Towards the Deployment of Machine Learning Solutions in Network Traffic Classification: A Systematic Survey

Fannia Pacheco, Ernesto Expósito, Mathieu Gineste, Cédric Baudoin, Jose Aguilar

Towards the deployment of Machine Learning solutions in network traffic classification: A systematic survey

Fannia Pacheco, Ernesto Exposito, Mathieu Gineste, Cedric Baudoin, and Jose Aguilar

***Abstract*—Traffic analysis is a compound of strategies intended to find relationships, patterns, anomalies, and misconfigurations, among others things, in Internet traffic. In particular, traffic classification is a subgroup of strategies in this field that aims at identifying the application's name or type of Internet traffic. Nowadays, traffic classification has become a challenging task due to the rise of new technologies, such as traffic encryption and encapsulation, which decrease the performance of classical traffic classification strategies. Machine Learning gains interest as a new direction in this field, showing signs of future success, such as knowledge extraction from encrypted traffic, and more accurate Quality of Service management. Machine Learning is fast becoming a key tool to build traffic classification solutions in real network traffic scenarios; in this sense, the purpose of this investigation is to explore the elements that allow this technique to work in the traffic classification field. Therefore, a systematic review is introduced based on the steps to achieve traffic classification by using Machine Learning techniques. The main aim is to understand and to identify the procedures followed by the existing works to achieve their goals. As a result, this survey paper finds a set of trends derived from the analysis performed on this domain; in this manner, the authors expect to outline future directions for Machine Learning based traffic classification.**

***Index Terms*—Internet traffic, Traffic classification, Machine learning, traffic monitoring.**

I. Introduction

Traffic analysis is the complete process that starts from intercepting traffic data to finding relationships, patterns, anomalies, and misconfigurations, among others things, in the Internet network. Particularly, traffic classification is a subgroup of strategies in this field that aims at classifying the Internet traffic into predefined categories, such as normal or abnormal traffic, the type of application (streaming, web browsing, VoIP, etc) or the name of the application (YouTube, Netflix, Facebook, etc). Network traffic classification is important because of several reasons that involve: a) Troubleshooting tasks: the main objective is to locate faulty network devices, device/software misconfigurations, locate the point of packet losses, network errors, etc. b) Security: avoid malware or prevent intrusion to private information. c) Quality of Service (QoS) management to guarantee the overall acceptability of an application or service perceived by end-users. In this field, identifying or classifying the name or type of application in the network helps to treat some of the beforehand aspects. For instance, identifying different applications from traffic is critical to manage bandwidth resources and to ensure QoS requirements.

In the past, traffic classification relied on the port-based approach where each application was identified by its registered and known port, defined by the Internet Assigned Numbers Authority (IANA) [1]. This approach became unreliable and inaccurate due to, among other factors, the proliferation of new applications with unregistered or random generated ports. Another approach that gained a lot of popularity in this field is called Deep Packet Inspection (DPI). DPI performs a matching between the packet payload and a set of stored signatures to classify network traffic. However, DPI fails when privacy policies and laws prevent accessing to the packet content, as well as the case of protocol obfuscation or encapsulation. In order to overcome the former issues, Machine Learning (ML) emerged as a suitable solution, not only for the traffic classification task, but also for prediction and new knowledge discovery, among other things. In this context, statistical features of IP flows are commonly extracted from network traces, and they are stored to generate historical data. In this way, different ML models can be trained with this historical data, and new incoming flows can be analyzed with such models.

This PhD thesis has been proposed to explore and to apply ML techniques over Satellite Communications in order to improve the QoS. In a satellite communication,

A. Related works

In this section, several review papers are studied in order to find trends, challenges and general steps to perform traffic analysis, and in particular, traffic classification. Traffic analysis with ML started being used in 2005, however, several problems persist due to the traffic evolution and scalability, among others. The work in [2] presents a review of the traffic classification advances using ML in its early years. Even though, most of the presented works were deployed in an offline manner, the authors also addressed to online deployments, establishing some critical operational requirements for the ML models. A more recent review, presented in [3], identifies several of these

Fannia Pacheco and Ernesto Exposito are with Univ Pau & Pays Adour, E2S UPPA, LIUPPA, EA3000, Anglet, 64600, France. Emails: {f.pacheco,ernesto.exposito-garcia}@univ-pau.fr

Mathieu Gineste and Cedric Baudoin are with D'epartement : Business Line Telecommunication, R&D department, Thales Alenia Space, TOULOUSE, 31100, France. Emails: {mathieu.gineste,cedric.baudoin}@thalesaleniaspace.com

Jose Aguilar is with CEMISID, Dpto. de Computacion, Facultad de Ingenieria, Universidad de Los Andes, 5101, Mérida, Venezuela. Email: aguilar@ula.ve

problems and presents some future directions in this field. On one hand, this work studies some of the main traffic classification problems, which can be summarized as follows: i) the data available with their ground truth is limited, ii) the scalability of traffic classification solutions is a challenge, iii) adaptive solutions are required due to the dynamism and evolution of the network, and iv) the solutions require a correct validation. On the other hand, these future directions encourage to the execution of more rigorous evaluations and comparisons of the ML approaches, to the development of tools for the ground truth definition, and to the use of multiclassifier systems, among other things. Moreover, the ML approaches require to fulfill different challenges, such as a provided performance, the management of the increasing amount of traffic and transmission rates, and the reconfiguration capabilities, as it is exposed in [4], and similarly in [5].

Recent advances in this field are presented by [6], focused on supervised and unsupervised techniques for traffic classification. The authors studied several works using Bayes based classifiers, Neural Networks (NNs), and Decision Trees (DT). Additionally, clustering techniques, such as DBSCAN, Expectation Maximization (EM) based approach and K-means, were studied for traffic classification. Some advantages and disadvantages are identified, in order to outline the necessary improvements for each approach.

Encrypted traffic has become a new way to prevent intrusion into transmitted information in the Internet network. ML is highly suitable for analyzing these type of communications due to it does not intrude into the packet content; for instance, in some cases the statistical behavior of the connection might be sufficient. In this context, [7] reports a comprehensive review of several studies focused on encrypted traffic. The survey studies some of the encryption protocols, their packet structure and standard behavior in the network, as well as, the observable features that can be extracted for traffic analysis. In addition, [8] surveys the most common methodologies for traffic classification.

Nowadays, abnormal traffic detection in the network has become one of the most important topics in traffic analysis. The purpose is to discover or to characterize anomalies that might come from malicious or unintentional sources affecting the network infrastructure, business or personal privacy, and digital economy, among others. Traffic analysis is necessary for this particular domain. [9] presents a comprehensive study where the ML techniques represent more than 30% of the solutions found in this review. More works in this field are found in [10], [11], [12].

In general, traffic classification can be achieved by a variety of ML techniques, in different domains, and with different objectives. However, one of the main concerns, in all the survey papers, is the lack of public data, which can be considered as a core resource for applying ML approaches. The difficulty of defining the ground truth values of the data collected, and the implementation of the ML solutions, represent some challenges and limitations. Additionally, another important aspect is the evolution of the Internet network that requires adaptive or self-configuring solutions to assure reliable traffic analysis procedures. Conversely, more efforts are required by the research community in order to propose ML solutions that can deal with the drawbacks beforehand discussed.

B. Contributions

The present survey paper tries to gather together different approaches, strategies and procedures about how and when to use ML techniques for traffic classification. It will study the process from the monitoring stage to the implementation of ML solutions. The main aim is to offer a comprehensive guide to practitioners in the field that intend to use ML techniques for traffic classification. In this sense, this paper is focused on the steps to achieve traffic classification based on the experience found by the scientific community. It is important to mention that the study is focused on the traffic classification at the IP level. For instance, this research will offer an overview of approaches that can be used to improve the QoS at the operator network level. In brief, the main contributions are summarized as follows.

- It provides a comprehensive workflow to understand how to achieve traffic classification with ML techniques over IP flows.
- It studies each step of the workflow correlated with the efforts found in the literature.
- It provides a set of paths that current approaches follow based on the workflow defined.
- As a final result, it provides a general overview of the traffic classification problem, and future directions according with the previous results.

To conclude this section, it is worth remarking the main difference of this systematic review to other related works is the approach proposed to present the papers. The reviewed papers are organized following the general procedures to apply ML techniques in the Internet traffic classification domain.

C. Organization

The remainder of this paper is organized as follows. Section II briefly presents the problem of traffic classification. This section presents at first some basis of the ML techniques and network traffic, to finally conclude with some of the most common approaches. Section III presents the methodology deployed by the authors for reviewing the papers, which is based on the ML procedures for knowledge extraction. Following the guidelines in the former section, Section IV reports common approaches to perform data collection in the Internet network. Section V reports strategies to extract features from the observed traffic. Section VI presents some methods used to reduce or to select the extracted features. In Section VII, it is studied how the works select the ML algorithms for traffic classification. Section VIII is intended to know the efforts found in the literature for implementing the ML solutions. Section IX analyses the results of the review, and Section X outlines the conclusions of the survey paper.

II. Background

In Section II-A, a synthesis of the ML basis, general steps and classical algorithms, are presented. Following, in Section II-B, an overview of the traffic classification approaches is given.

A. Machine Learning introduction

Machine Learning (ML) techniques are very popular approaches to identify and to classify patterns in different domains. Its main objective is to give to the computers automatic learning capabilities, where the machines are able to extract knowledge from a process under certain conditions. ML tries to extract knowledge from a set of features or attributes, which represents the measurable properties of a process or observed phenomena. In this way, the learning process is performed by training different models, i.e., classification, prediction or clustering model; and their use depends on the problem characteristics. The knowledge extraction is handled by a ML model, which is built with historical experiences recorded from case studies.

Different methodologies can be found in the literature to apply ML; for instance, [13] presents a set of iterative steps to discover knowledge in big databases. The work reports the traditional method of turning data into knowledge, called Knowledge Discovery in Databases (KDD). The main steps of the KDD comprise data selection, processing, transformation, mining and interpretation/evaluation. The data mining component refers to the application of data mining methods, which determine patterns from the data. The majority of the data mining methods are based on the ML techniques. We present as follows, the general steps to achieve knowledge discovery with ML techniques.

1) Data collection: this step aims at gathering information regarding a case study. Measuring procedures are established, in order to capture data either from physical or digital sensors. Such data describes the current or historical status, which is used to define the experimental testbed. A testbed is composed of all the software, hardware, and networking components, among others held by the process of interest. This testbed is necessary for building the model (learning and testing) with the ML techniques. Samples are captured and gathered from multiple scenarios set in the testbed.

2) Feature extraction (FE): it is one of the most important steps due to it allows measuring or computing features that might give information about the state of the process. In brief, a FE procedure computes different metrics that reflect specific properties in the data collected. The main aim is to obtain descriptors that better characterize the problem. The result of the FE process is a structured table formed by columns of attributes where each row is a sample, with an additional optional column with the current status of each sample (commonly called label or class). In case that the status is unknown, the samples are unlabeled.

Data processing procedures can be performed in order to delete unwanted missing values and to clean the data, among other things. This last one is related to outliers detection that might disrupt into the ML solution performance. Also, the data can be transformed through normalization or aggregation operations over the attributes values. In the aggregation procedures, the features are combined into a single feature that would be more meaningful to the problem.

At this stage, one could start with an initial study to comprehend the data. For instance, with labeled data, the table can be treated to find class-imbalance, which is the scenario where there are one or more classes with a considerable higher amount of samples than another class(es). Class-imbalance data can bias some ML models to learn more from a class than another. One way to treat the class-imbalance data is reducing the number of irrelevant samples from a class, if possible, as it is exposed in [14].

Finally, it is important to mention that the FE process can be embedded in the ML algorithm; moreover, an historical dataset might not be available, and the ML model should learn from scratch. These particularities will be extended in Section II-A4.

3) Feature reduction (FR) and selection (FS): this is an optional step that allows selecting or reducing the number of extracted features. FR is to create new attributes using the original ones, while FS is to find a low set of attributes that better describes a process. These steps aim at decreasing problems such as time consumption and curse of dimensionality, among others. Surveys about the performance and comprehension of the FR and FS processes are presented by [15], [16]. FR and FS are commonly divided into Filter, Wrapper and Embedded approaches, which in turn can be developed by supervised and unsupervised strategies. In the supervised strategy, the objective is to find the features that most contribute to define the classification decision. In the unsupervised strategy, the main aim is to determine the features that allow the grouping of the data.

In the filter approach, a score is given to each feature using a metric that measure their relevance. The features are ranked and the most relevant are those ones that meet an accepted threshold. Correlation analysis is a simple filter approach where relationships between pair of features are found by computing a correlation coefficient. Other techniques categorized as filter are the Gini Index [17], the Information Gain [18], the Laplacian Score and the Sparsity Scores [19], [20], among others. Unsupervised learning can be used by filter algorithms in order to find the best basis features [21], [22], [23], [24], or to select the features through structured sparsity regularization models, which preserve the cluster structure of the instances composing the dataset [25], [26], [27]. On the other hand, supervised learning is commonly used by wrapper approaches where an objective function is defined in order to determine the effect of different feature sets over a classifier's accuracy. The features that provide the best classification performance are selected. Genetic algorithms and sequential search strategies have been widely used as wrapper methods [28], [29].

