INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt 2014; **24**: 101–120
Published online 7 January 2014 in Wiley Online Library (wileyonlinelibrary.com) DOI: 10.1002/nem.1855

A multilevel taxonomy and requirements for an optimal traffic-classification model

Jawad Khalife,[1,*†] Amjad Hajjar[1] and Jesus Diaz-Verdejo[2]

[1] *IT department, Lebanese University, Beirut, Lebanon*
[2] *Department of Signal Processing, Telematics and Communication—CITIC-UGR, University of Granada, Granada, Spain*

SUMMARY

Identifying Internet traffic applications is essential for network security and management. The steady emergence of new Internet applications, together with the use of encryption and obfuscation techniques, ensures that traffic classification remains a hot research topic. Much research has been devoted to this topic by the research community in the last decade. However, an optimal traffic classification model has yet to be defined. Many techniques and formats have been described, with the current literature therefore lacking appropriate benchmarks expressed in a consistent terminology. Moreover, existing surveys are outdated and do not include many recent advances in the field. In this article, we present a systematic multilevel taxonomy that covers a broad range of existing and recently proposed methods, together with examples of vendor classification techniques. Our taxonomy assists in defining a consistent terminology. It could be useful in future benchmarking contexts by characterizing and comparing methods at three different levels. From this perspective, we describe key features and provide design hints for future classification models, while emphasizing the main requirements for promoting future research efforts. To motivate researchers and other interested parties, we collect and share data captured from real traffic, using two models to protect data privacy. Copyright © 2014 John Wiley & Sons, Ltd.

Received 19 April 2013; Revised 18 September 2013; Accepted 21 November 2013

1. INTRODUCTION

To deliver users' application data properly, the underlying computer network infrastructure requires administrative attention. Tackling the intensive use of network resources by each application is a challenge for both Internet providers and corporate networks. In addition to consuming bandwidth, unidentified traffic is becoming a growing source of security threats. The ability to identify different application types is therefore central to network management and security-hardening strategies. Traffic management and network security devices, such as routers, traffic shapers, firewalls, secure web gateways (SWG) and intrusion prevention systems (IPS), rely on traffic classification to enforce network access policies.

With the proliferation of various applications, classification techniques have to keep pace with many developments. General obfuscation techniques [1], encryption [2] and tunneling [3] are all used nowadays to disguise network control devices.

From this standpoint, classifying applications that are intrinsically hard to detect is a key indicator of the classifier's capabilities. For example, peer-to-peer [4] (P2P) applications are difficult to detect because of the decentralization and dynamic nature of their operation.

*Correspondence to: Jawad Khalife, IT department, Lebanese University, Beirut, Lebanon.
E-mail: jawad_khalife@hotmail.com

In the last decade, many research groups have focused on traffic classification, resulting in a huge number of publications. In most of these works, classifiers were assessed for few applications and under particular conditions. In most existing comparison studies [5–8], authors assess few methods and provide rarely deterministic results. It appears that there are now clear research trends in seeking an optimal traffic classification model.

However, existing surveys [5,9,10] are outdated because the number of publications has doubled in the last three years, with many promising approaches being proposed. Some surveys have focused on one specific research trend, such as machine learning (ML) [5].

In most of these surveys, just one criterion of interest [9,10], usually the technique or the input type, is taken into consideration in categorizing the various published methods. Classification methods are therefore mapped to non-disjoint categories. Moreover, methods often use different techniques for different input types and output formats. Comparing methods without considering these differences inevitably produces misleading results. A comprehensive survey should systematically categorize and characterize the different aspects of each method. Clearly, a systematic taxonomy of existing studies is relevant from a benchmarking perspective.

In this work, we survey long-standing and more recent achievements in traffic classification, together with examples of vendor classification techniques. We aim to provide a systematic way of categorizing and characterizing traffic classification methods. Specifically, we present a comprehensive three-level hierarchical and systematic taxonomy. This multilevel taxonomy assists in defining a single consistent terminology that should be useful in future benchmarking contexts. We characterize each method at three different levels: the input, the technique and the output. According to our taxonomy, methods are grouped into disjoint category groups. Based on survey publications, we describe their main features and provide design hints for an optimal traffic classification model. Finally, we discuss future trends in the field while outlining basic requirements that should promote research progress. To motivate researchers and other interested parties, we collect and share data captured from real traffic, using two models to protect data privacy.

The remainder of this paper is organized as follows. In Section 2, we detail the proposed multilevel taxonomy. Comparisons and general research requirements are detailed in Section 3, at the end of which we describe vendor technologies and future classification models. Conclusions are presented in Section 4.

2. A MULTILATERAL TAXONOMY OF TRAFFIC CLASSIFICATION METHODS

Traffic classification [10] involves attributing traffic objects (e.g. flows) to the traffic classes (e.g. applications) that generate them. Identification uses similar terminology to traffic classification. Identification is usually used when targeting granular application classes (as detailed in Section 2.1.1). For example, a flow might be classified as a download, but identified as a BitTorrent download. Throughout this paper, traffic classification will refer to both terms for simplicity.

Variants of the classification problem exist. Most traffic classification problems fall into the multi-classification category.

As mentioned above, hundreds of papers devoted to developing traffic classification techniques have been published. However, some existing surveys have taken a narrow view [5] by focusing on one category of techniques, while most surveys [5,9,10] are now outdated. Moreover, previous taxonomies have focused mainly on the technique used, such as port-based, statistical analysis or ML, as the basic categorization criterion. Many relevant features at the input and output levels, which might have been worthy of consideration, were omitted.

On the other hand, rigorous benchmarks usually compare methods within the same group before comparing methods across different groups. From this perspective, existing taxonomies have the drawback that the same method may fall into more than one category. In this case, comparing different categories, as do existing taxonomies, might be of little significance.

To overcome these limitations, we present a comprehensive multilevel taxonomy that covers most long-standing and recent traffic classification approaches. We characterize each classification method at three different levels, namely the classification input, the classification technique and the

DOI: 10.1002/nem

classification output. The input covers traffic characteristics that can be measured and analyzed at various levels, such as packet, flow or host. The technique describes the core of the classification process, which may involve a variety of approaches, such as payload inspection, statistical or ML. The output can be described in terms of traffic objects, such as packet, flow or host, and classes, such as file transfer, email or Skype chat. This three-level grouping provides richer information about each method while generating disjoint three-tuple categories.

Figure 1 illustrates the proposed multilevel taxonomy by showing category groups at each of the three defined levels. In this taxonomy, a classification method is necessarily a member of at least three groups: one at each level. Classification methods may rely on multiple choices or new uncategorized items at each level, referred to as hybrid and miscellaneous groups in Figure 1. A classification method is described as belonging to a three-tuple defined category that is associated with one distinctive path in Figure 1.

For example, some classification methods [11,12] belong to (Input: *payload* → Technique: *payload inspection* → Output: *flow* → *application Type*). However, it should be noted that choices at the three levels are not fully independent, because the classification technique may imply the form of the required input or output.

With such a multilevel taxonomy, comparisons should achieve higher significance. For example, it would be much more significant to compare methods having the same output group (e.g. comparing host-based approaches in Karagiannis *et al*. [13] to those in Allan *et al*. [14]), rather than comparing methods from different output groups (e.g. comparing flow-based [11] to host-based [14]).

Next, we detail existing methods from the perspective of this taxonomy. However, instead of reviewing hundreds of individual methods, we choose more recent and selective methods from each category for reasons of space.

2.1. *Categorization by the classification's output*

In the following, we present different classification methods according to the classification output. Mainly, the classifier's output consists of associating traffic objects with traffic classes.

