "TCP traffic classification using Markov models," *Traffic Monitoring and Analysis*, pp. 127 – 140, 2010.

[30] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257 – 286, 1989.

[31] G. Münz, S. Heckmüller, L. Braun, and G. Carle, "Improving Markov-based TCP Traffic Classification," in *KiVS* (N. Luttenberger and H. Peters, eds.), vol. 17 of *OASICS*, pp. 61–72, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2011.

[32] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, 2004.

[33] N. Williams, S. Zander, and G. Armitage, "Evaluating machine learning algorithms for automated network application identification," *Center for Advanced Internet Architectures, CAIA, Technical Report B*, vol. 60410, p. 2006, 2006.

[34] R. Alshammari and A. N. Zincir-Heywood, "Machine learning based encrypted traffic classification: identifying ssh and skype," in *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*, pp. 1–8, IEEE, 2009.

[35] G. Aceto, A. Dainotti, W. de Donato, and A. Pescapé, "PortLoad: Taking the best of two worlds in traffic classification," in *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*, pp. 1 – 5, IEEE, 2010.

[36] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine learning*, vol. 29, no. 2, pp. 103–130, 1997.

[37] R. Battiti and A. M. Colla, "Democracy in neural nets: Voting schemes for classification," *Neural Networks*, vol. 7, no. 4, pp. 691–707, 1994.

[38] L. Shapley and B. Grofman, "Optimizing group judgmental accuracy in the presence of interdependencies," *Public Choice*, vol. 43, no. 3, pp. 329–343, 1984.

[39] G. Rogova, "Combining the results of several neural network classifiers," *Neural networks*, vol. 7, no. 5, pp. 777–781, 1994.

[40] Y. S. Huang and C. Y. Suen, "A method of combining multiple experts for the recognition of unconstrained handwritten numerals," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 1, pp. 90–94, 1995.

[41] K.-D. Wernecke, "A coupling procedure for the discrimination of mixed data," *Biometrics*, pp. 497–506, 1992.

[42] S. H. Yeganeh, M. Eftekhar, Y. Ganjali, R. Keralapura, and A. Nucci, "CUTE: Traffic Classification Using TErms," in *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, pp. 1–9, IEEE, 2012.

[43] B.-C. Park, Y. J. Won, M.-S. Kim, and J. W. Hong, "Towards automated application signature generation for traffic identification," in *Network Operations and Management Symposium, 2008. NOMS 2008. IEEE*, pp. 160–167, IEEE, 2008.

[44] M. Dusi, M. Crotti, F. Gringoli, and L. Salgarelli, "Tunnel hunter: Detecting application-layer tunnels with statistical fingerprinting," *Computer Networks*, vol. 53, no. 1, pp. 81 – 97, 2009.

[45] D. Adami, C. Callegari, S. Giordano, M. Pagano, and T. Pepe, "Skype-Hunter: A real-time system for the detection and classification of Skype traffic," *International Journal of Communication Systems*, vol. 25, no. 3, pp. 386 – 403, 2012.

[46] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing skype traffic: When randomness plays with you," in *ACM SIGCOMM Computer Communication Review*, vol. 37, pp. 37 – 48, ACM, 2007.

[47] F. Gringoli, L. Salgarelli, M. Dusi, N. Cascarano, F. Risso, and K. Claffy, "Gt: Picking up the truth from the ground for internet traffic," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 5, pp. 13 – 18, 2009.

[48] M. Dusi, F. Gringoli, and L. Salgarelli, "Quantifying the accuracy of the ground truth associated with Internet traffic traces," *Computer Networks*, vol. 55, no. 5, pp. 1158 – 1167, 2011.

[49] P. Foremski, "Tracedump: A Novel Single Application IP Packet Sniffer," *Theoretical and Applied Informatics*, vol. 24, no. 1, pp. 23 – 31, 2012.

[50] "MuTriCs: Automatic trace generation." Available from: http://mutrics.iitis.pl/automatic-traffic-trace-generation [27 March 2013].

[51] M. Dusi, A. Este, F. Gringoli, and L. Salgarelli, "Taking a Peek at Bandwidth Usage on Encrypted Links," in *Communications (ICC), 2011 IEEE International Conference on*, pp. 1 – 6, IEEE, 2011.

[52] "CoMo-UPC: TMA evaluation service @ UPC." Available from: http://monitoring.ccaba.upc.edu/como-upc/ [27 March 2013].

[53] "Tstat - Skype Traces." Available from: http://tstat.tlc.polito.it/traces-skype.shtml [27 March 2013].

[54] "SPDY: An experimental protocol for a faster web." Available from: http://www.chromium.org/spdy/spdy-whitepaper [27 March 2013].