Moreover, some ML algorithms include the FS process embedded into their model design, such as regularized regression models and decision trees based models [30], [31]. Finally, other techniques are focused on generating more representative features based on the original ones [32], [33]. These techniques can be found in a procedure, commonly denominated as Feature Generation, explicitly separated from the FR and FS processes.

4) Algorithm selection and model construction: The results of the previous phases lead to a dataset that contains historical information about the case study. The historical dataset is a key resource for building ML models. Different ML algorithms

have been developed and tested for solving tasks, such as classification, clustering and regression. The selection of the ML algorithm is related to the problem to solve or the type of knowledge, that the practitioners want to discover.

In ML, there are two classical types of learning, supervised and unsupervised learning. Most of the supervised learning algorithms adjust their model parameters minimizing the error between the model output and the real expected output of an input. This means that the historical data has to be labeled. On the other hand, unsupervised algorithms try to find relationships between the inputs without beforehand knowledge of the outputs. These relationships can be similarities, proximities, and statistical relationships, among others. As a derived consequence of the learning process, the supervised algorithms are commonly used to perform classification tasks, while the unsupervised ones are rather used to cluster inputs in order to find anomalous or similar behaviors between themselves. In general, the ML model and the type of learning is associated with the type of problem to solve.

Different categorizations of the ML algorithms can be found [34], [35], [36]. For instance, a general classification groups the classical supervised algorithms based on statistical model, trees, rules, and neural networks (NNs), among others. In addition, several approaches do not necessarily belong to the groups mentioned above, and can be grouped into parametric and non parametric. For instance, in a parametric model the aim is to determine the parameters that minimize a cost function, such as Support Vector Machine (SVM). A non-parametric algorithm is the K Nearest Neighbors (KNN), which gather together similar samples through measuring their distance. In the unsupervised approach, one categorization divides the techniques into clustering based on prototypes, hierarchical methods and density based methods, among others.

Nowadays, there is a big variety of ML algorithms not only based on supervised and unsupervised learning. For instance, semi-supervised algorithms take advantage of unlabeled data to train classifiers, either training a classifier with the labeled samples and then evaluating the unlabeled ones in the classifier, or using unsupervised approaches for the unlabeled samples. Hybrid approaches are also found through the combination between supervised and unsupervised learning, due to the presence of labeled and unlabeled samples in datasets [37]. Moreover, ensemble techniques use a variety of ML models (commonly classifiers) and combine their results through a combination strategy. Strategies such as Bagging and Boosting are widely used to build ensemble models [38]. There are more advanced techniques such as, incremental learning that aims at updating in an online manner the ML models with new incoming inputs [39]. Finally, reinforcement learning is focused on the on-line performance (commonly cumulative reward) that is maximized any time that an action is taken [40].

5) Validation of classification models: This segment overviews the most common validation approaches for the classification solutions. Supervised learning requires the beforehand knowledge of the sample labels, which are key information to validate the ML models. The usual approach is to divide the dataset into a training and a test set. The ML models are built with the training set, while the resulting models are assessed by the test set in order to evaluate their prediction capabilities. Given the model predictions and the ground truth labels of the test set, several performance metrics can be deployed to quantify the classification capability of a ML solution. For instance, [41] presents a study of the classification performance metrics divided into the type of classification to achieve: binary, multi-class, multi-labeled and hierarchical. Binary classification occurs when an input sample can be classified into only one of two distinct classes. On the contrary, multi-class classification implies that the input can be classified into only one class within a pool of classes. Multi-labeled classification allows the classification of an input sample into more than one class in the pool of classes. Finally, hierarchical classification is similar to the multi-class classification but with more granularity, in the sense that the principal classes are divided into lower levels of subclasses.

In order to validate ML models, one of the most common approaches is to measure their performance in terms of the classification capabilities. Several relationships can be found among the model predictions and the ground truth labels, such as the number of samples correctly and incorrectly assigned to a class, among others. These counts allow computing metrics, such as Accuracy, Precision, Recall, F-score, Receiver Operating Characteristic (ROC), etc. For instance, for binary classification with a positive and a negative class, the counts of true positives (TP), false negatives (FN), true negatives (TN), and false positives (FP) can be used to compute the performance of the classifier through the sensitivity and the specificity metrics. A combination of these metrics, such as the F-score and the ROC, offers more precise information about the performance of the classifiers for both classes. Particularly, the ROC curve is obtained by computing the sensitivity and specificity varying the classifier's discrimination threshold [42]. In the ROC curve, the ideal value represents a lot of sensitivity and specificity (a very good diagnostic method). The result is an interpretable figure that illustrates the performance of the classifier in different operating points; in addition, the Area Under the Curve (AUC) can be computed to obtain a compact measure from the ROC. The analysis above can be mapped to multiclass-problems by computing overall performance measures with micro or macro averages of the binary performances. Each of these metrics exposes different aspects of the model performance [43], [44].

The traditional model evaluation approach is to use a training and test set in order to train a model and to compute the performance metrics, respectively. Moreover, cross-validation is an alternative approach for model validation that divides the data in k subsets. One subset is used as the test set, and the rest for training the model. The same procedure is performed for the k folds, and the global performance is given by the combination (average) of the evaluation score of each test set. Different variations of the cross-validation approach are available such as the leave-p-out and the leave-one-out.

In closing, there are more advanced approaches that measure the statistical significance of the classifier performance given different test scenarios, such as the Friedman test, Wilcoxon test, among others. These scenarios propose different partitions

of the dataset, different initial conditions on the classifier's setting, or embedded random initializations in the training process. These approaches are mainly used to compare a set of classifiers, and to select the one with the highest performance under different conditions, as it is exposed in [45].

B. Traffic classification

In traffic classification, several trends can be found to classify, comprehend, diagnose or observe the status of the network. A complete taxonomy of traffic classification can be found in [46]. Nonetheless, a modification of this taxonomy is proposed in Fig. 1, which in turn is focused on the branch of interest, Machine learning. This figure counts with five main divisions: *Data*, *Techniques*, *Feature engineering*, *Algorithm selection* and *Output*. A brief description of each component is given as follows.

- *Data*: it refers to the type of input data used to create the traffic classification solution. It is noticeable that the traffic data can be encrypted or unencrypted. The traffic can be labeled either by DPI tools or real time captures. More details about this component will be provided in Section IV
- *Techniques*: four main branches are detected, such as Machine learning, Statistical based, Behavioral based and Payload inspection. In this section, these four approaches are briefly described in order to have a general idea of how they work. However, the ML branch will be extended throughout the survey paper.
- *Feature engineering*: this component concerns only to the ML branch, and presents the potential and existing trends of the feature analysis techniques applied over IP flows, according to the strategies presented in II-A2 and II-A3. This section will be extended in Section V, while a formal definition of an IP flow is given in II-B1.
- *Algorithm selection*: This component depicts the available approaches for building ML solutions, according to the learning process used (supervised, unsupervised, hybrids, etc), and the task to accomplish (classification, clustering, etc).
- *Output*: Finally, the output principally depends on the objective to achieve, such as classifying the flows into categories or application names, system status, etc.

Following the items above, it is introduced this section as follows. Section II-B1 formally defines an IP flow; which is the most common representation of communication sessions in the Internet network. Following, the classical traffic classification techniques are described: Payload inspection in Section II-B2, Statistical based techniques in Section II-B3, Behavioral techniques in Section II-B4 and ML techniques in Section II-B5.

1) IP flow definition: In traffic classification, it is commonly used the term "flow" to describe a set of packets that are transmitted from a source to a destination. An IP flow, according to [47], is a set of packets or frames in the network that can be intercepted at a point in the network during a time interval. The packets belonging to the same flow share several common properties, such as: a) one or more packets, with transport or application header fields (e.g. source and destination IP address, port number and type, among others), b) characteristics of the packets as the number of MPLS labels, and c) additional fields, such as next-hop IP address, etc. Therefore, it can be outlined the following classical definition of an unicast flow F_i:

Definition 1: An IP flow F_i is described by the set,

$$F_i = \{H_i, P_i\} \tag{1}$$

where H_i is the header of the flow i, and $P_i = \{p_{i1}, ..., p_{in}\}$ is a set of packets belonging to the flow i.

Definition 2: A flow header H_i is described by the tuple,

$$H_i = (IP_{src}, IP_{dest}, port_{src}, port_{dest}, proto) \tag{2}$$

where IP_{src} and IP_{dest} are the IP source and destination addresses; $port_{src}$ and $port_{dest}$ are the source and destination transport ports, respectively; and $proto$ is the transport protocol.

Additionally, it is defined P_i as in expression 3. P is the complete set of packets passing through a monitoring point. h_k is the header of the packet p_k, and H_i the header of the flow studied.

$$P_i = \{p_k \in P | h_k == H_i\} \tag{3}$$

For an unicast bidirectional flow (assuming that a is the client and b is the server), the definition above is extended in the sense that,

Definition 3: An IP flow F_{ab} is bidirectional when,

$$F_{ab} = F_a \cup F_b \tag{4}$$

where the union of the unidirectional flows is achieved through the matching of some elements in their tuples.

In Definition 2, some port numbers can be reserved for type of applications as it is established by IANA, however, the random generation of the ports is often deployed for most of the applications. In addition, several ports can be opened for a communication session (denoted as a multicast session), which in turn may affect the definition above. In the past, the dimension of the Internet traffic allowed using a matching between an opened port and registered ports (using IANA) to obtain the name of the application. However, the proliferation of new applications with random ports, and the growth of the Internet network, provoked inaccuracies over the port-based approach. They became obsolete, and new means of traffic classification appeared.

2) Payload inspection: Also called Deep Packet Inspection (DPI), this technique analyzes the content of the Internet packets, i.e., the IP header and payload. DPI compares the information extracted from the packets with a set of signatures (previously defined and known) to identify different application protocols. Some of the DPI tools are nDPI, Libprotoident, PACE, L7-filter, and NBAR, among others.

Recently, DPI tools have suffered several drawbacks due to the growing number of new applications and protocols. Particularly, when a new protocol is created, then the DPI tools must be updated; otherwise, they will fail in their prediction getting as result an unknown or an erroneous signature. As

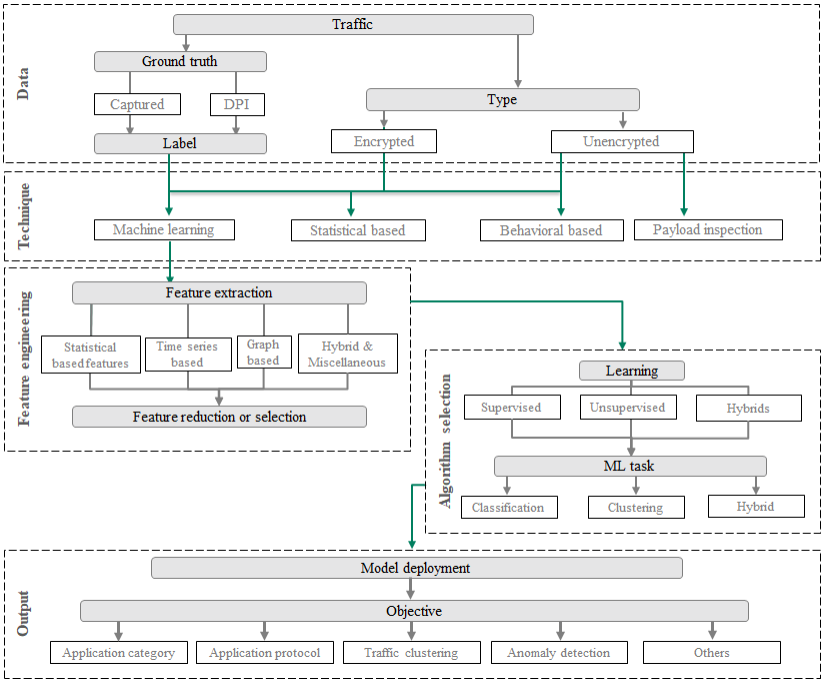

Fig. 1: General view of the traffic classification approaches focused on the ML branch.

a consequence, the list of the tools' signatures has to be constantly updated. On the other hand, DPI is not adequate when a) packet encryption is used to protect the content in communication sessions, b) HTTP2 is deployed for multiplexing the packet content, c) NAT networks are utilized because they unable to differentiate between communication sessions, and d) Virtual private networks (VPNs) are deployed for data privacy and integrity, among others.