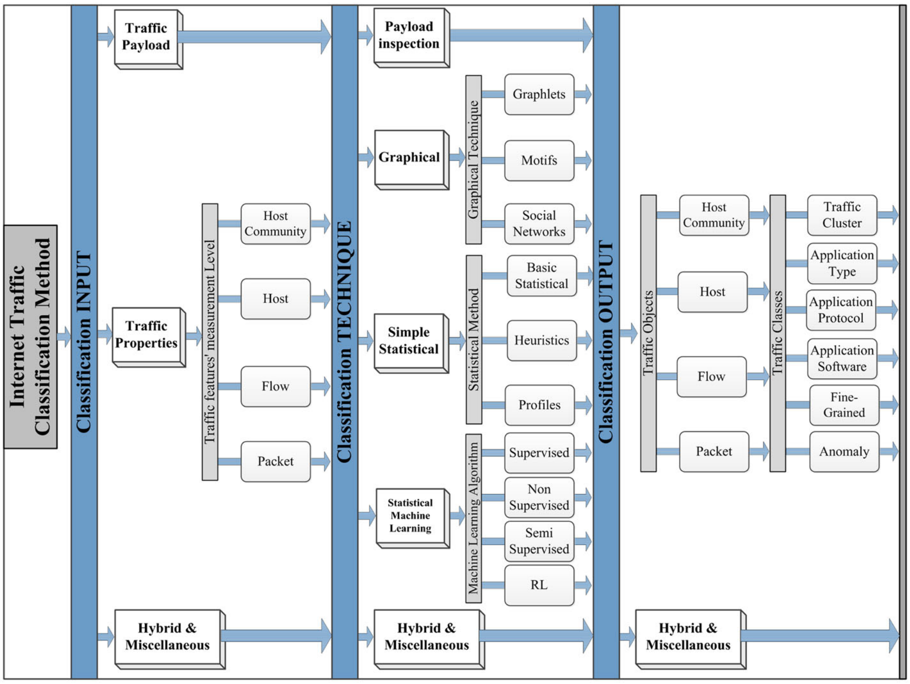

Figure 1. A multilevel taxonomy of traffic classification methods characterized on three levels

DOI: 10.1002/nem

2.1.1. Traffic classes

We categorize traffic classes [15] (see Figure 1) according to the degree of classification detail. They range from fine-grained (e.g. a web application function) to coarse-grained (e.g. traffic cluster) levels. Specifically, traffic classes include: (i) traffic cluster [16] such as bulk or small transactions; (ii) application type [17] such as game, browsing or chat; (iii) application protocol [18] such as hypertext transfer protocol (HTTP), HTTP secure (HTTPS), file transfer protocol (FTP), domain name system (DNS), simple mail transfer protocol (SMTP), post office protocol (POP3), secure sockets layer (SSL) or secure shell (SSH); (iv) application software such as a specific FTP or BitTorrent client software; (v) fine-grained traffic classification [19] such as Facebook chat, Google search or Skype voice call; and anomaly [20] class.

Anomaly is one of the main classification targets of IPS security systems. Anomaly alarms are generated whenever a deviation (e.g. SYN-cookie threshold for FTP servers) from a pre-estimated 'normal' behavior (discussed in Section 2.3.2) is detected. Traffic anomalies include network, transport and application layer anomalies.

Fine-grained traffic classification is crucial for multi-channel applications that open multiple connections for different purposes. For instance, Skype [22] offers several services (user authentication, voice and video communications, file transfer, chat, etc.) over transmission control protocol (TCP) and user datagram protocol (UDP) connections. Although belonging to Skype at a coarse-grained level, these flows should be identified differently at a finer-grained classification level. Fine-grained traffic classification is crucial for mobile applications [21], most of which rely on HTTP and HTTPS flows.

Dynamically changing application type should be also considered in traffic classifiers. A typical example is Skype [22] voice calls, during which the user might issue a file transfer, an instant messaging conversation or switch to video mode. Such a dynamic variation in the flow application type should be detected by the classifier at a fine-grained level during the whole lifetime of a flow.

The scope of detected traffic classes is an important factor in evaluating traffic classification methods. In each of the surveyed works, authors cover few traffic classes as implicitly found in the training datasets and tested network environments. These included three common groups: (i) standard application protocols such as HTTP, FTP, DNS, SMTP and POP3; (ii) encrypted applications such as SSL and SSH; and (iii) the P2P application type such as BitTorrent, Gnutella, eDonkey and Skype.

2.1.2. Traffic objects

Given a traffic class, different objects of interest may be associated as ranging from fine-grained (packets, flows) to coarse-grained levels (hosts, host communities). Traffic objects (see Figure 1) include: (i) packet [11]; (ii) flow [23,24]; (iii) host [13,14,18,25–28]; and (iv) host community [29–31]. Bytes [32] are usually adopted for statistical evaluation (e.g. monitoring volume consumption). They are not literally defined, nor can be targeted as, traffic classification objects.

Specifically, flow-based classification defines a flow as a unidirectional or bidirectional series of IP packets having the same IP addresses, port numbers and transport layer protocol. A bag of flows [33] (BoF) includes the set of correlated flows generated by the same application as one classification object. At a higher level, a host (represented by an IP address) or a host community can be targeted for classification as one traffic object. A host community includes a set of hosts involved in the same application.

As per most of the surveyed works, flows are the most adopted traffic classification objects. Ideally, a classifier should be able to identify traffic at the most possible granular level, which is supposed to be computationally intensive. Coarse-grained classification objects are potentially more robust against network fluctuations usually affecting low-level statistics. Host and host community-based classification models, except for a few modelling studies [27], associate a single application to each host or host community such as DNS host, file transferring host, P2P host community or SMTP host community. However, a host might be running more than one application simultaneously. As per the normal behavior, a user might, for instance, be surfing the web and making a Voice over IP (VoIP) phone call while running a P2P program in the background. Deciding upon the preference of classification objects still depends on the classifier's intended use. For example, high packet accuracy is recommended for critical VoIP signaling packets, controlling a whole voice session.

2.2. *Categorization by the classification's INPUT*

In the following, we present a survey of different traffic classification methods according to the input type. The classification input may include the traffic payload, general traffic attributes, hybrid sets of both and miscellaneous types.

2.2.1. *Traffic payload*

The traffic payload is used by many classification techniques (detailed in Section 2.3.1) that disclose the packet payload, beyond the layer 4 header. Though many studies [12,34] proved that few packets have to be disclosed, payload-based classification is still less preferred because of privacy concerns. Alternatively, classifiers can analyze general traffic attributes, which are most adopted in the literature.

2.2.2. *Traffic attributes*

Relying on non-payload input is referred to as blind or in-the-dark classification. Non-payload traffic attributes are usually based on information at the network and transport layers. They can be gathered at different levels such as packet, flow, host and host community. They are similar to, but not necessarily the same as, the level of the traffic classification objects (shown in Section 2.1.2). Specifically, traffic attributes include: (i) packet-level attributes such as packet headers, sizes and inter-arrival times; (ii) flow-level attributes such as the flow size and duration; (iii) host-level attributes such as the number of connections and opened ports; and (iv) host community-level attributes including graph metrics such as connection degree and graph diameter.

Throughout the surveyed works, packet inter-arrivals and sizes, together with their derivations (e.g. average values), were mostly adopted.

A common and relevant practice in the literature consists of reducing the size of the classification input to enhance the performance. Input reduction can be achieved through sampling [12], feature selection [5] and early classification methods [35–37] focusing on the first few packets in a flow. Deciding upon which input type to use should take into consideration many factors, such as input size, user privacy, network dynamics and monitoring feasibility. For instance, packet inter-arrival time can be highly affected by jitter. Bidirectional traffic attributes are not applicable for multi-homed networks using a single monitoring location.

2.3. *Categorization by the classification technique*

According to the classification technique, existing methods can be categorized as: (i) payload inspection; (ii) simple statistical; (iii) statistical ML; (iv) and graphical.

In the early phase of traffic classification, the port-based technique [38] used to be typically the fastest and simplest one. To classify flows and packets, it simply relies on the Internet Assigned Numbers Authority's (IANA's) registered list (e.g. HTTP uses TCP port 80). However, the use of port obfuscation, address translation, port forwarding and protocol tunneling, together with the use of unregistered ports and multichannel applications, have deprecated the use of this technique. For instance, a Skype session using TCP port 80 will be identified as HTTP or web browsing using port-based classification, although it might be carrying different types of traffic such as voice call, chat or file transfer. Similarly, a P2P client might obfuscate its default port number and bind to TCP port 80 to generate traffic that would appear as web browsing. Moreover, different types of traffic can be encrypted and sent over SSL tunnels, using TCP port 443. Apparently, network security controls based on port classification can be easily bypassed. Port-based classifiers will have an unrealistic view of the types of traffic being exchanged over the network, particularly for new emerging mobile applications [21], most of which rely on HTTP and HTTPS. Nevertheless, port-based classification can still be useful for legacy applications [32,34] such as DNS or SMTP that use their default assigned port numbers, particularly for contexts where accuracy is not a major concern (e.g. traffic monitoring).

The historical evolution of traffic classification methods shows that payload inspection [11,39] has emerged after port-based [38] classification became unreliable. Then, statistical and ML [7,16,23,24,33,36,40–48] techniques emerged to overcome the limitations of most previous methods. Hybrid techniques [18,32,34,35,49,50] received some attention with different combination approaches.

DOI: 10.1002/nem

Moreover, the literature shows few miscellaneous [17,51,52] and graphical techniques [13,14,29–31] that were less tracked.