[55] L. Salgarelli, F. Gringoli, and T. Karagiannis, "Comparing traffic classifiers," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 3, pp. 65 – 68, 2007.

[56] "MuTriCs: Datasets review." Available from: http://mutrics.iitis.pl/traffic-traces [27 March 2013].

[57] "CAIDA Data." Available from: http://www.caida.org/data/overview/ [27 March 2013].

[58] "MAWI WG Traffic Archive." Available from: http://mawi.wide.ad.jp/mawi/ [27 March 2013].

O wielu sposobach klasyfikacji ruchu internetowego: krótki przegląd wybranych publikacji

Streszczenie

Artykuł prezentuje przegląd 13 wybranych prac z dziedziny klasyfikacji ruchu internetowego pod kątem różnorodności w zastosowanych metodach. Prace zostały wybrane z najciekawszych naszym zdaniem publikacji z ostatnich kilku lat (2009-2012). W porównaniu do istniejących przeglądów literaturowych—np. [13], [14], czy [3]—niniejszy artykuł dotyczy nowszych badań, oraz wykazuje, że łączenie wielu metod klasyfikacji w jeden system może być ciekawym kierunkiem dla przyszłych badań w tej dziedzinie.

Klasyfikacja ruchu internetowego polega na odgadnięciu nazwy protokołu komunikacyjnego lub aplikacji, która wygenerowała dany ciąg pakietów IP. Informacja ta jest przydatna np. w zarządzaniu ruchem w sieciach internetowych, gdy potrzeba kształtować ruch w zależności od jego rodzaju. Klasyfikacja ruchu znajduje zastosowanie także w zagadnieniach sieciowych związanych z wdrażaniem zasad bezpieczeństwa (np. zakaz stosowania aplikacji Skype), monitorowaniem natężenia ruchu (np. wykrywanie ataków DoS), oraz wielu innych.

Przegląd literatury został podzielony na 4 kategorie: klasyfikacja ruchu (rozdział 3.1., prace nr 1-6), detekcja pojedynczych aplikacji (rozdział 3.2., prace nr 7-8), metody pozyskiwania "wiedzy bazowej" (ang. *ground truth*, rozdział 3.3., prace nr 9-11), oraz inne (rozdział 3.4., prace nr 12 i 13). Wszystkie prace zostały podsumowane w Tabeli 3.

W ostatnim rozdziale (str. 10) prezentujemy wyniki przeglądu. Pokazujemy na przykład, że istnieje wiele metod klasyfikacji, które mogą być połączone w jeden system i wzajemnie się uzupełniać—przez *multiklasyfikację* (ang. *multi-classification*) lub obsługę różnych części ruchu (np. [31] dla TCP i [15] dla UDP). Podajemy także nasze rekomendacje dotyczące walidacji metod klasyfikacji i zbierania śladów ruchu internetowego.

Paper	Traffic features	Experimental dataset
1) Finamore et al. [15]	For 80-packet windows: amount of randomness in the first 12 bytes of payload	100GB of real and testbed traffic (P2P-TV, Skype)
2) Carela-Español et al. [26]	Size of the first few packets; port numbers	<1TB of real traffic from CoMo-UPC [52]; ground truth set with DPI
3) Bermolen et al. [9]	Histogram of packet counts received from each peer, in a time window (5s)	26GB of testbed traffic from 30 peers; <4GB of real traffic without P2P-TV
4) Münz et al. [29]	For the first few TCP packets: payload size, packet direction, position in stream	Self-made traces: 300 connections for training, 500 for testing
5) Dainotti et al. [11]	Various	Self-made 59GB trace of real traffic (Oct 2009); ground truth set with DPI
6) Yeganeh et al. [42]	Existence of precomputed terms in packet payload	Two 30-minute traces from tier-1 ISPs on different continents; no encrypted flows
7) Dusi et al. [44]	Packet size and logarithm of inter-arrival time (quantized values)	Self-made HTTP and SSH traffic (legitimate and tunneled)
8) Adami et al. [45]	Packet size, packet payload (signatures), inter-arrival times	Self-made dataset, Tstat Skype traces [53], and DARPA dataset
12) Dusi et al. [51]	For time-windows: histogram of packet sizes; vector of packet counts and sizes until change in transmission direction occurs	Self-made, real traffic: 36GB captured with “gt" [47] (Oct 2009), 10GB with ground truth set using DPI (Jul 2010)
13) Bermudez et al. [10]	DNS response received within a time-window preceding the IP flow	5 diverse sets of real traffic from EU and US; 64 million TCP flows, almost 2 days of traffic

Table 3. Summary of the reviewed papers: traffic features and datasets used for experimental validation