Even though, it is clear that DPI tools have particular deficiencies, they are still widely used for traffic classification. Several studies report their accuracy and popularity for traffic classification [48], [49]. Some works have tried to solve the manual setting of signatures when new protocols are detected by the DPI tools [50], [51]. The work in [52] proposes the automatic generation of traffic signatures based on a fixed bit offset mechanism. The work proposes to randomly select packets from flows, and to compare from pair to pair every bit of the packets in the same bit offset. The signatures are taken from a fixed length, recorded and counted, to generate the final signatures. The same authors, in [53], propose another approach using a clustering technique to gather together similar flow behaviors. From the resulting clusters, a set of flows is taken in order to apply a token-based algorithm (Hamsa) and MSA (t-coffee algorithm); both algorithms have been used to extract regular expressions in biology, and adopted for extracting signatures in binary sequences. These new strategies can help to support traffic classification with DPI.

3) Statistical based techniques: The main aim is to find statistical differences between flows, communicating end systems and network configurations, among others. Such differences can be the result of two or more different applications or behaviors, characterized by the statistical properties. In some contexts, statistical distributions can be used to model the network traffic patterns. The construction of probabilistic models has been applied by several works to know different status of the network. The work in [54] introduces a monitoring scenario with public and private IP addresses, and measures statistics for each profile, such as the number of TCP and UDP packets, as well as the number of failed flows. The objective is to construct different statistical distributions, such as negative exponential or Gaussian, to detect the reachability in P2P communications. In a similar manner, the work in [55] proposes the categorization of the flows using statistical distributions. On the other hand, the work in [51] introduces text classification over the packet headers, the result is evaluated by a statistical binary model that will determinate if it is a new signature. The main deficiency of this approach is the static construction of statistical models that do not integrate learning processes. As the previous approach, this disadvantage affects its performance in the presence of dynamic growing and evolution of the Internet traffic patterns. In addition, some of the statistical based approaches are adapted and improved by the ML techniques.

4) Behavioral techniques: This approach commonly tries to find patterns among end-to-end communications in a network. It also studies community patterns where the communities are conformed of hosts at different points.

The most common representation of behavioral patterns in the network is through graph modeling, in which graph theory is used to find highly connected nodes (hosts), number of connections, and opened ports, among others [56]. As an example, [57] analyzes traffic behavior to identify P2P traffic. The first step is to cluster together similar flows through a k-means model. Following, the clusters are represented by a Traffic Dispersion Graph (TDG), where the nodes are represented by the IP addresses and the link between the

nodes are the registered flows. Finally, a set of rules is applied over the graphs to detect the name of the application. These rules take into account features, such as the percentage of nodes and the average node degree of the graph. The work in [58] proposes an approach to identify P2P communities, where the interactions in the network are represented by graphs. The nodes are formed by the tuple (IP, port), and the connections are given by the number of packets interchanged between nodes. P2P networks are identified by using the port distribution of known remote peers, in order to do so, a multinomial classifier is built to decide whether a graph represents one of the known networks.

A different target that is normally studied by these techniques is identifying the traffic activity patterns, such as the works in [59], [60]. For instance, [59] presents the Traffic Analysis Graphs (TAGs) for visually unveiling the behavior of different type of applications. In a TAG, the nodes are the IP addresses and the edges are the flows of interest; the flows of interest are defined according to the purpose of the study, in order to build TAGs that capture relevant traffic activities among hosts. [60] builds bipartite graphs and compute their similarity matrix. This matrix will serve as input to a clustering algorithm (k-means) that will gather together similar nodes.

This approach has also been reported for detecting anomalies. The work in [61] present a general overview of the state of the art in this area.

5) Machine learning: For this particular domain, one of the main goals is to classify traffic based on the status of the Internet network. For such case, IP flows are reported as the most common representation of Internet communications, where representative features can be extracted and used for traffic classification. As it is denoted in Fig. 1, some of the FE approaches used can be divided into statistical flow features, time-series, and graph based, among others. The FR or FS processes are optional. They are normally applied along ML, and it has been demonstrated the benefits added to the ML models in terms of performance [16].

The ML algorithms can be supervised, unsupervised, semi-supervised or hybrids, and it will depend on the ML task to perform and the data available. The ML task is directly linked to the objective of the study. One of the most popular objectives is anomaly detection to prevent network attacks that may cause severe damage among service providers and final users. Moreover, anomaly detection can also be used to identify failure or misconfiguration in the network [9]. On the other hand, application protocol detection also attracts interest in this field, in particular to service providers that want to improve the service offered to their costumers. For instance, improving the QoS is one of main objectives in network resource management.

It is important to mention that the ML approach is able to handle encrypted communication thanks to the FE process deployed. Normally, the features extracted from IP flows do not intrude in the packet content, which allows creating classification models for encrypted communications. However, this approach can encounter problems with the use of HTTP2, VPNs and NAT networks, due to the separation of communication sessions is not explicit.

To conclude this section, in Fig. 1 is noticeable that most of the traffic classification works are focused on and grouped into different objectives, such as detecting the application protocol, category or anomalies. In order to achieve traffic classification, the techniques beforehand discussed can be used; however, this survey paper will only extend the ML branch. In this branch, we will study several works that try to achieve different traffic classification objectives.

III. Methodology

This survey paper differs from the survey papers reviewed, in the way that the works are presented. They are mainly organized following the general procedures defined in Section II-A, which are mapped to the traffic classification problem. In this section, the methodology to survey the papers is detailed, which follows the workflow proposed in Fig. 2. In this figure, a distinction between the blocks that can be performed in an offline (green arrows) and online (yellow arrows) manner is made.

Generally speaking, the offline procedures treat historical datasets stored by the *Data collection* block at one or more monitoring points in the network. The *Data Collection* step allows measuring different scenarios in Internet. This block mainly collects IP flows within a time window. Additionally, this block carries several steps, such as packet management, flow reconstruction and storage. In Section IV, the works related to this step are reviewed. In the offline run, a historical dataset must be collected; on the contrary, in the online run, streams of packets are continuously treated.

Once the data that characterizes the problem is recorded, relevant features are extracted following the approaches in Section V. Likewise, in online and offline phases, the features are computed from the historical dataset and the streams of packets, respectively. At this point, the resulting features can be treated by either FS or FR approaches, to obtain a reduced space or a set of new features. The most common procedures and works are reported in Section VI. The FR or FS procedure also has a connection with the *Algorithm Selection* and *Model deployment* blocks, due to some approaches select the most relevant features based on the performance given by the ML models.

Now, from the original dataset, a new dataset is obtained based on the selected features. In the offline run, the new dataset is used to build models that will allow performing classification and regression tasks, among other things. The *Algorithm Selection* block refers to the procedures and methods intended to select the most adequate ML algorithm. Several works deploy different means of comparison to justify or validate their selection, as it will be pointed out in Section VII. Finally, the *Model Construction* block is focused on the efforts found in the literature for implementing such models on the Internet network. These two last blocks can also be evaluated in an online manner, mainly to offer evolving or upgraded ML solutions. It is also important to mention that this workflow represents a guide that comprises the general steps to use ML for traffic classification. However, the order of these steps may vary, and can be also found combined.

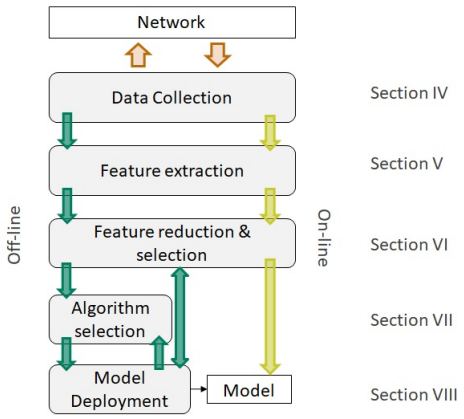

Fig. 2: Workflow for surveying the papers for traffic classification with ML.

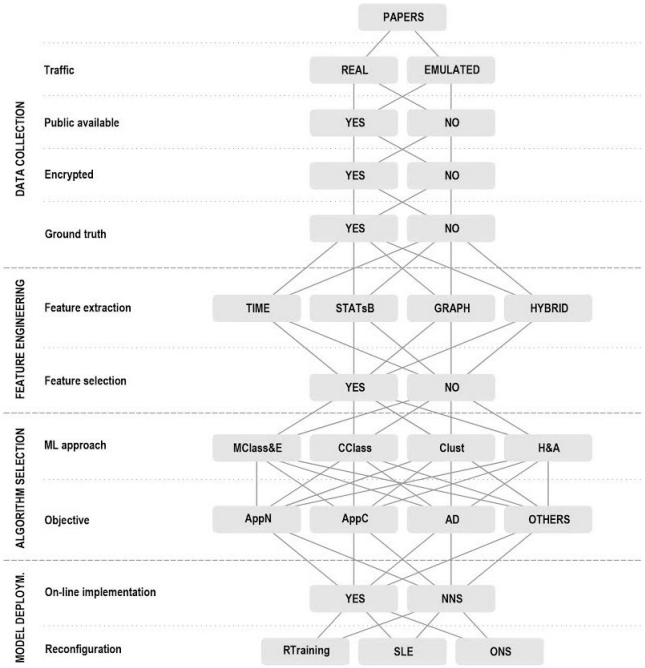

Fig. 3: Workflow of the most important characteristics that will be reviewed in each section.

More concisely stated, particular characteristics are taken from each block in Fig. 2. These characteristics are detailed in Fig. 3, likewise they represent some paths that the reviewed papers commonly follow for traffic classification. In this figure, it can be noticed that for the *Data collection* block, it will be studied if the Internet traffic is: real or emulated, publicly available, encrypted and labeled (ground truth). These aspects will become very important for characterizing the problem. The *Feature Engineering* block comprises the FE and FS approaches used by the reviewed papers. In this sense, four FE approaches were found as the most common ones, such as statistical based (STATsB), graph based (GRAPH), time series based and hybrids approaches. In the FS block, it will be denoted if the reviewed papers performed or not this procedure. Following, for the *Algorithm selection* block, the ML approach used and the objective to achieve are defined. The trends studied are classical classification (CClass), Multi-classification and ensemble approach (MClass&E), clustering for classification and anomaly detection (Clust), and hybrids and advances techniques (H&A). Among the classification objectives studied were found application name (AppN), application category (AppC) and anomaly detection (AD). Also, other objectives are considered, such as user behavior detection and community search, among others. Finally, in the *Model deployment* block, it is differentiated the papers that implemented the ML solution (YES), and the ones that do not do it or do not specify it (NNS). The reconfiguration of the solution will be a key aspect to study, in this sense, it is verified if the reviewed papers offer either a retraining (RTraining), a self-learning or evolving (SLE), or other/non-specified (ONS) process.

The paper search was performed over the most important academic databases. The documents selected have either on their title, abstract or the keywords, terms such as “traffic monitoring”, “traffic analysis”, “Internet classification”, and “encrypted traffic classification”, among others. The papers considered were published between 2010 and 2017, and they were located in academic databases, such as ScienceDirect, ACM, IEEE Xplore and Scopus. It is important to notice that works from conference proceedings and journals were equally taken into account.

IV. Data Collection

Traditionally, historical data is a very important source of knowledge for building ML solutions. A rich and complete set of observations regarding a problem can improve the performance and generalization of the ML models. However, in the traffic classification domain, this aspect is critical due to: complexity and scalability of the Internet network, constant evolution of the traffic, and privacy policies that do not allow data collection, among others. In consequence, real Internet traffic data is hardly available for analysis and knowledge extraction. Several tools and strategies have been developed to cope with this gap, some of these will be studied in this section.

The overall structure of the data collection procedure takes the form of Figure 4. In this figure, an abstraction of three levels is given in the following order : i) the network environment that is defined by the conditions in which the traffic is triggered, such as real, generated or emulated, ii) the Internet network itself, and iii) the data measurement procedure, which refers to how network packets can be collected. The main issues addressed in this section concern to the components that help finding good traffic classification results. First at all, in Section IV-A, a distinction between the network environments to monitor traffic is made. Following, in Section IV-B, the data measurement procedure will be reviewed. This procedure is indifferent to the environment adopted, and takes into account steps such as packet extraction, flow reconstruction and storage, among others. Finally, Section IV-C will outline the importance of labeling traffic flows with their ground truth values.

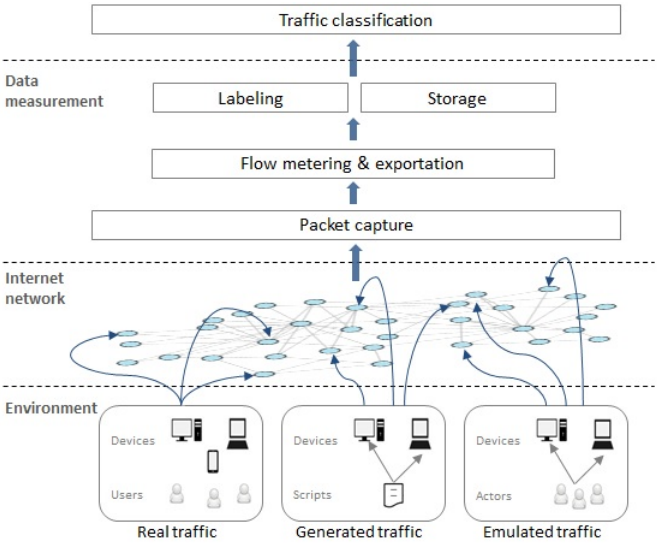

Fig. 4: Main data collection components for traffic classification

A. Network enviroment

In particular, three main trends were found in the reviewed papers: real traffic acquisition, traffic generation and emulation. Real traffic is normally collected from the network, obfuscating private information among communication entities (e.g. clients-servers). Traffic generation tries to simulate similar real traffic conditions by copying or modeling real interactions through scripts. Finally, traffic emulation aims at setting scenarios as close as real ones, where one or more actors can intentionally emulate common interactions in the network.