2.3.1. Payload inspection techniques

Most payload inspection techniques [11,39] rely on deep packet inspection (DPI), [53] which checks the packet payload against a set of known protocol signatures (e.g. '\GET' signature in web traffic). DPI will first need to parse the packet headers, up to layer 4 (shallow packet inspection), which is essential to identify the flow to which the packet belongs. To classify the packet, DPI will then need to inspect the entire payload, beyond the layer 4 header, to match applications' signatures.

The matching mechanism itself can be extended. For instance, statistical protocol identification (SPID) [49] uses entropy-based comparisons of probability distributions, relying on the payload content. It measures, for example, the frequency at which all of the possible 256 values occur in a packet. For example, the message type code Ox16 (SSL server hello packet) should have higher frequency in SSL traffic.

Most payload methods rely on DPI; a few, however, disclose the payload using alternative techniques. For instance, exchanging similar payload contents [54] may indicate the use of P2P applications, based on the likelihood that a P2P peer usually redistributes the same content it receives to other peers.

Alternatively, blind classification techniques rely on general traffic attributes without inquiring payload analysis. These include simple statistical, statistical ML and graphical techniques.

2.3.2. Simple statistical techniques

Since researchers are more concerned with the technical significance, we categorize most approaches relying on statistical derived terms (heuristics, behavioral, profiling and characterization) under the simple statistical group, at the technique level in Figure 1. This group includes basic statistical techniques [55,56] and extended ones such as heuristics [25,28] and profiles [13,17,26,27,37,57,58].

Statistical methods are based on the underlying assumption that the traffic at the network and transport layers might have some statistical attributes that are unique for certain applications. For this purpose, a group of general traffic attributes was early defined in Moore *et al.* [59] for flow discrimination.

Basic statistical. Basic statistical techniques rely on general traffic attributes and simple statistical properties to identify applications. For instance, using probability density functions of packet sizes [55,56] and inter-arrival times, standard application protocols can be identified with more than 87% overall accuracy. Statistical methods can be extended by constructing sets of experimentally validated rules, such as heuristics or profiles, to describe traffic attributes for specific applications.

Heuristics. A heuristic is an approximation about statistical traffic attributes, generally defined as a set of rules [25,28]. An example of heuristics [25] is the concurrent use of TCP and UDP sessions with the same port number between two hosts, which supposes the existence of P2P activity. Port-based classification [38] is categorized as a simple heuristic rule relying on port numbers.

Profiles. Profiling and behavioral techniques measure the heterogeneity level and formalize it in a metric value (e.g. entropy) that can be used to compare traffic objects. Profiling and protocol modeling can build the normal behavior for each application, which is useful for anomaly detection [20] in security contexts. Profiling techniques are also applied in traffic classification [13,17,26,37,57,58] to provide higher-level views of the statistical characteristics of applications [37,57], hosts [13,17,26] or even users [58].

For instance, profiling P2P applications based on the payload length and direction [37] can yield to more than 90% recall. At a higher level, the user behavior [58] can provide discriminative information for some applications (e.g. nightly download activity of P2P users). Host profiles were explored for traffic classification and expressed in several ways, such as host interactions in BLINC [13] and number of connected networks [26]. For instance, BLINC models host interactions at the application layer using port numbers and IP addresses, together with few traffic statistics. BLINC, can yield up to 95% P2P accuracy.

DOI: 10.1002/nem

Unconstrained endpoint profiling (UEP) [17] is another promising approach suggesting a fundamental change in host profiling. Key differences exist with state-of-the-art approaches, such as BLINC [13], although both methods are used for host profiling. First, UEP is different in design by actively crawling web information (e.g. Google engine) instead of exclusively relying on network traces. When no or sampled packet traces are available, UEP outperforms BLINC, which needs a sufficient traffic mix to obtain the necessary traffic statistics. Second, UEP is able to provide finer-grained host classes such as Kazaa or Yahoo chat, whereas BLINC provides generic host classes such as P2P or chat. Third, UEP is online capable by requiring a single observed packet, whereas BLINC might be challenging for core network deployments. As per the conducted experiments in Trestian [17], UEP was able to classify over 60% of traffic compared to 53% with BLINC [13].

2.3.3. Statistical machine learning techniques

With ML [60], inferring the relevance of different statistical parameters to the traffic class is automated with less or no required human intervention. The traffic classification problem falls into the pattern recognition [60] scientific discipline. ML algorithms [61] are categorized as supervised, unsupervised, semi-supervised or reinforcement learning (RL). For readers needing deeper insights into ML algorithms, directly relevant papers [60] can be more enlightening.

Unsupervised traffic classification. Unsupervised traffic classification [16,23,46] groups unlabeled traffic objects (e.g. flows [23]) into clusters based on their traffic attributes' vector and according to a given similarity function (e.g. Euclidean distance). Clustering does not require any labeled instances. It is able to provide an in-depth classification of similar traffic types generated by different protocols.

Early clustering [16] mapped traffic objects into generic classes such as bulk transfer or small transactions. To discriminate further among standard application protocols, subsequent clustering approaches (e.g. AutoClass [23]) made use of additional traffic attributes such as packet inter-arrival time and flow duration, yielding 80% accuracy. Different clustering algorithms (k-means, Gaussian mixture, etc.) can identify encrypted and P2P applications with more than 90% accuracy. Clustering methods can be easily adapted for online classification [35] by classifying flows based on the first few packets.

Supervised traffic classification. Supervised traffic classifiers build a model during the training phase based on a set of pre-labeled samples. During classification, this model will map new instances to output classes based on their traffic attributes. A wide range of supervised algorithms [7,36,40–43] with various features were proposed for traffic classification. For instance, context-dependent classifiers, such as Viterbi [41] and hidden Markov [43], assume that different classes are interrelated. However, most supervised models used for traffic classification are context free. Particularly, statistical supervised classifiers such as k-nearest neighbor (kNN) [44] or naïve Bayes [42], together with support vector machines (SVM) [36], are able to detect standard and P2P application protocols with more than 90% accuracy. The same applies to complex algorithms such as artificial neural network (ANN) [45], whose accuracy results ranged from 85% to 90%. Some supervised algorithms can be adapted for real-time deployments such as SVM [36] and decision trees [7]. Particularly, decision trees [7] are able to detect encrypted, standard and P2P applications on high link speeds and at high accuracy rates, reaching, in some cases, more than 95%.

Semi-supervised traffic classification. Semi-supervised [24,47] classification relies on a mixture of labeled and unlabeled input samples that are fed into unsupervised clustering algorithms. Based on few labeled flows, semi-supervised approaches can yield high-accuracy results ranging from 90% to 97% and covering a variety of standard and P2P application protocols.

Reinforcement learning traffic classification. Reinforcement learning (RL) [48] algorithms are developed to interact with dynamic environments where labeled samples and examples of optimal outputs are not explicitly provided, but must instead be discovered by a process of trial and error. Many of the existing ML techniques have their origin in different scientific disciplines. Particularly,

DOI: 10.1002/nem

RL's widest application is in the field of intelligent control and robots. To the best of our knowledge, RL has not yet been explored for Internet traffic classification.

2.3.4. Graphical techniques

Graphical techniques [13,14,29–31] illustrate interactions in computer networks. Edges represent exchanges of interest among nodes representing hosts or Internet protocol (IP) addresses. The underlying assumption is that hosts involved in the same application should reveal specific patterns of interactions at the application and the network layers. Visualizing graph patterns is generally suggestive; however, graph metrics (e.g. graph diameter) make the intuitions precise, which allows for appropriate classification. Graphical methods include graphlets [13], motifs [14], traffic activity graphs (TAG) [31] and traffic dispersion graphs (TDG) [29,30], as depicted in Figure 2.

Graphlets. Graphlets are used in BLINC [13] to reflect host interactions at the application layer. A sample of P2P graphlets, shown in Figure 2(a) (taken from Karagiannis *et al.* [13]), portrays a peer functional role by capturing the relationship between the use of source and destination ports. A library of these graphlets is supposed to classify hosts by specifying the closest match. BLINC is primarily used for host profiling (section 2.3.2); nevertheless, it inspired many subsequent works using graphical techniques.

Motifs. Advanced graph-mining techniques [14] search for frequently recurring patterns of basic structural elements called motifs. Figure 2(b) (taken from Allan *et al.* [14]) shows a sample motif composed of three nodes. Applications are identified based on the node's contributions in the set of motifs mined for each application during the training phase. With this approach, 85% of hosts can be identified as examined across seven protocols including P2P.