The optimal solution is to capture real traffic from the network in order to have a reliable source of realistic data; however, this solution is hard to conduct, mainly due to privacy matters. Even though some works manage to obtain realistic conditions for monitoring traffic, the data is hardly ever publicly available for the research community. Some of the publicly available data are listed in Appendix XI-A.

Traffic generation solutions are mainly destined to evaluate network device performance, communication security, resource management, etc. In the literature can be found several works to generate traffic, where their strategies are mainly divided into: flow-based generation and model-based generation. Flow-based generation is to reproduce the content and arrival times of flows captured in real scenarios, the packet content is given by dummy or random payload for technical tests. Some of the flow-based generation tools are BRUTE [62], iPerf [63], and Ostinato [64], among others. For instance, BRUTE is designed to generate Ethernet traffic either IPv4 or IPv6. Ipref measures the maximum achievable bandwidth on IP networks by tuning parameters such as timing, protocols, and buffers. In the same manner, the Ostinato tool can generate traffic with different protocols at different rates. In contrast, model-based approaches for traffic generation try to reproduce the statistical properties of realistic traffic and to represent them into a model, which will later generate traffic for experimental tests. For instance, [65] generates client workload by training a Hidden Markov Model (HMM). The model is built with real human interactions in a network testbed, the main objective is to send a sequence of events to each application via the operating system in order to create synthetic data. The training data is conformed of the event ID, a process IP and the arrival time between events. Finally, the model can inject realistic client-server interactions to create synthetic traffic data. Moreover, the model based generation can be performed at the packet level instead of the application level; as an example, [66] proposes a HMM for synthetic series generation of both packet time and packet size. The work in [67] reviews some of the most popular tools for traffic generation, and proposes a synthetic workload generator that can be used as a platform to place both flow and model based generation approaches. One limitation about these tools is that they are not commonly able to generate encrypted traffic. Encrypted traffic covers a big volume of the current Internet traffic, and it is important to consider its collection for its analysis.

Another approach, to model the communication behaviors, is based on network emulation systems. An emulated network environment involves the construction of a communication configuration between two or more end-points. The general architecture is to set several clients at different virtual or physical machines, to configure monitoring points for each client, and to set data storage components. Traffic can be induced in order to record backward and upward IP flows. The works in [68], [69], [70], [71] present frameworks for traffic emulation, such tools are publicly available, and require adaptation into the traffic classification interests. The platform in [68] presents a framework based on automatic user behavior emulation for emulating the network traffic. Several scenarios, that represent series of user's actions for interacting with applications, are mimicked; this is achieved through a tool that creates automation scripts or macros for Microsoft Windows programs. Similarly, the works in [69] and [70] emulate user behavior recording the graphical user interface, for the generation of Internet traffic. In contrast to the traffic generation approach, emulation of traffic can cover applications that use encryption or traffic encapsulation; this feature is desirable for traffic classification with ML.

B. Data measurement

In this section, it will be presented the common approaches for performing data measurement over the Internet network. Moreover, how the traffic observation is done at monitoring points. These two aspects are considered in order to easily understand how the labeling assignment can be performed, and how the ML solution can be implemented in real network environments. Normally, one of the most common strategies for network traffic monitoring is extracting a set of packets within a time window; this approach is the most used for the reviewed papers, and it will be studied in this section.

Once a network traffic environment is adopted, the following issue to address is how to export the packets, construct IP flows, and assign the application name (if needed) of such flows, in order to finally store or use this data. The work in

[72] presents a comprehensive procedure for flow extraction using NetFlow and IPFIX. That work defines the steps for data measurement as: i) packet capturing, ii) flow metering and exportation, and iii) data collection. The packet capturing step refers to the procedure of extracting the binary data from monitoring points, in this step each packet is considered as a single independent entity. The key aspects to be considered at this stage are the size of the sample and the sampling time [73]. Following, the flow metering process aims at aggregating the packets into flows as defined in Section II-B1. The exportation process occurs when it is considered that a flow is culminated, meaning that a communication was finished. The metering and exportation processes are related and can be united. Finally, the data collection is to store the flows exported. Having as a reference the same schema described by work in [72], only one additional component might be necessary for traffic classification with ML, which is the Label assignment. This additional component is normally placed within the traffic classification step, and aims at defining the application name per flow, or any other identifier needed to enrich the knowledge base, in particular for supervised ML techniques. The label assignment importance will be discussed in Section IV-C.

Regarding the data measurement steps, several works try to improve separately each of their deficiencies. For instance, [73] presents a taxonomy to categorize the packet sampling techniques, this work aims at giving guidelines to select the most adequate method according to the objectives to achieve. The work in [74] presents packet capture engines running on commodity hardware, which are useful for reducing the time response in traffic classification. Using commodity hardware, the works in [75], [76] show implementations for packet capture.

Traffic observation at a monitoring point starts with the packet capture. In this component, sampling and filtering procedures are involved. These packets are aggregated into flows through a metering process, to later be exported for their use. The flows collected can be stored for further analysis or directly used by the system analyzer. Some implementations of traffic monitoring are found in the literature. The work in [77] shows an implementation over industrial networks, while [78], [79] apply it over home networks. The work in [80] reports the most common implementations of popular network monitoring approaches for packet capture (Tcpdumb, Wireshark, etc), flow metering (nProbe, YAF, QoF, etc), and data collecting (nProbe, flowd, nfdumb, etc).

One of the main challenges regarding traffic observation relies on capturing packets in real time. Large volume of data at high speed is involved. In order to deal with streaming data, classically batch-based methods are deployed; however, these methods do not offer fast responses in critical environments, such as multimedia monitoring [81] or network threats. Classically, the batch-based method is deployed after the *Flow metering & exportation* block. Its main purpose is to manage and store the exported flows, usually in interval of time batches into binary files (e.g. pcap files). The batch-based method must fulfill certain requirements such as data processing speed and fault tolerance. In the case of the stream-based approach, the former requirements must be accomplished with more exigent demands, in particular, the data processing speed. For instance, the work in [82] analyzes the traffic observation process in a streaming way to reduce delays for traffic classification. The authors propose a workflow that distributes the exported flows by using a messaging system. The IP flows are transformed into a data serialization format. The selected data serialization format was the Binary JavaScript Object Notation (BSON). The final aim is to offer a distributed data system that is more efficient that a batch-based method. The work in [83] evaluated the performance of one of the most used distributed stream processing systems for traffic monitoring. The authors propose an architectural benchmark, which is publicly available for such as task. The same authors extended this work to compare three stream processing systems, in order to find the suitability of each one for real time network flow processing [83].

C. Label assignment

The label assignment procedure refers to set an identifier for each flow; this identifier is related to particular patterns in a communication session, e.g., name or type of application. Associate ground truth information with traffic traces can be a tedious task due to the complexity of tracking the flows belonging to specific applications; moreover, this procedure is a key task for validating traffic classifiers. For the best of our knowledge, few works implement a reliable ground truth assignment, due to the most common approach is to use DPI tools for such task. For instance, the work in [84] proposes a tool that is able to establish the label for each flow by using a socketing process. However, this solution is also complemented with a DPI tool.

Given the network environment and the data measurement process, the label assignment can be established by the emulation or generation systems. The emulation and generation systems are controlled environments; in this sense, it is known the type of traffic that is triggered and it can be recorded to be matched with the flows collected. Therefore, the labeling process can be performed while the monitoring process is running, or in the traffic classification system by itself. For real traffic monitoring, DPI or port-based tools are still used for this task. However, it is been demonstrated that these techniques add incertitude and error to further analysis, in particular to ML. [85] quantifies the error obtained when using port analysis and DPI tools to associate protocol labels with flows. The study reveals that port analysis is accurate only for a set of protocols; while, DPI tools provide better results. The authors demonstrated experimentally that DPI tools dramatically fails for more than 14% for P2P traffic, and for almost 100% for Skype on TCP and Streaming applications. Nonetheless, it can be found different opinions on this topic. For instance, the study made by [86] shows a performance comparison between several DPI tools, where the error between the ground truth labels and the DPI label results is low for the majority of the cases. It is important to consider that for all these comparisons, the data used plays a major impact on the results; as well as, the version of the DPI tools and the years in which the studies were performed. In general, the traffic traces show an important increase in communication patterns urging

traffic classifiers (such as DPI) to constantly upgrade their engines. Additionally, DPI tools are inaccurate in presence of encrypted traffic, this scenario makes them very inadequate for the labeling task.

Several works try to propose strategies or architectures to correctly define the ground truth labels. The work in [87] evaluates the most common approaches for measuring traffic data focused on malicious detection, and proposes a semi-manual practice to define the ground truth. Additionally, different tools have emerged to support traffic monitoring and labeling, such as Tsat [88] and Volunteer based system [89], among others. Moreover, [90] proposes a framework to large-scale network monitoring and analysis, called DBStream, which is a data repository of network monitoring data capable of processing data streams coming from a wide variety of sources. Finally, EventFlow is proposed as a new strategy to label flows, in order to retain semantic relations based on user's actions [91].

D. Discussion

In view of the most important aspects to fulfill in this section, one can remark some recommendations and highlight future works as follows.

- Real world measurements and tests on the network are difficult to achieve due to the reasons previously discussed in Section IV-A. Therefore, it is recommendable instead to set an emulated network, which may help to reproduce current application and user behavior. Additionally, it would help to serve as a testbed to set the traffic classification solutions.
- Traffic generation solutions might help creating synthetic data and reducing the amount of experimental tests for data collection. Therefore, a combination between traces captured from an emulated network, and traces from traffic generator tools might offer a great deal of network traffic data.
- The emulated network or traffic generation solutions also have to move at the same rate as the evolution of the network. Upgrading strategies need to be established for achieving this task. For instance, semi-supervised or incremental learning techniques could feed information regarding new patterns or anomalies to the data collection step.
- In order to set a traffic monitoring strategy, it is necessary to study which is the information needed from the network. The resources and time consuming will be key factors to minimize, in order to get fast and reliable data measurements from observation points. Depending on the objectives to achieve or the FE process selected, the data to be measured can be a set of packets, only their headers, or only events related to them, among others (refer to Section V).
- It is important to establish the labeling mechanism for the collected data. In specific, this is a very important procedure not only for supervised ML techniques, but also for validating unsupervised models. A reliable strategy has to be outlined, considering its goodness and deficiencies.

V. Feature extraction

This section will define some FE approaches in traffic classification with ML. Four main groups of FE procedures were established, along with some works that studied them. This study will allow us to outline discussions about each approach, as well as to guide the reader to chose the most convenient one. Three main groups were found in the reviewed papers: statistical based features, graph based features and time-series based features, which will be presented in the next sections; additionally, Section V-D covers some miscellaneous and hybrid approaches. For a more detailed study refer to the work in [92], which specifies the procedures of data construction, FE and FR from protocol, packet and flow levels. Additionally, the work in [93] proposes an automatic process to perform feature engineering. Despite of the work presented in [93] is mainly focused on detecting tunneled connections, it provides a logic work-flow for extracting and processing flows to obtain features, which can be used by ML techniques.

A. Statistical based features

The features extracted from packet flows are mainly statistical based features, which are defined under the assumption that traffic at the network layer has statistical properties (such as the distribution of the flow duration, flow idle time, packet inter-arrival time and packet lengths) that are unique for certain type of applications, and enable different source applications to be distinguished from each other. Under this assumption, the work in [94] proposes 249 statistical features, which can be extracted from flow network traffic.

Properties such as inter-arrival time (IAT) and packets length seem to be the most important characteristics considered, with their metrics such as maximum, minimum, mean and standard deviation, among others. Additionally, the amount of packets sent from a to b and vice versa, control packets, and some other properties in the packet header as the transport protocol, can be used to model communication networks.

The majority of the reviewed papers show a preference for statistical based features, as it will be noticed in Section IX. In practice, statistical features from IP flows are largely used due to its simplicity. Traffic characterization based on statistical features has been largely reported and demonstrated. For instance, for anomaly detection, the work in [92] lists all the classical statistical features for unidirectional and bidirectional flows, as well as content type. [10] presents a similar work, remarking the FE approaches from the flow and the packet level.

The work in [95] presents a complete study about statistical features. The authors tested ten classifiers to get into the conclusion that these features allow the classifier to get a good performance. On the other hand, [96] shows that with only the counts of packets and byte exchange in a session is enough to identify some applications, specifically the P2P applications.

The work in [97] studies the abnormality caused by packet retransmission in TCP connections. Packet retransmission can change the packet sequence, and in consequence, unable statistical based approaches to perform an accurate classifica-

tion. The authors proposed a system to discard retransmitted packets, detecting and using the original packets.

In general, statistical based features are commonly preferred in this field because of their computational simplicity, which is very important when dealing with high speed communications. Additionally, this approach does not intrude into the packet content, enabling its use for both non-encrypted and encrypted traffic while respecting privacy.

B. Graph based features

The intrinsic composition of the Internet network allows modeling or representing its interactions into big interconnected graphs. In consequence, this approach uses graph theory to find valuable information from the network. A network can be viewed as a set of interconnected nodes based on the assumption that the nodes are the hosts, and the edges represent the interaction between the hosts. These interactions can be viewed as communication sessions where packets are exchanged. The common procedure is to set a monitoring point, for instance a router. Groups of packets passing through the monitoring point can be aggregated into flows. The work in [98] presents the Traffic Dispersion Graph (TDG), how to build it, and how to find quantitative metrics to extract information. In a TDG, the metrics used are normally the cardinality of the graph, counting the nodes that only have incoming and outgoing edges, symmetry and connectivity of the graph, and the average degree of a node, among others. These metrics are used to train a classifier that groups the applications into collaborative or not. On the other hand, in order to compare two or more TDGs, the authors proposed to compute the relative inclusion, edge similarities and edge volatility of a graph regarding one another. These metrics are used to detect applications, such as games and DNS.

From a different point of view, the graphs can be used to represent traffic activities in the network. For instance, the work in [59] presents the Traffic Activity Graphs (TAGs) to unveil behaviors between hosts. As the previous cases, the nodes can be viewed as the hosts, and the edges the flows in opened sessions. Different variations can be found to create the links between the edges; for example, the work in [58] proposes the edges as the pair IP and port, motivated by the application behaviors that can open more than one port for P2P communications.

Statistical graph decomposition techniques are widely used by these approaches to extract the most dominant subgraphs. For instance, the nonnegative matrix factorization (NMF) or the orthogonal nonnegative matrix tri-factorization (tNMF) aims at extracting the dominant substructures and characterizing their structural properties, in order to analyze network applications. K-means is applied to find distinct application behaviors using these structural properties [60].

In summary, the main aim of this approach is to model application behaviors through graph representations, and to try to find similarities between Internet network graphs. These similarities will allow grouping graphs into pools of applications. Such similarities can be based on different features; the most common features extracted are graph structural properties, graph connectivity metrics, and community based features, such as density or similarities between well connected graphs [99], [100].

C. Time-series based features

Generally speaking, a time-series data can be viewed as a sequence of events indexed in time order. In this sense, the FE process is normally performed over a discrete-time data.

The network traffic problem has suitable characteristics to be treated as an event-driven problem. The interaction between a pair (e.g. client-server) is strongly dependent on events ordered in time, such as to open or close communication sessions, initiate or finish transmission of data, among other things. Such scenarios motivate to use time-series features or data-driven approaches to discover patterns in the network.

Most of the time-based approaches in this field try to find relationships between the inter arrival times (IATs) and packets sizes belonging to a flow, in order to characterize application patterns through time-series representations. As an example, the work in [101] proposes a time activity vector for unidirectional flows, which takes into account characteristics such as, active time of a flow, maximum and minimum amount of consecutive seconds that the flow does and does not show activity, among others. Also, it can be found different approaches that intent to use the temporal behavior of the network, such as [102].

Particularly, signal processing approaches can be used to transform the inputs in time domain to frequency domain. The main aim is to obtain the equivalent magnitude and phase of the signal data to unveil new features that can be used for traffic classification. For instance, the work in [103] presents a method to detect anomalies with a Fourier-based method, for such case the packets sent and received in a period of time are analyzed as time series inputs. Similarly, [104] addresses the FE process based on the Wavelet Leaders (WL) technique. Bidirectional flows are transformed into time series of transferred byte numbers in time windows, to later obtain multifractal features with WL. Multifractal representation refers to a spacial or time-domain statistical scaling, where its main aim is to describe the irregular or fragmented shape of features, as well as other complex objects that traditional Euclidean geometry fails to analyze.

Usually, this FE approach is not conventionally applied for traffic classification; however, these features are widely use to identify anomalies, as it is explored in [10].

D. Miscellaneous and Hybrids

The categorization below mentioned some of the approaches founded, however, there are other interesting methods to perform FE. Such approaches try to combine or to propose new features that better suit the classification problem. For instance, Bag of Flows (BoFs) derives from the statistical based approach, and it only differs in the way that the bidirectional flows are built. A BoFs comprises a set of traffic flows that are injected by the same application. A BoF can be defined as a group of flows that shares the same destination and source IP, and that presents variations in the opened

ports of a connection. This behavior is commonly seen in communication sessions. The statistical features are computed for all the flows in the BoFs. A ML model is trained with all the flows, and each bag has a label defined through some criteria; for example, the higher occurrence of a label in the bag. For a new incoming flow, the class label is obtained with the ML model; following the flow can be associated to a BoFs. Several works already tested the advantages of using BoFs to improve the performance using ML classification and clustering approaches [105], [106], [107].

On the other hand, statistical based features can be combined with graph based features to find better classification performance as it is proposed in a framework for traffic classification in [57]; and similarly in [54] for anomaly detection.

Studying the order, sequence and time in which packets are sent from a source to a destination, may help to characterize applications with correlated behaviors in time and event based approaches. As an example, the work in [108] uses only the sequence inter-packets times and payload size, to build sequences of the flows as input features of a classification model. On the other hand, [109] presents time series combined with statistical based features for their work. The time series features are obtained through the syntactic structure of the applications. The syntactic structures are represented by finite state machines. The main aim is to model the packet sequence of flows along with the packet length and packet orientation (forward and backward) by the machine states.

E. Discussion

This section presents in essence a synthesis of the reviewed FE approaches with a qualitative description and comparison, and with a brief discussion of future directions.

1) Comparison: Table I summarizes the FE trends studied in this section. In addition, their advantages and disadvantages are remarked, as well as several works that applied these approaches for traffic classification. From the table, it is noticeable that statistical based features are the most used in traffic classification. Moreover, they are suitable to deal with encrypted and unencrypted traffic.

2) Guidelines and future trends: To summarize, it is possible to find different categorizations for FE processes, and this will mainly depend on the objective to achieve. In this sense, we outline some comments regarding the FE process.

- The FE process has to be defined in accordance to the objective to achieve and the ML approach to select, and it also has to consider the computational and response time.
- One might consider to compute better descriptors that prevent misclassifications and class imbalance behaviors. For instance, in the statistical based approach, additional metrics –such as a variation of the mean and variance computation– can be taken into account (e.g. moving average [150]).
- Hybrids FE approaches can be proposed to better characterize the process. For instance, the statistical features comprise time based behaviors with the IATs statistics. Moreover, studying the time and event based behavior seems to be a logical approach to exploit in this field.

VI. Feature reduction and selection

ML models might face problems when a large number of features is given. The models are pruned to increasing computational burden, decreasing accuracy, increasing overfitting, among others. These problems are typically related to the curse of dimensionality. In general, FS is widely applied in this field to select the most relevant features and to improve the accuracy of the ML models.

ML processes might or might not count with a FR or FS process. Several studies state that a low amount of features is needed to wholly obtain patterns that differentiate an application to another. The works in [151], [152], [153] study the most relevant statistical features for traffic classification. In [152], several FS techniques are used to obtain the most important features, while a new proposed method select the smallest set. The results were crossed validated with three datasets measuring the goodness, similarity and stability of each feature; giving as a result a small set, between 6 and 14 statistical features, that offers the best performance measured through the accuracy.

The work in [152] studies the importance of FS and FR for traffic classification using ML. Ten network traffic datasets were used to show the advantages and disadvantages of different well-known FS techniques, such as Information Gain, Gain radio, Principal Component Analysis (PCA), and Correlation-based Feature Selection, among others. The authors proposed three new metrics to measure the performance of the resulting features. These metrics are based on the classifier accuracy, the stability of the results every time that a test is performed under different conditions, and the similarity between the set of features given by the FS and FR techniques. The results shows that none of the techniques provide a good performance with the new metrics; therefore, a new method that combines the FS and FR solutions is proposed. The former study is extended in [153] with an optimization process, called the Global Optimization Approach (GOA), to estimate the optimal and stable features. GOA combines multiple well-known FS techniques that yield to a possible optimal feature subsets across different traffic datasets; then the optimum entropy threshold will select the stable features.

Multi-class imbalance behavior in data is normally found in traffic classification. In this sense, the work in [143] remarks this problem when using traffic data for training ML models. The authors proposed to select a set of features for each class by using a metric based on the entropy measure. The most relevant features for class are the ones with values upper than a predefined threshold, the subset of features are sent to an ensemble classifier. Similarly, [154] proposes a new FS method that aims at obtaining the most relevant features, and at the same time at decreasing the multi-class imbalance. [155] proposes a model that can discriminate among the most relevant and irrelevant features. Such model is based on the representative deep architecture Deep Belief Networks (DBNs). One of the main characteristics of the DBN design is that from the first layer high-level features representations can be obtained, while to the latest low-level feature representations. This characteristic makes them suitable for feature

TABLE I: Summary table and related papers for the Feature Extraction process.

Approach	Description	Advantages	Disadvantages	Related works
Statistical based features	network traffic layer has statistical properties that are unique for certain type of applications, and enable different source applications to be distinguished from each other	It does not intrude into the packet content It has a lightweight computation It shows a high performance for characterizing the applications	Some of them are computed under the assumption that the properties values are normally distributed, which might not be true for some cases.	[110], [95], [96], [97], [111], [112], [113], [114], [115], [116], [117], [118], [119], [120], [121], [122], [123], [124], [125], [126], [127], [128], [129], [96], [130], [131], [132], [133], [134], [106], [135], [136], [137], [138], [105], [139], [140], [107], [141], [142], [143]
Graph based features	Internet interactions are modeled as graphs and valuable features can be extracted from these representations	They are ideal for understanding communication patterns	The cost of building the graph based representations is high. Dynamism of the Internet network demands the fast update of graph representations.	[54], [98], [60], [100]
Time-series based features	The ordered transmission of the packets can be viewed as time-series signals where different features can be extracted	They are suitable for anomaly detection	They need a wide time windows capture for extracting representative features, this can delay the response time of the classifiers.	[144], [101], [109], [104]
Miscellaneous and hybrids	It can represent a combination o variation of the previous approaches	It can take advantage of the benefits of the previous approaches	It might carry the disadvantages of the combined approaches	[145], [146], [147], [148], [102], [149], [109], [108], [57], [54]

generation as it was reported by this work.

A. Discussion

Due to the dynamism of the network, the traffic data can influence the results of ML from one dataset to another, affecting several aspects, among which the performance is one of the most important. Therefore, we can state the following discussion:

- Using the most significant features in ML impacts the run time response, improves the classifier performance and complexity for training and retraining procedures, and discards redundancy in the data. In this sense, it is recommended to always perform a previous study of the extracted features with FS or FR approaches. This process can also help treating class imbalance and discarding non significant features.
- One interesting approach to exploit is the dynamic selection of the features. In such a case, the FS process should adapt to the current condition in which the process is. Therefore, the most adequate features can be selected following a set of guidelines; for instance, the features that provide the higher entropy or information gain to the classes.
- Additionally, determining in an unsupervised manner, the features that provide with more information to the ML problem is a challenging task. This approach will lead to find a more adaptive algorithm for the dynamic selection of the features according to the current state.

VII. Algorithm Selection

It is common to find in this field different solutions using a variety of ML algorithms. Given the wide number of ML algorithms, finding the most adequate is very important in traffic classification. Particularly, most of the works have based their selection on building and testing several models until finding the one with the highest performance. In this section, selected works are detailed to outline the challenges in the moment to select a ML algorithm.

Along the review, it will be noticed that the studies are focused on particular objectives to achieve: either to identify the type of application, the protocol application, anomalies or tunneled connections. The reader will notice that the majority of the papers aim at performing the classification task (see Section VII-A). Additionally, specific works that uses multi-classification and ensemble approaches are presented in Section VII-B. Moreover, some of the papers treat the classification problem in a unsupervised manner in Section VII-C. Finally, hybrids and advanced approaches will be studied in Section VII-D.

It is important to mention that the works that treat unencrypted and encrypted traces are commonly described separately. This distinction is normally set due to the behavior of unencrypted traffic may differ from the encrypted one; this is motivated for the use of different communication protocols. However, in this section, the aim is to mainly study the traffic classification problem from the ML point of view. In consequence, an emphasis on the ML algorithm selected is given, and works that studied encrypted and unencrypted traffic will be explicitly separated in Section VII-E.