Social networks. Social network graphs [30,31] include TAG [31] and TDG [30,31], which detect host communities involved in the same application activity. TAGs are simply referred to as TDGs [30] to incorporate temporal relations between connections. Edges in a TDG are labeled in the order in which the corresponding node interactions were observed, as depicted in Figure 2(c) (taken from Iliofotou *et al.* [30]). In the TDG of Figure 2(d) (taken from Iliofotou *et al.* [29]), the P2P decentralized architecture is captured by high diameters and high connection degree. TAGs proved to be able to detect standard application protocols. TDGs [29] are more suitable for detecting P2P applications such as FastTrack and Soribada.

2.3.5. Miscellaneous traffic classification techniques

Few approaches [17,51,52] were shown in the literature with less measurable traction in their directions. The most prominent examples include content-aware [52] and distributed [51] classifiers. These approaches raise many concerns at the administrative and technical levels that need to be investigated by future works.

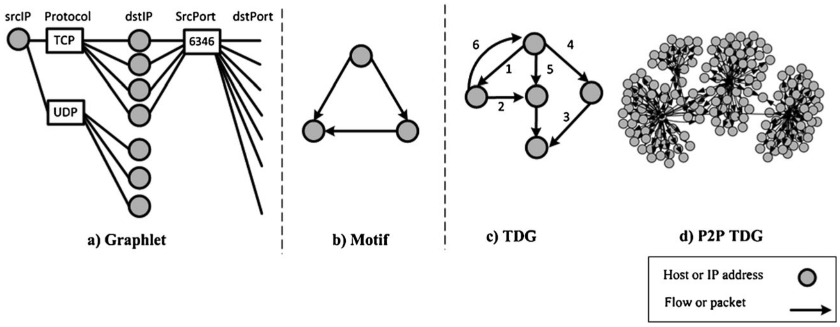

Figure 2. Examples of graphical patterns used in traffic classification: (a) graphlet; (b) motif; (c) TDG; (d) P2P TDG

DOI: 10.1002/nem

2.3.6. Hybrid traffic classification techniques

One of the research trends in traffic classification is to combine different classification techniques, referred to as hybrid or multi-classifier systems.

Although focusing on one at the core of the classification decision, many works in the literature [18,32,34,35,49,50] incorporate more than one technique. For instance, SPID [49] is a hybrid technique integrating DPI with statistical analysis. Its traffic models contain a set of fingerprints represented as probability distributions of different traffic attributes. Examples of payload measurements at the application layer include byte frequencies and offsets for common byte values. SPID showed promising results with an average 92% recall in identifying standard and P2P application protocols. Enhanced SPID [62] can identify additional applications in real time by using a smaller-size fingerprint database through a modified set of attributes such as the number of direction changes and the first payload size. Enhanced SPID can identify 17 standard application protocols including real-time transport protocol (RTP), real-time messaging protocol (RTMP), Internet relay chat (IRC) together with progressive tunneled video download protocols.

ML-based multi-classifiers [63–65] are developed based on intelligent combination and meta-learning algorithms such as ensemble classifiers [63], boosting [60] and gating networks [64] that provide optimized classification decisions. Such systems can achieve higher accuracy than any single classifier, and are more robust to changes affecting the classification input. In fact, boosting results in a combination of a sequence of designed classifiers with better performance than each individual classifier. For example, the adaptive boost (Adaboost) [2,65] meta-learning algorithm outperforms decision trees [7] in classifying encrypted traffic.

However, to the best of our knowledge, ML-based multi-classifiers are not yet used in traffic classification problems. They are mostly applied to speech and image recognition and other disciplines. In the context of network traffic classification, researchers have only referred to simplified approaches [18,32]. Examples included resorting [18] to host-based detection after payload inspection fails or relying on simple decision-making systems [32] when combining multiple techniques.

3. TOWARD AN OPTIMAL TRAFFIC CLASSIFICATION MODEL

Despite remarkable advances in traffic classification, a large proportion of today's network traffic is still unidentified and an optimal classification model is not yet defined. Determining an optimal classification model is a multistage process during which the research community has to work in collaboration.

In this section, we recommend key features and research requirements for an optimal classifier (as detailed in Table 4), which remains an open research question. Since this work is a survey paper rather than a report of experimental results, benchmark studies are left for future work. Instead, we summarize the key findings obtained in previous comparisons (as detailed in Table 3).

3.1. Evaluation of traffic classification methods

In order to provide a certain level of confidence in the classifier's results, comparisons should be performed referring to a well-defined validation method and evaluation metrics. Validation results are used to build the ground truth data, i.e. the set of annotated objects used as reference for validating and evaluating various classification methods. Thus validation methods should be reputed by the highest classification accuracy level. DPI [53] (explained in Section 2.3.1) and agent-based [66] techniques are most commonly used in collecting ground truth results. Agent-based validation consists of deploying special middleware programs on each end system. The agent adds extra information to outgoing packets (e.g. application label), based on which a central server can directly determine the application name and type.

Evaluation metrics indicate the preference of one method over another. In order to quantify the evaluation process, basic and composed classification metrics are used. Figure 3 illustrates basic evaluation metrics for a binary classification scenario, with two classes A and B. In the figure, perfect circles represent the ground truth. The ellipsoid forms, deviated and deformed in reference to the

DOI: 10.1002/nem

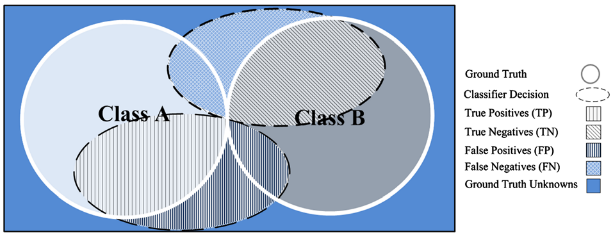

Figure 3. Basic evaluation metrics for a binary classification scenario (classes A and B)

original circles, represent the classifier's decision. Unknown classification decisions are illustrated by the zone outside both circles (for the ground truth) and outside both ellipsoids (for the classifier). When the actual class for the classified instances is A, true positives (TP) are instances classified as A and false negatives (FN) are those classified as B. When the actual class for the classified instances is B, true negatives (TN) are instances classified as B and false positives (FP) are those classified as A. Classifiers should maximize TP and TN while minimizing FN and FP. To the illustration in Figure 3, this refers to restoring the two ellipsoids to the original circle places and re-forming them to near-perfect circular shapes. Unknowns should be eliminated in all cases.

To provide deeper insights into the classifier's performance, composed evaluation metrics (examples are shown in Table 1) can be used. As per most of the surveyed works, the overall accuracy (averaged for all classes) is the most commonly used. However, when the tested dataset is unbalanced, results would be biased towards the most dominant applications. A confusion matrix, instead, is able to illustrate the basic classification results on a per class basis, based on which composed metrics can be inferred. Table 2 shows an example of a confusion matrix for a three-class (A, B and C) scenario, where N_{ij} denotes the number of instances with actual class i classified as j.

3.2. *Comparison and discussion of existing methods*

In this section, we compare the key features of the main published methods, as highlighted in the few existing comparisons between approaches [5–8].

To date, DPI and agent-based methods are the most accurate classification techniques. Payload-based classification is less preferred because of its privacy breaching and its inability to identify encrypted

Table 1. Examples of composed classification metrics for one class (class A)

Metrics	Significancecfh	Formula	Optimal value
Accuracy (precision)	The ratio of traffic instances correctly classified as class A to the total number of instances classified as class A	$\frac{\text{TP}}{(\text{TP}+\text{FP})}$	
Sensitivity (recall)	The ratio of traffic instances correctly classified as class A to the number of actual class A instances	$\frac{\text{TP}}{(\text{TP}+\text{FP})}$	Maximized to 1
Completeness	The ratio of instances associated to class A to the number of actual class A instances	$\frac{(\text{TP}+\text{FP})}{(\text{TP}+\text{FN})}$	Equal to 1
F-measure	A measure that combines precision and recall	$2.\frac{\text{precision.recall}}{(\text{precision}+\text{recall})}$	Equal to 1

Table 2. A confusion matrix for three-class multi-classification scenario (classes A, B and C)

		Predicted class		
Confusion matrix		A	B	C
Actual class	A	N_{aa}	N_{ab}	N_{ac}
	B	N_{ba}	N_{bb}	N_{bc}
	C	N_{ca}	N_{cb}	N_{cc}

traffic. Non-payload classification methods have received more attention in the literature. These include simple statistical, statistical ML and graphical techniques.

As reported by many papers, statistical methods have a relatively low computational requirement while providing accurate results (up to 90%) for standard, encrypted and P2P protocol applications. However, statistically defined profiles may be exhibited by more than one application (e.g. P2P peers and highly active web clients [26]). Moreover, statistical rules can become very complex when expressing variations in multidimensional spaces for large datasets, which might require tedious human interventions. Statistical techniques are therefore useful for simple traffic classification tasks, where few applications and low-dimensional spaces of attributes are analyzed.

Alternatively, ML-based classifiers can partially (or even fully) automate the classification process while providing increased accuracy in a high-dimensional space of traffic attributes. In particular, unsupervised ML techniques (e.g. AutoClass [23]) offer rapid classification with the ability to classify unidentified applications. As reported in many papers, clustering can provide accurate results (up to 90%) for standard, encrypted and P2P protocol applications, without depending on training sets. However, identifying applications that do not predominate in any of the clusters obtained is one of their major limitations. Consequently, clustering is best suited to online deployment, as a first step in the classification process, where the traffic is completely unidentified.

Supervised classifiers are generally more accurate than unsupervised classifiers, as shown in earlier comparisons [5,6]. As documented in many papers, they can identify standard, encrypted and P2P protocol applications with very high accuracy (up to 95%). However, each supervised learning algorithm has specific strengths and weaknesses. For example, context-dependent classifiers are preferred for describing inter-class dependence. kNN classifiers are simple classification models with zero training time. Naïve Bayes classifiers might be preferred for their low memory requirements and immunity to irrelevant traffic attributes. Most of these models are built on assumptions (e.g. Gaussian distribution [42]) that may be hard to generalize in some cases.

When considering datasets with too many attributes [61], more complex algorithms (e.g. ANN) are preferred, to fit data variations more readily while providing more stable performance. In the presence of many irrelevant attributes [61], other alternatives (e.g. SVM) might be preferred. In fact, some experimental comparisons [8] point to a preference for SVM over ANN in traffic classification, in terms of precision, generalization, performance and training time. SVM performed better than did ANN in classifying many standard and application protocols. There are many differences between these two algorithms [60], such as ANN being much more dependent on the training data than SVM. However, there are limitations common to both algorithms, including complexity and overfitting [61]. Recent comparisons [7] point to a preference for decision trees over other ML algorithms, particularly for real-time and encrypted traffic classifications.

Most supervised methods suffer from persistent limitations related to the training process itself and to relatively high training costs. To be as generalized as possible, supervised classifiers should train on large significant traffic sets. When training data are scarce, semi-supervised learning is preferred, yielding up to 97% accuracy with better computational efficiency.

Graphical techniques rely on traffic attributes that are less affected by network dynamics and are more suited to detecting dynamic applications such as P2P. However, graphical techniques might require the capture of a large number of host interactions, which are mostly available at the service provider level. They are usually used as helper techniques, such as assisting DPI [29].

Finally, multi-classifier is a promising research approach that may not only inherit the advantages of a variety of techniques but also complement and cross-validate their results. However, its efficiency should be considered, given the increased input dimensionality.

Classification features are summarized in Table 3, showing more recent and selective methods for each category at the three levels. Results for the overall classification accuracy are shown as presented in the original papers. Most features are qualitatively presented as high (H), medium (M), low (L) or not applicable (N). Here, H indicates that the feature is strongly offered by a method, with L indicating the opposite. M corresponds to moderate cases where neither L nor H applies and N indicates that a feature is not applicable for a method. Note that M cases may require additional future assessment.

Nevertheless, further research effort and benchmarking will be required to unravel the relations and affinities between the various techniques, in the context of traffic classification.

Table 3. Comparison of selective state-of-art methods

Method description							General features															
							Technique							Input			Output					
Reference	Publicationyear	Technique			Input	Output [Classification object] [Metrics for all classes] [P2P protocols]	Immune to missing traffic attributes	Suitable for large set of traffic attributes	Simple to update	Fast to train	Independent of the training set	Able to describe inter-class dependence	Simple classification model	Protects user privacy	Immune to network dynamics	Immune against obfuscation	Overall classification accuracy	Provides fast classification	Ensures memory saving	Ability to detect encrypted applications	Ability to classify unknown applications	Scope of detected applications
42	2011	Machine Learning	Supervised	Bayes	Flow size, duration, packet size and packet rate	[Flow] [accuracy: 96%] [BitTorrent, eMule, PPLive, Skype]	H	L	M	M	L	L	H	H	M	L	M	M	H	M	M	M
44	2009			kNN	TCP header fields, packet payload size	[Flow] [accuracy: 90%] [BitTorrent, eMule]	M	L	M	H	L	L	H	H	H	L	M	M	L	M	M	M
36	2012			SVM	Flow duration, idle time, packets, bytes, size and t inter-arrival times	[Flow] [accuracy: 84%] [BitTorrent, Gnutella, live-streaming, Skype]	L	H	L	L	L	L	L	H	L	L	M	L	M	M	M	M
45	2010			ANN	Packet number, size, inter-arrival time (in both direction)	[Flow] [accuracy: 86%] [Bittorrent, eDonkey, eMule, PPstream, PPlive]	L	H	L	L	L	L	L	H	L	L	M	L	M	M	M	M
7	2012			Decision Trees	Packet number, size, inter-arrival times, transferred bytes	[Flow] [accuracy > 90%] [Kazaa, BitTorrent, GnuTella]	H	M	H	H	L	L	H	H	L	L	M	H	M	H	M	M
43	2008			Markov, Viterbi	Packets' sizes and inter-arrival times	[Flow] [accuracy > 90%] [eDonkey, PPlive]	M	M	M	M	L	H	M	H	L	L	M	M	H	M	M	M
46	2012			Unsupervised	Flow size, duration, packets' number, sizes and inter-arrival times (in both direction)	[Flow] [accuracy, F-measure > 90%] [BitTorrent]	M	M	M	N	N	L	H	H	L	L	M	H	H	L	H	Mr
47	2012			Semi Supervised	Number of packets and packet size	[Flow] [accuracy > 90%, F-measure 60–90%] [BitTorrent]	M	M	M	M	M	L	M	H	H	L	M	M	M	M	H	M

DOI: 10.1002/nem

Table 3. Continued

14	2009	Graph Based	Motifs	IP addresses; port numbers	[Host] [accuracy: 85%] [Kazaa]	M	M	L	N	N	L	L	H	H	M	M	L	M	M	L	L
31	2008		Social graphs	IP addresses	[Host community] [–] [Bittorrent]	M	M	M	N	N	L	M	H	H	M	M	M	M	M	L	L
28	2009	Simple statistical	Heuristic	Netflow TCP flags	[Flow] [accuracy > 83%] [P2P]	M	M	H	N	N	L	H	H	H	L	M	H	H	M	L	L
58	2010		Profiles	UDP packet size, IP addresses, port numbers, traffic volume (in both directions)	[Flow][accuracy: 96%] [eMule, Skype, Bittorrent]	M	M	H	N	N	L	H	H	H	L	M	H	H	M	L	L
39	2012	Payload	Payload inspection	First 100 payload bytes	[Flow] [precision, recall > 90%] [Bittorrent, Gnutella]	M	M	M	N	N	L	H	L	H	L	M	M	M	L	M	H
34	2010	Hybrid	Statistical & Payload inspection	IP addresses, port numbers, first 32 payload bytes	[Byte] [accuracy: 97%] [BitTorrent, eDonkey]	M	M	M	N	N	L	M	L	H	L	M	M	M	L	M	H

DOI: 10.1002/nem

3.3. Requirements for appropriate benchmarks

Clearly, one of the main obstacles to advancing research in the field is the lack of appropriate benchmarks for existing techniques. Most comparisons (as described in Section 3.2) have not considered the categories in the methods being assessed. Determining the optimal classification model is a multistep process that should involve systematic comparisons. For example, techniques that provide the best output results while using the same input type should be selected from each category as expressed by our taxonomy. Techniques representing different categories should then be compared.