A. Classical classification

Labeled datasets are used to train supervised algorithms in order to select the best model, which in turn is obtained measuring the classification performance. For instance, for unencrypted traffic, the works in [117], [95] present a comparison between several classifiers using statistical features. The comparison was performed using the classifier accuracy and computational cost. The best performance was given by different classifiers given a variety of datasets; in general, it is possible to find classifiers that behave better than others, given certain characteristics of the dataset. This case is common among the ML algorithms due to their model structures and learning procedures are more suitable for one type of problem than another.

The work in [123] groups flows into application categories (video streaming, VoIP, etc) to improve the QoS. Several experiments were carried on varying the condition in which the data is collected (e.g., packet loss and high or low latency). Several performance metrics were computed getting as a result that the classifiers were good depending on the class (category) that is evaluated. One of the main reasons stated in the work was the class-imbalance presence; in addition, the drop outs of Internet communications that generate noise to the classifiers. In relation with the class-imbalance behavior normally found in the Internet traffic, the authors in [126] propose to compare the results of SVM classifiers trained with biased and unbiased data. Biased data contains a large number of samples belonging to only one class(es) than the rest of classes; on the

contrary, in the unbiased data all the classes have a similar number of samples. The results showed that the classes with lower amount of samples provided more than 10% of false negative, while the global accuracy did not vary significantly in the biased case. On the other hand, [156] proposes to use cost-sensitive learning for solving the multi-class imbalance problem. The idea is to compute the cost of misclassifying a flow in a class. The result lead to a confusion matrix like structure that contains the prior weighted probabilities of each class. The weight is defined as an heuristic function depending on the geometry mean of the flow rates. The cost function helps to obtain better performance when training a tree based classifier with imbalance data.

In [109], statistical features are combined with time-series features in order to construct a Hidden Markov model for traffic classification. They illustrate the syntactic structure of the chosen applications, such as Bit torrent, Skype and Emule, among others. The results showed that the proposal had higher classification accuracy than some statistic-based methods; however, the construction of the model requires labeled data and re-training when new applications appear in the network.

For anomalous or malicious traffic detection, supervised approaches and statistical based features are widely deployed, such as [116] that exposes common threats in the web to be identified, and [157] that compares several ML models for intrusion detection. The authors in [158] implemented a wireless sensor network scenario to capture traffic for specific environmental conditions. Following, a Gaussian Mixture Model is trained to detect normal and abnormal behavior in the network. On the other hand, several works try to deploy classification solutions in mobile/cellular networks. For instance, [130] collected IP traffic extracted from mobile networks in a fixed time windows. Statistical based features from normal and abnormal traffic are computed, and a Bayesian classifier is trained for the analysis of the massive network users' traffic behaviors. The work in [136] collected network traces from Wi-Fi controllers at a large university campus. These controllers connected access points to the campus backbone network, allowing the wireless devices to access Internet. The traces come from network traffic to/from malicious and benign domains, and statistical based features were computed over these traces. A binary ML classifier was trained for detecting malicious domains.

The proliferation of encrypted traffic over the network is clearly growing exponentially. Unveiling encrypted traffic attracts attention to different actors in the network, in the same manner as the non-encrypted case. Also, supervised learning is the most popular approach for encrypted traffic. In this sense, the work in [7] presents a theoretical comparison between several works that created ML models for encrypted traffic classification. In the comparison, it is noticed the difficulty of comparing the approaches due to each of them uses different and private datasets, in which the labeling process is not clearly identified for most of the datasets.

One can imagine that all the approaches deployed for unencrypted traffic should be valid for the encrypted case, particularly the works that do not intrude in the packet content during the FE process (such as the statistical based). However, it is possible to find that the new encryption methods or protocols also differ in behavior from one application to another. For instance, the work in [159] demonstrated that two encrypted applications can be classified using supervised learning. Secure Sell (SSH) and Skype traces were selected to evaluate several classifiers using statistical features (such as IAT and packets size), without considering the payload information, IP addresses and port numbers, due to the nature of the encrypted traffic. In the same context, the authors in [111] tested different ML approaches to classify VoIP encrypted traffic. They proposed a FE process using only the statistical based features. Other approaches follow the same process, keeping the statistical features as a common point, and varying the classifier or the dataset, such as in [115] with Naive Bayes and in [114] with Decision trees.

One of the particularities found in the reviewed papers is that they are normally focused on detecting a type of encryption at certain level. For instance, the work in [129] proposes a two phase method that: 1) Identify the flows under Secure Socket Layer protocol (SSL) or Transport Layer Security protocol (TLS) by using signature matching methods, and 2) Compute statistical features to classify the application within these flows. One of the deficiencies is that the signature matching method is based on standard protocol specifications and documentations, or on manual observation and analysis.

VoIP communications have risen in popularity and their identification is a key factor in the telecommunication field either to prioritize or unable them. In consequence, several approaches have been proposed, such as the works in [160], [118], [114], [161], [128]. Some of these works tries to characterize and identify Skype, one of the most complex VoIP applications in the network due to its intricate communication protocol.

Classifying applications within HTTP2 or IPsec connections is highly complex. For instance, the works in [162], [163], [164] train ML classifiers with encrypted traffic data. Several applications are launched and without a VPN and statistical based features are captured. Following, several ML models are trained and tested getting satisfactory results. On the other hand, for tunneled connections, reviewing the characteristics of a flow in Definition 2, the classical FE process cannot be applied. The tunneled connection may be identified as only one flow, when in reality there are more flows embedded. This scenario has not been largely studied, however, some works aim at identifying just a tunneled connection at first, such as in [93], [110]. An interesting method, based on ML techniques aims at detecting Denial of Service (DoS) attack in HTTP2 connections [112]. The authors prepared a network environment with normal traffic and attacked traffic. The dataset obtained is passed through a FE process where statistical features are computed, following a FS process to reduce the characteristic space. Finally, several classifiers were tested to detect the attacks in a binary classification manner.

B. Multi-classification and ensemble approaches

The combination of several classifiers might solve the generalization problem encountered by the classical classifiers

in the traffic classification field. These types of solutions aim at creating more specialized classifiers. For instance, [149] presents a combination of several ML models to get a better performance. Several classifiers (two tree based, one rule based, two statistical based and SVM) are trained with the same data; in addition, a DPI tool is included as classifier. A new incoming input is evaluated by all the classifiers, and following the results are merged by means of a combination method, such as Maximum likelihood, DempsterShafer, Enhanced DempsterShafer and perfect combination, among others. For example, in the maximum likelihood combination, all the classifiers vote on one class, and the most voted class is the output. The work in [165] compares the performance of seven ensemble algorithms based on decision trees. The study demonstrated that most of the ensemble algorithms overcome the classical single classifier approach in terms of performance. The main deficiency found in the experiments is the time cost for the training of the models and the online classification. The work in [121] proposes to compute the FE process and to divide them into subsets; this procedure is applied over three different case studies. These cases vary from the original dataset, such as with and without the zero-payload packets. Statistical features are computed for each case, and a dedicated classifier is trained. On the other hand, the work in [143] defines multiple feature subsets through a FS process; each subset of features is defined regarding a specific application protocol class. A classifier is trained for each subset, and the output is given by a voting process. In this work, it is considered that relevant features might vary from class to class, and how it can affect the classifier performance in presence of class-imbalance data.

C. Clustering for classification and anomaly detection

Unsupervised approaches are normally associated with anomalous detection, due to its capabilities to detect patterns that are not similar to normal or nominal conditions; but, also to perform classification tasks. In this sense, clustering techniques are widely used for network traffic classification. This section is dedicated to the works that apply unsupervised learning for Internet traffic classification.

One of the most common methods, to classify network traffic using unsupervised learning, is K-means, which builds K clusters based on the similarity between the samples and the centroids of the cluster. The work in [106] uses K-means and several properties of the flows for network classification. In [140], the classic random initialization of the clusters when using K-means is improved. The initial clusters are defined by the variance between the flow attributes. After the clusters are built, a mapping process cluster-application is performed with a probabilistic assignment. The maximum likelihood estimates the membership of labeled samples to the K clusters. One of the main deficiencies of this approach is that the number of clusters K must be defined beforehand, tying the solution to a fixed number of labels. Moreover, noisy samples and new label appearance are not considered.

For encrypted traffic, the work in [145] uses K-means model to identify encrypted video streaming in HTTPS connections, obtaining a good average accuracy. The dataset was extracted with different bit rates, in order to prove the classifier performance under different scenarios (change of the IATs). Similarly, the work in [166] uses K-means for P2P traffic identification. Several comparisons between unsupervised techniques are presented in [167], [168], which in turn could give insights of the approach to use when facing unlabeled encrypted traffic. Finally, due to the ground truth class is unknown, an adequate guidance must be deployed for using clustering techniques. For instance, [169] proposes a framework that allows the unsupervised approach to work for traffic classification considering the ground truth class.

D. Hybrids and advanced techniques

In this section, it will be presented some of the works that use hybrid, semi-supervised, and novel approaches for traffic classification.

To start, there are approaches that apply a two-phase process to classify network traffic, combining supervised and unsupervised learning. The first phase is in charge of clustering traffic classes with the same type (e.g. Video streaming, P2P torrent, etc), and the second phase uses supervised models to classify the applications (Youtube, Neflix, etc). The work in [137] proposes a two-phase clustering, one using the statistical flow features and the other one using the packet payload features. A third phase integrates both clustering results to create a classifier. A bag-of-words (BoWs) model is constructed to represent the content of the clusters obtained with the flow statistical features, and then a latent semantic analysis (LSA) is applied to aggregate similar traffic clusters based on their payload content. The work in [122] groups the traffic classes using K-means, and a decision tree classifies the applications in order to provide more granularity to the results. Similarly, a hybrid algorithm that combines K-means and KNN for online classification of encrypted traffic is proposed by [113]. The combination is given as a two phase process, where K-means clusters the traffic in a real time embedded environment. The performance was evaluated through a cache-based mechanism that combines elements of port-based and statistical based classification.

Also, more specialized algorithms are used to obtain more fine grained performance. As an example, [120] proposes a classification system based on the Online Sequential Extreme Learning Machine (OS-ELM) for intrusion detection. Irrelevant features are discarded using an ensemble of FS techniques. The work in [144] presents a traffic classification approach using Wavelet Kernel Extreme Machine Learning (WK-EML) and Genetic Algorithm (GA) combined to classify flows using statistical based features. GA allows optimally finding the parameters needed to use WK-EML, and in consequence, to train the model instead of using a classical random setting approach.

A novel approach to identify encrypted applications is given by [170], which presents the traffic classification problem without an explicit FE process. Instead of using the classical statistical features, the authors build a deep learning architecture that will learn from the packet content. The approach do not inspect the packet content for keywords as

DPI techniques, instead, it aims at learning new features for each application with the deep learning architecture. The FE process is embedded and these new features have not a real meaning, instead they are binary data with relationships found by the deep neural network. This last characteristic makes it suitable for encrypted traffic, and presents an alternative approach to statistical based features. However, it has not been quantified its boundaries against classical FE and classification approaches.

E. Discussion

A summary of this section is presented with a short description of the results obtained in the papers. In addition, a brief discussion of future directions is given.

1) Comparison: The tables II and III summarizes the classification approaches trends for unencrypted and encrypted traffic, respectively. Their advantages and disadvantages are remarked, as well as several works that applied these approaches for traffic classification.

2) Guidelines and future trends: Some approaches select the algorithm based on either: i) the experience of former works, ii) performance comparison with different datasets, or iii) qualitative advantages and disadvantages between the algorithms. Some specific conclusions are:

- Selecting the most adequate algorithm is highly related to the available data. Hence, it is necessary to make a preliminary study of the conditions in which the ML model works optimally, and the type of data that is able to receive. It was noticed that one or several classifiers might not generalize all the classes, and this might be caused by the class-imbalance presence in the historical data.
- One of the most common approaches found to select the ML approach is by comparing their classification performance. This comparison is normally based on the accuracy metric. However, a more accurate approach is to perform a multiple or pairwise comparison with parametric and nonparametric tests that measure the statistical significance of the classifier's performance [45].
- More flexible solutions given by ensemble classifiers might provide more accurate results with class-imbalance data. Additionally, the variability of the features for the ensemble classifiers can reinforce the solution.
- The unsupervised approaches are mainly exploited for performing anomaly detection. However, an evident challenge is the discovery of novel classes that might be represented by new clusters constructions.
- Finally, one interesting approach that might support all the deficiencies –related to imbalanced-class data, new application discovery and generalization– is proposing meta-learning processes for the selection and construction of the ML solutions.

VIII. Model deployment

The main objective of this section is to inspect the implementation and reconfiguration attempts of the ML solutions in real network scenarios. This study was done only taking as reference the papers reviewed.