The research community should also promote convergence to common standards covering related terminologies, procedures and policies. Table 4 shows some of these requirements, from the perspective of our taxonomy. For instance, publicly available DPI tools [67–69] differ in the set of signatures and the payload-matching mechanisms used, which may yield inconsistent ground truth results. Validation tools should be standardized. Nevertheless, most of the problems involve complex policy issues rather than being purely technical. For example, as highlighted in Table 4, the existing heterogeneity in the definitions of traffic objects and classes at the output level is among the obstacles that have to be overcome by the research community. The lack of publicly available traffic traces at the input level is another critical obstacle to further advances in traffic classification. Sharing disclosed data is mainly constrained by privacy concerns. In a symbolic step to address part of these difficulties, we regularly collect recent traffic traces obtained from a real operational network of a multi-branch institution, while maintaining the privacy of our full payload captures. To achieve this, we follow two sharing models. First, we publish non-payload data, named *tcrsg-collection* [70], through the Internet Measurement Data Catalog public repository. In particular, we publish ground truth data with anonymized IP addresses, together with various statistical parameters (61 variables) that are useful for blind classification. Second, as part of our ongoing research work, we act as an entity [71] offering evaluation of payload-based classifiers following the move-code-to-data [72] model. We invite researchers to send their tools for offline evaluation on protected full-payload data before sending the classification results back to them. We hope that this sharing model becomes common practice for other researchers and interested parties.

3.4. Future classification

3.4.1. Vendor classification engines

In this section, we present an overview of classification techniques used in commercial traffic management and network security products [73–76]. These include routers, firewalls, IPS, SWG and traffic shapers. Unfortunately, there is very little information available about the protocol classification performed in most of these proprietary systems.

An example of a proprietary algorithm used in TippingPoint systems is Protocol Identification via Statistical Analysis (PISA) [75]. PISA creates a ten-dimensional representation of each fingerprint for each protocol, based on a training set of captured traffic. It uses simple average and standard deviation values of general flow attributes (packet size, inter-arrivals) in both directions, in addition to the Shannon entropy of the data at the application layer. It uses k-means to cluster flows for standard and P2P applications including Skype. However, one of its main limitations is the required number of packets to be analyzed before a flow is identified. For example, Skype results[75] stabilize after the 600th packet.

Although relying on common techniques, commercial products often rely on proprietary methods. For instance, Juniper's [73] DPI mechanism matches patterns in the first packet of a session using deterministic finite automata. It has the ability to chain signatures and to specify a maximum number of transactions wherein the signature must occur to be a match.

Cisco routers use network-based application recognition (NBAR) [74], which relies on DPI and many application-specific attributes. It is a state-oriented classification mechanism that supports applications with dynamically negotiated port numbers, such as RTP. It is able to support sub-classifications, such as HTTP user agent, content type and uniform resource locator (URL). NBAR2 is an extended version of NBAR that supports evasive applications such as Skype and Tor, cloud-based applications such as Office 365, and even mobile applications such as FaceTime. Cisco's Service Control Engine (SCE) is a dedicated hardware DPI appliance that incorporates protocol state analysis together with behavioral and heuristic analysis.

In most of these commercial products, the increasing requirement for content awareness and application visibility explains the DPI integration with statistical and ML techniques, which are

DOI: 10.1002/nem

Table 4. Research requirements and features of future classifiers

	Future classifiers' features	Future research requirements
[INPUT]	Should include a minimal set of discriminative traffic attributes with less or no payload; attributes should be difficult to obfuscate, immune to network dynamics and adapted to new technology trends (e.g. IPv6)	To explore new information sources (e.g. application layer attributes, cloud-based reputation analysis [73–75] … etc.); to have one publically available, free and open source tool for attribute extraction (e.g. similar to Tstat [69]); to define a standard list of discriminative traffic attributes and protocol signatures associated with each application (similar to the IANA [38] list for registered port numbers); to offer public repositories of recent traffic traces (e.g. similar to Internet Measurement Data Catalog [70]) with enough payload obtained from various real operative networks; to establish entities that offer execution on their traces (move-code-to-data) as to have minimal privacy sensitivities
[TECHNIQUE]	Should train quickly, with less dependence on the training data; should be easy to update with low complexity; should provide accurate (minimizing error rates) and online classification (minimizing computational costs); should handle multi-label host classification; might integrate more than one technique built on intelligent multi-classifier algorithms; should be adapted to new technology trends (e.g. IPv6, CCN, SDN, SaaS)	To select main state-of-art techniques and algorithms for comparison; to accomplish community-driven and validated benchmarks based on standard formats and procedures. Techniques that provide the best output results while using the same input type should be selected from each category. Techniques representing different categories should be compared; to define evaluation metrics and thresholds associated with 'accurate' and 'online' classification; to have one publically available, free and open source traffic classification platform (e.g. TIE [81]) and validation tool (e.g. L7-filter [67], nDPI [68]), with well-defined algorithms and consistent signatures. Multi-label [79] classification and multi-classifier [63–65] algorithms should be explored in future works; to propose new techniques adapted to roaming users (e.g. Software as a Service classification), high link speeds (e.g. special hardware [80]) and new technology trends (e.g. SDN)
[OUTPUT]	Should identify a wide scope of contemporary application protocols, controlled at granular level, including multi-channel, Web 2.0, P2P, encrypted tunnels, mobile and social networking services based applications	To define standard traffic classification objects and classes

DOI: 10.1002/nem

port-based in Juniper [73], behavioral in IPOQUE [53] and SVM-based in Websense [77], together with SSL decryption. We now consider the future requirements that are driving the market for next-generation products and the key features of future classifiers.

3.4.2. Future classifiers' features

As mentioned above, determining the optimal classification model remains an open research question. Table 4 summarizes the key features recommended for future traffic classification models.

Theoretically, an optimal traffic classification model would be one that can best achieve all of the features shown in Table 4. As mentioned above, defining the optimal traffic classification method (i.e. the optimal path in Figure 1) remains an open research question. Nevertheless, there is one essential question: is it feasible to design a single technique that can offer all the features in Table 4 simultaneously?

In practice, according to most existing comparison work [6], no single technique has been able to uniformly outperform all others in a majority of scenarios, when submitted to an analysis that takes into account different networks and application types. In fact, it has been acknowledged [10] that the classification decision requires many trade-offs involving reliability, performance and privacy protection. For example, the ability to identify unknown applications might best be achieved through clustering, whereas an ability to update the classification model rapidly might require a supervised decision tree. Theoretically, no single technique is capable of providing both of these capabilities simultaneously.

From this perspective, it seems reasonable to assert that a multi-classifier design (as discussed in Section 2.3.6) would best answer these requirements. We believe that multi-classifiers would achieve the required trade-offs for future traffic classification because they should theoretically inherit the advantages of many combined techniques. We consider that the multi-classifier is a promising approach that should be considered amongst candidate models for future classifiers. This is validated by many of the detected trends, both in academic research and in network security products. In fact, most of the current products described in Section 3.4.1 already integrate DPI with helper techniques. Their next-generation series will integrate behavioral techniques and correlate traffic information across multiple vectors [76], together with full-stack visibility. Moreover, the research work described in Section 2.3.6 provided a few valid attempts in this direction. The multi-classifier examples in Szabo et al. [32] and Callado *et al.* [50] were able to outperform individual classifiers. Moreover, the multi-classifier in Callado *et al.* [50] was shown to be robust against bias towards any scenario, which was an issue with previous classification algorithms. Nevertheless, future multi-classifiers should be developed using more complex decision systems that use confidence values [78] and intelligent combination algorithms [60,63,64]. This should involve a wider range of techniques, including DPI, Auto-class and decision trees. The potentially high computational cost associated with multi-classifiers should be addressed in future work.

As shown in Table 4, for the new world of very high link speeds, user mobility and collaboration services, future methods should be able to identify new application trends such as Web 2.0 and mobile applications. They should also be customized for new technology trends while answering new modeling needs. For example, host classification might refer to multi-label classification [79], where an instance can belong to more than one class at the same time. New sources of information should be explored, such as application-layer message sizes (instead of packet sizes) and external cloud-based information sources [73,74,77], and specialized hardware [80] designs would have to be implemented.

4. CONCLUSION

In this article, we have surveyed network traffic classification methods. We have proposed a multilevel taxonomy that characterizes methods at three different levels and which is suitable for comparison studies. By highlighting current research trends, we have shown that classification methods are mostly dominated by ML techniques, with a slight bias towards decision trees for rapid and accurate classification. Statistical and graphical techniques are used as helper technologies. Payload inspection methods are less preferable, despite their high accuracy, because they can breach privacy. However, we have pointed

DOI: 10.1002/nem

out the lack of an optimal model for traffic classification, for which we have identified a critical need for appropriate benchmarks. We have also recommended key features for future classification models, while identifying their future challenges. Finally, based on a few promising approaches in research work and commercial products, we argue that a well-designed multi-classifier model could theoretically meet most future requirements. However, this is left for future work. In the future, we should see an increasing number of studies relevant to multi-classifier design and associated efficiency issues.