Along the papers, several FE, FS and ML approaches were studied, which in turn all together comprise the necessary steps to classify traffic. The main question that arise is how the ML solutions can be deployed into real scenarios. In most of the paper reviewed, the implementation of ML solution is not performed and they normally show a proof of concept. However, some of the works above give some hints about how the ML approach can be deployed. In addition, it is analyzed the importance of the ML solution reconfiguration; which plays an important role in the traffic classification task.

A. On-line implementation

One of the most important features, that the Internet network has, is that transmission rates are normally very high and the dimension of the network is big. These main characteristics make the classifier implementation efforts challenging. The most common approach is to deploy the ML solutions over the traffic monitoring tools (see Section IV-B); hence, each time that a packet flow is observed, it is possible to perform the classification. For instance, [180] uses a NetFlow enabled router for monitoring the traces, which are forwarded in an online manner to a ML classifier. NetFlow is a Cisco protocol that aims at exporting IP flow information from routers and switches. Similarly, DBSstream [90] integrates the traffic classification solutions into its monitoring platform. Another approach is to implement the ML solution in a standalone classification module; for example, the work in [133] implements the ML solution into a Field-Programmable Gate Array (FPGA) based embedded system. The FPGA device uses information at the network layer, such as the packet sizes and IATs.

However, the monitoring module may or may not contain the classification model directly; the captured information can be sent to a server or traffic controller where the operations needed to classify the traffic are performed [139], [158]. The work in [110] proposes an implementation using a simple proxy server. Several hosts are connected to this server, and VoIP and other traffic are injected into it. Particularly, a VoIP detection algorithm is placed at the proxy in order to give priority to this traffic. The work in [113] implemented their ML solution into a Service Control Engine (SCE), which is a Cisco platform designed for session-based classification and control of all network traffic. [127] presents a QoS-aware traffic classification framework implemented in a network controller. This controller is placed in a Software-Define Networking (SDN) technology that allows performing monitoring and traffic classification.

B. Reconfiguration

One of the main issues, founded in most of the ML solutions studied so far, is that when new patterns are appearing in the network, the ML models must be updated. The ML based classification is pruned to rapidly be out-of-date due to the dynamism of the network. Therefore, self-learning, evolving or retraining strategies must be taken into account. In general,

TABLE II: Summary table and related papers using unencrypted traffic for the Algorithm selection trends.

Approach	Description	Advantages	Disadvantages	Related works
Classical classification	Labeled datasets are used to train supervised algorithms.	Simple solutions for knowing the status of the network	Retraining of the solution must be continually performed The performance can be affected by inner class imbalance	Several classical classifiers [123], [95], [149], [70], [116], [157], Decision trees [171], [54], Support vector machines (SVMs) [126], [104], [96], Expectation Maximization [141], Laplacian SVM [127], Bayesian Networks (BNs), Decision Trees (DTs) and Multilayer Perceptrons (MLPs) [117],K Nearest Neighbour (KNN) [137], [119], Naive Bayes [138], [105], [142], [130], [172]. Hidden Markov Model(HMM) [144], [173], [93], Gaussian model [147], [158]
Multi-classification and ensemble approach	It aims at creating a combination of several classifiers	The combination of several classifiers might solve the generalization problem encountered by the classical classifiers. More specialized classifiers improve the performance	In some cases, the complexity of these solutions is higher than the classical ones	AdaBoost and decision trees [139], Random forest [136], Several ML models [149], [143], [121], Several decision trees [165]
Clustering	Unlabeled datasets are used to cluster "similar" behaviors	Unsupervised approaches are normally associated with anomalous detection, due to its capabilities to detect patterns that are not similar to normal or nominal conditions	The performance is low compared to supervised approaches	K-means [106], [140], [137], Hierarchical clustering [174]. Quantile-based clustering [175]. Graph based clustering [100], Fuzzy Gustafson-Kessel clustering [101], Tree-based clustering algorithm [134], KNN and Centroid based clustering [176]
Hybrids and advanced techniques	It combines the previous approaches. It presents novel approaches to do traffic classification	More fine classification for specific tasks	The cost might be high compared to the previous approaches	k-means and decision trees[122], [177], [120], Artificial Immune System (AIS) algorithm [178], Genetic Algorithm [124], Online Sequential Extreme Learning Machine (OS-ELM) [120], Wavelet Kernel Extreme Machine Learning (WK-EML) and Genetic Algorithm [109]. Convolutional Neural Networks [170]

TABLE III: Summary table and related papers using encrypted traffic for the Algorithm selection trends.

Approach	Description	Advantages	Disadvantages	Related works
Classical classification	The same procedures applied for unencrypted data can be reused	It helps to identify behaviors of encrypted protocols	Labeling of the traces is a difficult task. Retraining and imbalance behaviors are also present	Several classical classifiers [111], [7], [160], [118], [114], [161], [128], [159], [167], [168], [7], [164]. Naive Bayes [115], [166]. Decision tree [114]. Naïve Bayes (NB), Decision Tree (DT), JRip, Support Vector Machines (SVMs) [112], Markov chains [141]
Multi-classification	-	-	-	AdaBoostm C5.0 and genetic programming based approach [128], Bagging-based model [162]
Clustering	-	-	-	k-means [145], [166], Several classical clustering methods [167], [168]
Hybrids and advanced techniques	-	-	-	Convolutional Neural Network [170], [179], KNN and K-means [113]

Note: the symbol "-" indicates that the information given in Table II is equivalent in this table

for most of the ML techniques an update implies a model retraining with a new historical dataset (commonly labeled). This means that the model has to be constantly retrained any time that a new application or behavior appears on the network. The cost of performing this step can be significant; nonetheless, if this point is not considered then the model performance is in risk.

On this subject, [181] presents a Self-Learning Traffic Classifier (SLTC) for P2P identification. The authors propose an architecture where passive monitoring components observe the network at specific links. The distributed monitoring components have embedded the classifier (based on payload inspection). If this component can correctly classify the flow, then this will be marked as known, otherwise, the flows are sent to a logic server. The logic server is in charge of more fine operations to identify the traffic when the monitoring component fails. Additionally, the logic server forward policies rules to feed the monitoring component with new behaviors. It is important to mention that the operations performed in the logic server are based on a statistical based approach, payload inspection and application analysis; however, this proposal can be modified replacing their solution by a ML solution, defining retraining or updating policies for online ML based classifiers. Comparably, the work in [107] proposes a Self-Learning Intelligent Classifier (SLIC) for traffic classification. SLIC requires only a small number of labeled flows for its learning process, the system is able to evolve to a new configuration with unlabeled data. SLIC uses BoFs structure to classify a set of flows into the same group using KNN. First at all, a KNN model is trained with the labeled samples. After the training is performed, unlabeled samples can be marked by the system as possible new training samples, and they are saved in a batch. When the batch reaches a defined threshold, a retraining process is activated inducing the new training samples to the KNN model. The decision to induce new training samples or not resides in the prediction step, where two conditions, based on the distance between the samples and their two nearest classes, will allow taking the final decision. Although, the system presents a good performance in term of traffic classification with two dataset, it might present problems with imbalance or high dimensional data where distance based methods tend to fail. Additionally, the computational cost of this approach should be quantified.

In the ML field, semi-supervised learning may help to deal with the evolving feature of the network, and with the lack of labeled data. A semi-supervised work-flow may deal with the updating of the ML models in a online manner. For instance, [141] proposed a semi-supervised approach for traffic clustering. As a first step, a Gaussian Mixture Model (GMM) with set-based equivalence constraint is used, and

following the clustering algorithm based on the Expectation Maximization (EM) method. The semi-supervised process is handled with side information, which aims at building a set of flows given some constraints with a Gaussian model. The side information principle is based on the BoFs definition. Finally, classical statistical features of the flows in the sets are used to cluster the application protocol with EM. [135] also addressed the traffic classification problem using a semi-supervised approach. Support vectors machines (SVMs) are built using the co-training approach, in which multiple classifiers are trained on different sets of features. This approach is used when the historical dataset counts with both labeled and unlabeled data; where taking into account the unlabeled data can also provide more knowledge to the model. The work in [131] presents a similar approach. The work in [146] analyses time based and host based features for intrusion detection in the network. The proposal is based on a semi-supervised approach that trains a classifier with labeled samples. A membership vector is obtained for the unlabeled data with the classifier. The membership vector is used to obtain fuzzy groups that are further incorporated into the training set for a re-training of the classifier.

C. Traffic Classification in Cellular, WiFi and Satellite networks

In this section, we make an overview of the application/implementation of traffic classification in three important network infrastructures. In this sense, we perform a short analysis about the IP traffic classification with ML in these domains.

In cellular networks, the mobile IP traffic classification can be performed at different levels either using the port, the packet payload [182], [183] or the flow statistical distribution. On this subject, ML learning is applied when statistical attributes are available from cellular network traces. For instance, [130] collected IP traffic extracted from mobile networks in a fixed time windows. Statistical based features from normal and abnormal traffic are computed, and a Bayesian classifier is trained for the analysis of the massive network users' traffic behaviors. The work in [184] presents an approach to correctly collect and label mobile IP network traces, while in [24] a taxonomy to define the labels is proposed in the same domain. The work in [185] presents a fine grain process to correctly extract the IP flows from mobile networks in view of the ML solution implementation. More on this subject, it can be found in [186], [187], where dedicated surveys are presented. For instance, the work in [186] exposed a generic architecture of a cellular network, and the possible positions where passive monitoring can be deployed, such as in a Packet Switched (PS) Core. To this end, IP-based data in passive monitoring points can be analyzed by ML classifiers following the complete process exposed along this survey paper.

In a similar manner, in WiFi networks, IP-data can be extracted in order to apply ML approaches for the traffic classification. For instance, the work in [136] collected network traces from Wi-Fi controllers at a large university campus. These controllers connected access points to the campus backbone network, allowing the wireless devices to access Internet. The traces come from network traffic to/from malicious and benign domains, and statistical based features were computed over these traces. A binary ML classifier was trained for detecting malicious domains. Similar approaches can be found in [188], [158], [187]. The difference with cellular/mobile networks and WiFi network resides in the technology used for the data exchange that might affect the speed, cost and security.

Finally, in satellite networks, the traffic management is a key task due to it allows improving the QoS. Commonly, traffic data is captured form satellite Internet Service Providers (ISPs). The works in this area aim to classify and to analyze Internet traffic in large networks [189], [190], [191], [192]. The principle is the same as the previous cases, passive monitoring is deployed in order to perform traffic classification. These monitoring points can be at routers [189], [190] or point of presence (PoP) [191] of large ISP networks. Another emerging approach is the use of Software-defined networks(SDNs) in satellite-terrestrial networks. In SDNs, traffic classification can be easily deployed in the SDN' master controllers as it is exposed in [193], [194].

To conclude, this section presented possible applications of ML-based traffic classification in three related Internet network infrastructures, where the methodological steps presented by this survey paper can be used.

D. Discussion

To conclude, it is presented as follows some important challenges and comments regarding the content of this section.

- A fast response of the ML models and the monitoring process are required for traffic classification solutions. Measuring the effective time that the complete classification process will take, is critical in this field. Normally, the Internet communications take place in milliseconds to open and close sessions, and to transfer data. The traffic identification has to be faster, in order to take actions and to validate results.
- Commonly, the ML solutions are validated when the training of the model is executed, and it is based on the validation techniques reviewed in Section II-A5. In order to measure the performance of the ML solution in an online context, it is necessary to establish a framework that provides the ground truth applications. In this sense, metrics such as the accuracy and the F-score can be computed for ensuring the reliability of the traffic classification.
- An important challenge, when using and implementing ML in the networking field, is the scalability of the solution due the size of the Internet network.
- Finally, ML solutions that allow considering the dynamism of the network must be deployed. The development of autonomic architectures [195], [196], [197], allowing context adaptation (behavioral or structural adaptation), can be applied by using ML models.

IX. Analysis

This study has organized its structure in a manner that remarks some important procedures and challenges for achiev-

ing knowledge extraction with ML techniques. The proposed procedures involve data collection, feature engineering (FE, FR and FS), algorithm selection and model deployment. The question that arises is: which is the best path to take or how to start looking for related references? 49 papers were selected as they present the complete procedure in Fig. XI-A using a variety of strategies for traffic classification, compelling to show the challenges remarked in the previous sections. The results of the present review are summarized in Figure 2. A path is drawn in order to know the procedural trends commonly taken. The most important remarks concerning Figure 5 are listed as follows.

A. Data collection phase

This block in Fig. 5 is related to the characteristics observed in Section IV. First at all, the papers are divided into two: the papers that used *Real* and *Emulated* scenarios for traffic observation. Traffic generation is not considered due to is not a common method used for traffic classification with ML. It is noticeable that the amount of papers in both cases is similar. Following, it is studied if the data is publicly available or not. The patterns remain the same as the previous case; however, most of the papers with real traffic have used well-known datasets, some listed in Appendix XI-A.