With the revolutionary advances in network technology and the proliferation of Internet applications, the decreasing reliability of most existing methods will be recognized in the near future. In the new world of very high link speeds, user mobility and collaboration services, future methods will have to refer to more innovative system architectures while mining more advanced traffic attributes. In the longer term, future designs should adapt to next-generation networks as part of the network management portfolio. Opening new avenues in the field of traffic classification will be vital for managing future networks.

ACKNOWLEDGEMENT

This work has been partially supported by Spanish MICINN under project TEC2011-22579.

REFERENCES

1. Zink T, Waldvogel M. Bittorrent traffic obfuscation: a chase towards semantic traffic identification. In *12th IEEE International Conference on Peer-to-Peer Computing*, 2012; 126–137.
2. McCarthy C, Zincir-Heywood A. An investigation on identifying SSL traffic. In *IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)* 2011; 115–122.
3. Mujtaba G, Parish DJ. Detection of applications within encrypted tunnels using packet size distributions. In *International Conference for Internet Technology and Secured Transactions (ICITST)* 2009; 1–6.
4. Shen X, Yu H, Buford J, Akon M. Handbook of Peer-to-Peer Networking. Springer: New York, 2010; 3–113.
5. Nguyen TTT, Armitage G. A survey of techniques for Internet traffic classification using machine learning. *IEEE Communications Surveys and Tutorials* 2007; **10**: 56–76.
6. Verticale G, Giacomazzi P. Performance evaluation of a machine learning algorithm for early application identification. In International Multiconference on Computer Science and Information Technology (IMCSIT). IEEE: New York, 2008; 845–849.
7. Hu L, Zhang L. Real-time Internet traffic identification based on decision trees. In World Automation Congress (WAC). IEEE: New York, 2012; 1–3.
8. Pradhan A. Network traffic classification using support vector machines and artificial neural networks. *International Journal of Computer Applications* 2011; **8**: 8–12.
9. Zhang M, John W, Claffy K, Brownlee N. State of the art in traffic classification: a research review. In *Passive and Active Network Measurement Conference (PAM2009)*, Korea, 2009; 1–2.
10. Callado A, Kamienski C, Szabo G, Gero B, Kelner J, Fernandes S, Sadok D. A survey on Internet traffic identification. *IEEE Communications Surveys and Tutorials* 2009; **11**: 37–52.
11. Sen S, Spatscheck O, Wang D. Accurate, scalable in-network identification of P2P traffic using application signatures. In *13th International Conference on the World Wide Web* 2004; 512–521.
12. Khalife J, Verdejo J, Hajjar A. Performance of OpenDPI in identifying sampled network traffic. *Journal of Networks* 2013; **8**: 71–81.
13. Karagiannis T, Papagiannaki K, Foloutsos M. BLINC: multilevel traffic classification in the dark. In Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM). ACM: New York, 2005; 229–240.
14. Allan E, Turkett W, Fulp E. Using network motifs to identify application protocols. In *IEEE Global Telecommunications Conference (GLOBECOM)* 2009; 4266–4272.
15. Park B, Hong J, Won Y. Toward fine-grained traffic classification. *IEEE Communications Magazine* 2011; **49**: 104–111.
16. McGregor A, Hall M, Lorier P, Brunskill J. Flow clustering using machine learning techniques. Passive and Active Network Measurement. Lecture Notes in Computer Science, Springer: New York, 2004; 205–214.
17. Trestian I, Ranjan S, Kuzmanovic A, Nucci A. Googling the Internet: profiling Internet endpoints via the World Wide Web. *IEEE/ACM Transactions on Networking* 2010; **18**: 666–679.
18. Keralapura R, Nucci A, Chuah, C. A novel self-learning architecture for P2P traffic classification in high speed networks. *Computer Networks* 2010; **54**: 1055–1068.
19. Kim J, Yoon S, Kim M. Study on traffic classification taxonomy for multilateral, hierarchical traffic classification. In *14th Asia–Pacific Network Operations and Management Symposium (APNOMS)* 2012; 1–4.
20. Kim Y, Jo J, Suh K. Baseline profile stability for network anomaly detection. In 3rd International Conference on Information Technology New Generations (ITNG). IEEE: New York, 2006; 720–725.
21. Choi Y, Chung JY, Park B, Hong JWK. Automated classifier generation for application-level mobile traffic identification. In Network Operations and Management Symposium (NOMS). IEEE: New York, 2012; 1075–1081.

DOI: 10.1002/nem

22. Bonfiglio D, Mellia M, Meo M, Rossi D, Tofanelli P. Revealing Skype traffic: when randomness plays with you. In Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), Vol. **37**. ACM: New York, 2007; 37–48.
23. Zander S, Nguyen T, Armitage G. Automated traffic classification and application identification using machine learning. In *Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)* 2005; 250–257.
24. Yuan J, Li Z, Yuan R. Information entropy-based clustering method for unsupervised Internet traffic classification. In IEEE International Conference on Communications (ICC). IEEE: New York, 2008; 1588–1592.
25. Karagiannis T, Broido A, Faloustsos M, Claffy K. Transport layer identification of P2P traffic. In Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement (IMC). ACM: New York, 2004; 121–134.
26. Cheng WQ, Gong, J, Ding W. Identifying BT-like P2P traffic by the discreteness of remote hosts. In 32nd Conference on Local Computer Networks. IEEE: New York, 2007; 237–238.
27. Chang S, Daniels T. Correlation-based node behaviour profiling for enterprise network security. In Third International Conference on Emerging Security Information, Systems and Technologies (SECURWARE). IEEE: New York, 2009; 298–305.
28. Jinsong W, Weiwei L, Yan Z, Tao L, Zilong W. P2P traffic identification based on NetFlow TCP flag. In International Conference on Future Computer and Communication (ICFCC). IEEE: New York, 2009; 700–703.
29. Iliofotou M, Kim H, Faloutsos M, Mitzenmacher M, Pappu P, Varghese G. Graption: a graph-based P2P traffic classification framework for the Internet backbone. *Computer Networks* 2011; **55**: 1909–1920.
30. Iliofotou M, Pappu P, Faloutsos M, Mitzenmacher M, Singh S, Varghese G. Network monitoring using traffic dispersion graphs (TDGs). In Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC). ACM: New York, 2007; 315–320.
31. Jin Y, Sharafuddin E, Zhang Z. Unveiling core network-wide communication patterns through application traffic activity graph decomposition. In Proceedings of the 11th International Joint Conference on Measurement and Modeling of Computer Systems (SIGMETRICS). ACM: New York, 2009; 49–60.
32. Szabo G, Szabo I, Orincsay D. Accurate traffic classification. In IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM). IEEE: New York, 2007; 1–8.
33. Zhang J, Chen C, Xiang Y, Zhou W. Classification of correlated Internet traffic flows. In 11th IEEE International Conference on Trust, Security, Privacy in Computing and Communications. IEEE: New York, 2012; 490–496.
34. Aceto G, Dainotti A, de Donato W, Pescapé A. PortLoad: taking the best of two worlds in traffic classification. In IEEE Conference on Computer Communications Workshops (INFOCOM). IEEE: New York, 2010; 1–5.
35. Bernaille L, Teixeira R. Early recognition of encrypted applications. In Proceedings of the 8th International Conference on Passive and Active Network Measurement (PAM'07). Springer: Berlin, 2007; 165–175.
36. Tabatabaei T, Karray F, Kamel M. Early internet traffic recognition based on machine learning methods. In 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE). IEEE: New York, 2012; 1–5.
37. Wang P, Guan X, Qin T. P2P traffic identification based on the signatures of key packets. In 14th IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD). IEEE: New York, 2009; 1–5.
38. Touch J, Lear E, Mankin A, Kojo M, Ono K, Stiemerling M, Eggert L, Melnikov A, Eddy W. Service Name and Transport Protocol Port Number Registry. The Internet Assigned Numbers Authority (IANA) 2013. Available: http://www.iana.org/assignments/port-numbers. [13 September 2013].
39. Yeganeh S, Eftekhar M, Ganjali Y, Keralapura R, Nucci A. CUTE: traffic classification using terms. In 21st International Conference on Computer Communications and Networks (ICCCN). IEEE: New York, 2012; 1–9.
40. Dehghani F, Movahhedinia N, Khayyambashi MR, Kianian S. Real-time traffic classification based on statistical payload content features. In Second International Workshop on Intelligent Systems and Applications (ISA). IEEE: New York, 2010; 1–4.
41. Wright C, Monrose F, Masson G. On inferring application protocol behaviours in encrypted network traffic. *Journal of Machine Learning Research* 2006; **7**: 2745–2769.
42. Zhenxiang L, Mingbo H, Song L, Xin W. Research of P2P traffic comprehensive identification methods. In Proceedings of the 2011 International Conference on Network Computing and Information Security (NCIS), Vol. 1. IEEE: New York, 2011; 307–310.
43. Dainotti A, de Donato W, Pescapé A, Salvo Rossi P. Classification of network traffic via packet-level hidden Markov models. In IEEE Global Telecommunications Conference (GLOBECOM). IEEE: New York, 2008; 1–5.
44. Huang S, Chen K, Liu C, Liang A. A statistical-feature-based approach to Internet traffic classification using machine learning. In International Conference on Ultra Modern Telecommunications and Workshops (ICUMT). IEEE: New York, 2009; 1–6.
45. Gu C, Zhuang S. A novel P2P traffic classification approach using back propagation neural network. In 12th IEEE International Conference on Communication Technology (ICCT). IEEE: New York, 2010; 52–55.
46. Wang Y, Xiang Y, Zhang J, Yu S. Internet traffic clustering with constraints. In Eighth International Wireless Communications and Mobile Computing Conference (IWCMC). IEEE: New York, 2012; 619–624.
47. Zhang J, Chen C, Xiang Y, Zhou W. Semi-supervised, compound classification of network traffic. In 32nd International Conference on Distributed Computing Systems Workshops. IEEE: New York, 2012; 617–621.
48. Qiang W, Zhongli Z. Reinforcement learning model, algorithms and its application. In *International Conference on Mechatronic Science, Electric Engineering and Computers (MEC)* 2011; 1143–1146.

49. Hjelmvik E, John W. Statistical protocol identification with SPID: preliminary results. Statistical Protocol IDentification (SPID) Project. Available: http://spid.sourceforge.net/sncnw09-hjelmvik_john-CR.pdf. [13 September 2013].
50. Callado A, Kelner J, Sadok D, Kamienski CA, Fernandes S. Better network traffic identification through the independent combination of techniques. *Journal of Network and Computer Applications* 2010; **33**: 433–446.
51. Xu B, Chen M, Hu C. DEAPFI: a distributed extensible architecture for P2P flow identification. In IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC). IEEE: New York, 2009; 59–64.
52. Miruta R, Stanuica C, Borcoci E. Content-aware classification methods. In Ninth International Conference on Communications (COMM). IEEE: New York, 2012; 241–244.
53. IPOQUE Deep packet inspection-technology. Available: http://www.ipoque.com/sites/default/files/mediafiles/documents/ipoque_WP_Deep_Packet_Inspection_2009_DPI.pdf [13 September 2013].
54. Lu X, Duan H, Li X. Identification of P2P traffic based on the content redistribution characteristics. In International Symposium on Communications and Information Technologies (ISCIT ‘07). IEEE: New York, 2007; 596–601.
55. Crotti M, Dusi M, Gringoli F, Salgarelli L. Traffic classification through simple statistical fingerprinting. *SIGCOMM Computer Communication Review* 2007; **37**: 5–16.
56. Wang X, Parish D. Optimised multi-stage TCP traffic classifier based on packet size distributions. In Proceedings of the 3rd International Conference on Communication Theory, Reliability, and Quality of Service. IEEE: New York, 2010; 98–103.
57. Zhu K, Hu H, Yi P. Identifying P2P flow with behaviour characteristics. In *Second International Conference on Future Computer and Communication (ICFCC)* IEEE: Wuhan, 2010; 140–142.
58. Ullah I, Doyen G, Bonnet G, Gaiti D. A survey of synthesis of user behaviour measurements in P2P streaming systems. *IEEE Communications Surveys and Tutorials* 2011; **14**: 734–749.
59. Moore A, Zuev D, Crogan M. Discriminators for use in flow-based classification. Queen Mary University of London, 2005. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.7450&rep=rep1&type=pdf [13 September 2013].
60. Theodoridis S, Koutroumbas K. *Pattern Recognition* (4th edn). Elsevier/Academic Press: New York, 2003; 18–23.
61. Kotsiantis SB. Supervised machine learning: a review of classification techniques. *Informatica* 2007; **31**: 249–268.
62. Köhnen C, Überall C, Adamsky F, Rakočević V, Rajarajan M, Jäger R. Enhancements to Statistical Protocol IDentification (SPID) for self-organised QoS in LANs. In *Proceedings of the 19th International Conference on Computer Communications and Networks (ICCCN)* 2010; 1–6.
63. Dietterich T. Ensemble methods in machine learning. In *Proceedings of the 1st International Workshop on Multiple Classifier Systems (MCS)* 2000; 1–15.
64. Su M, Basu M. Gating improves neural network performance. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Vol. 3, 2001; 2159–2164.
65. Alshammari R, Zincir-Heywood A. Can encrypted traffic be identified without port numbers, IP addresses, payload inspection? *Computer Networks* 2011; 1326–1350.
66. Szabo G, Orincsay D, Malomsoky S, Szabo I. On the validation of traffic classification algorithms. In Proceedings of the 9th International Conference on Passive and Active Network Measurement. Springer: Berlin, 2008; 72–81.
67. L7-filter, Application Layer Packet classifier for Linux. Available: http://l7-filter.clearfoundation.com [13 September 2013].
68. nDPI, Open and Extensible GPLv3 Deep Packet Inspection Library. Available: http://www.ntop.org [13 September 2013].
69. Tstat, TCP STatistic and Analysis Tool. Available: http://www.tstat.polito.it [13 September 2013].
70. Ground truth and statistical data. Available: http://imdc.datcat.org/data/1-NF6L-V=tcrsg-D09.gt_param [13 September 2013].
71. Traffic Classification Research Group. Available: [http://tcrsg.ul.edu.lb]. [13 September 2013].
72. Dainotti A, Pescapé A, Claffy K. Issues and future directions in traffic classification. *IEEE Networks* 2012; **26**: 35–40.
73. Juniper Intrusion Detection and Prevention. Available: http://www.juniper.net/techpubs/software/junos-security/junos-security10.0/junos-security-swconfig-security/jd0e49039.html#jd0e49039 [13 September 2013].
74. Cisco Next Generation Network-Based Application Recognition (NBAR2) Protocol Pack. Available: http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6558/ps6616/qa_C67-723689.pdf [13 September 2013].
75. Dhamankar R, King R. Protocol Identification via Statistical Analysis (PISA). *TippingPoint Technologies*, Black Hat 2007. Available: http://www.blackhat.com/presentations/bh-usa-07/Dhamankar_and_King/Whitepaper/bh-usa-07-dhamankar_and_king-WP.pdf [15 December 2013].
76. Pescatore J, Young G. Defining next-generation network intrusion prevention. Sourcefire computer and network security company 2013 Available: http://www.sourcefire.com/content/gartner-research-defining-next-generation-network-intrusion-prevention [15 December 2013].
77. Websense Advanced Classification Engine (ACE). Available: http://www.websense.com/content/security-overview-websense-ace.aspx [15 December 2013].
78. Ichino M, Maeda H, Yamashita T, Hoshi K, Komatsu N, Takeshita K, Tsujino M, Iwashita M, Yoshino H. Internet traffic classification using score level fusion of multiple classifier. In Ninth IEEE/ACIS International Conference on Computer and Information Science. IEEE: New York, 2010; 105–110.
79. Tsoumakas G, Katakis I. Multilabel classification: an overview. *International Journal of Data Warehousing and Mining* 2007; **3**: 1–13.
80. Zhou Z, Song T, Fu W, Rocket TC. A high throughput traffic classification architecture. In International Conference on Computing, Networking and Communications (ICNC). IEEE: New York, 2012; 407–411.
81. Dainotti A, de Donato W, Pescapé A, Ventre G. TIE: a community-oriented traffic classification. In Proceedings of the 1st International Workshop on Traffic Monitoring and Analysis (TMA). Springer: Berlin, 2009; 64–74.

DOI: 10.1002/nem

AUTHORS' BIOGRAPHIES

Jawad Khalife holds a master's degree in telecommunications networks, University of Saint-Joseph, Beirut, Lebanon. Since 2002, besides his teaching activities, he has worked as a computer network engineer in the central administration of the Lebanese University, Beirut, Lebanon.

Amjad S. Hajjar is an assistant professor at the Faculty of Engineering of the Lebanese University, Beirut, Lebanon. He obtained his PhD in 1992 in computer-aided design (CAD) from the university of Paris VI, France.

Jesús Díaz Verdejo is a professor in the Department of Signal Theory, Telematics and Communications of the University of Granada. He received his BSc in physics in 1989 and a PhD degree in physics in 1995 from the University of Granada, Spain.

DOI: 10.1002/nem