Regarding encryption, it was found that most of the approaches start with a non-encrypted study before proposing a mapping to the encrypted case. The review is mainly conformed of 60% papers with non-encrypted data, and the remaining for encrypted studies. Continuing, the ground truth is normally defined by DPI tools, where few papers defined a strict process for labeling the flows. The general conclusion of this block is that the tendency is to use real traffic measurement for well-known public data, but also to set network architectures to emulate traffic. Most the traffic is unencrypted and the ground truth of the flows is not commonly available.

B. Feature engineering phase

In this block, the sections V and VI are comprised. The first aspect considered is the type of FE performed, while the second one reports if the FS process is performed or not. It is noticeable from the Fig. 5 that the most common trend is the FE process by using statistical based (*STATsB*) features. This result is expected due to these features can be interchangeable between encrypted and non-encrypted traffic analysis, besides they do not intrude into the packet content, and their computation is lightweight.

C. Algorithm selection

In algorithm selection, the classical classification (*CClass*) is dominant, which is an expected outcome because this study is focused on traffic classification. However, some approaches have made some efforts in the multi-classification and ensemble approaches (*MClass&E*). This last one motivated from the class-imbalance and generalization problems. Also, it can be found some efforts to use clustering (*Clust*) and hybrid or advances (*H&A*) techniques for traffic classification. In terms of the objective to achieve, the application category (ApPC) and protocol (AppN) are commonly searched, but anomaly detection (AD) is also an interest. Others objectives are to detect the behavior of the users, find communities, and create user profiles, among others.

D. Model deployment

Finally, in the model deployment block, the interest is to show the amount of papers that implements the solution. In addition, it is important to know the reconfiguration strategies proposed by the papers selected. This last one is important due to the constant dynamism of the network that causes problems to some ML approaches.

From Fig. 5, it is noticeable that more than 65 % do not specified (*NNS*) the implementation of the solution. In the same manner, most of the solutions do not offer reconfiguration methods (*ONS*); although, some of the works propose scheduled retraining processes to upgrade their solution. Self-learning or evolving (*SLE*) is not found in the reviewed papers; however, this is a challenging topic for the ML techniques in general.

E. Future trends

Finally, to conclude the analysis, it is remarked in red the approaches less taken by the reviewed papers. It is noticeable from Fig 5, that most of the works studied did not use encrypted data, and the ground truth establishment was not commonly performed. In the feature engineering process, few works applied FS for their solution. In terms of the ML approach used, the less explored are the unsupervised techniques, the multi-classification and ensemble approaches. Finally, for the model deployment phase, there are few works that implemented the solution in real world scenarios. Moreover, the reconfiguration processes do not handle evolving or autonomic reconfigurations. Given these scenarios and the discussions of the previous section, the following future trends are concluded.

- The proposition of emulated traffic architectures, oriented to create ML solutions, can allow the generation of encrypted and unencrypted traffic, as well as a reliable flow labeling process.
- Given the efficiency of the statistical based approach, an exploration to improve its computation will provide robust classification solutions. In the case of the FS, an efficient dynamic selection of the features might offer a higher classification performance.
- In the algorithm selection block, the future trends to exploit are the deployment of multi-classification and ensemble approaches, which are very promising path given the characteristics of the Internet data.
- For the ML solutions, more experimental tests are needed to measure the performance in term of response time and complexity. In this particular case, it is necessary to propose distributed and scalable ML solutions that can deal with the dimension of the Internet network.
- In terms of the Model reconfiguration, an alternative solution will be to deploy the concept of incremental

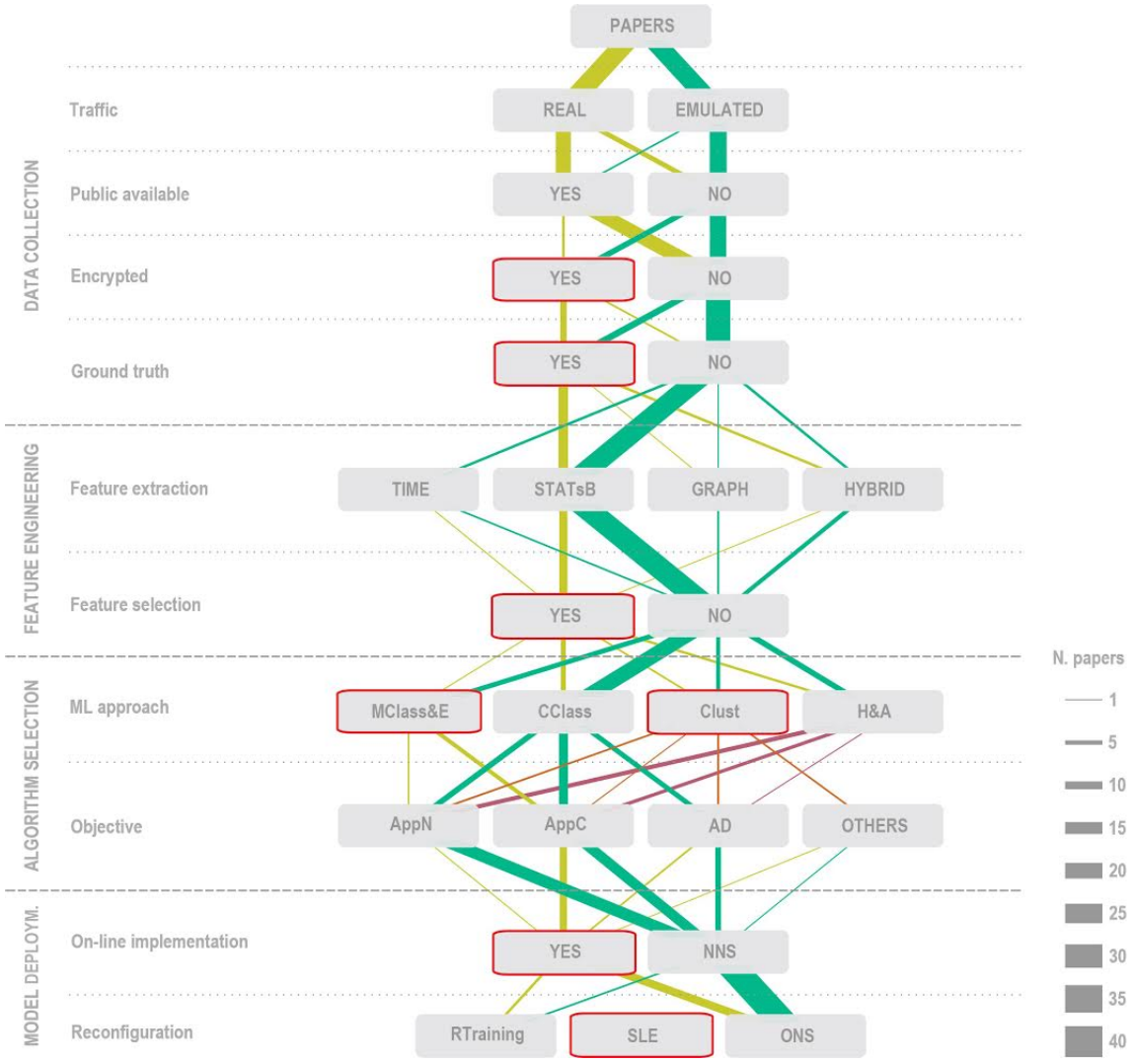

Fig. 5: Trends of the selected papers for classifying traffic with ML.

learning, in order to deal with the dynamism of the Internet network.

X. Conclusion

This survey paper presented the general procedure to achieve traffic classification through ML techniques. Each step of the procedure has some relevant works that follow different paths to achieve the results. In this way, the reviewed papers were organized, where each step defines the category to which each paper belongs. These categories were displayed into a graphic that illustrates the most common trends in the field. Discussions were settled in each section, in order to identify its most common trends and challenges.

This study tried to unveil several challenges in the traffic classification field. For instance, working with encrypted traffic is challenging, where the type of features should lead to correct classifications in absence of the packet content. The current publicly labeled data is scarce, which make it hard for comparing ML solutions. Following, for knowledge extraction with ML, the classification task was found as the most popular. However, multi-classification and ensemble approaches present some advantages that make them compelling for dealing with some problems in traffic classification, such as class-imbalance and generalization. The clustering approach can help to find new or anomalous behavior in the Internet traffic; therefore, its study in this field should be extended. Finally, the implementation of such solutions remains as an important task to achieve, due to different factors, mostly related to performance and adaptability of the solutions.

The difference of the present review, regarding most of the surveys found in the literature, is that it shows the whole picture of the steps needed for network traffic classification. Moreover, it analyses the challenges at each stage of the process and outlines future directions. To summarize, some of the most important directions that this works want to emphasize are:

- Reliable label assignment, this procedure will play a key role for the construction and validation of the ML models
- Dynamic feature selection, this will try to create adaptive models that use the most suitable features given the context and the objective to achieve
- Integration of meta-learning processes for dealing with

the imbalance and the dynamism of the Internet network data
- Strategies for the online reconfiguration of the ML solutions.

ACKNOWLEDGMENT

This work is sponsored by Thales Alenia Space, TOULOUSE, 31100, France. We want to thank the Département : Business Line Telecommunication, R&D department, for their assistance.

REFERENCES

[1] "Internet Assigned Numbers Authority (IANA)," https://www.iana.org/, accessed: 2017-09-27.
[2] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.
[3] A. Dainotti, A. Pescape, and K. C. Claffy, "Issues and future directions in traffic classification," *IEEE Network*, vol. 26, no. 1, pp. 35–40, 2012.
[4] N. A. Khater and R. E. Overill, "Network traffic classification techniques and challenges," in *2015 Tenth International Conference on Digital Information Management (ICDIM)*, 2015, pp. 43–48.
[5] P. Foremski, "On different ways to classify internet traffic: a short review of selected publications," *Theoretical and Applied Informatics*, vol. 25, no. 2, 2013.
[6] N. Namdev, S. Agrawal, and S. Silkari, "Recent advancement in machine learning based internet traffic classification," *Procedia Computer Science*, vol. 60, pp. 784 – 791, 2015, knowledge-Based and Intelligent Information & Engineering Systems 19th Annual Conference, KES-2015, Singapore, September 2015 Proceedings.
[7] P. Velan, M. Cermák, P. Celeda, and M. Drasar, "A survey of methods for encrypted traffic classification and analysis," *International Journal of Network Management*, vol. 25, no. 5, pp. 355–374, 2015.
[8] H. Alizadeh and A. Zúquete, "Traffic classification for managing applications networking profiles," *Security and Communication Networks*, vol. 9, no. 14, pp. 2557–2575, 2016.
[9] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.
[10] A. Marnerides, A. Schaeffer-Filho, and A. Mauthe, "Traffic anomaly diagnosis in internet backbone networks: A survey," *Computer Networks*, vol. 73, pp. 224 – 243, 2014.
[11] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.
[12] M. F. Umer, M. Sher, and Y. Bi, "Flow-based intrusion detection: Techniques and challenges," *Computers & Security*, vol. 70, pp. 238 – 254, 2017.
[13] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "Advances in knowledge discovery and data mining," U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1996, ch. From Data Mining to Knowledge Discovery: An Overview, pp. 1–34.
[14] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems with Applications*, vol. 73, no. Supplement C, pp. 220 – 239, 2017.
[15] H. Liu and H. Motoda, *Computational Methods of Feature Selection*. Boca Raton, Florida: Chapman and Hall, Taylor and Francis Group, 2008.
[16] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.
[17] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, and Z. Wang, "A novel feature selection algorithm for text categorization," *Expert Systems with Applications*, vol. 33, no. 1, pp. 1 – 5, 2007.
[18] L. E. Raileanu and K. Stoffel, "Theoretical comparison between the gini index and information gain criteria," *Annals of Mathematics and Artificial Intelligence*, vol. 41, no. 1, pp. 77–93, 2004.
[19] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Advances in neural information processing systems*, 2005, pp. 507–514.
[20] M. Liu and D. Zhang, "Feature selection with effective distance," *Neurocomputing*, vol. 215, pp. 100 – 109, 2016.
[21] S. Wang, W. Pedrycz, Q. Zhu, and W. Zhu, "Unsupervised feature selection via maximum projection and minimum redundancy," *Knowledge-Based Systems*, vol. 75, pp. 19 – 29, 2015.
[22] N. Zhou, H. Cheng, W. Pedrycz, Y. Zhang, and H. Liu, "Discriminative sparse subspace learning and its application to unsupervised feature selection," {*ISA*} *Transactions*, vol. 61, pp. 104 – 118, 2016.
[23] D. Wang, H. Zhang, R. Liu, X. Liu, and J. Wang, "Unsupervised feature selection through Gram-Schmidt orthogonalization-a word co-occurrence perspective," *Neurocomputing*, vol. 173, Part 3, pp. 845 – 854, 2016.
[24] Z. Liu, R. Wang, and D. Tang, "Research on mobile network traffic taxonomy," in *2016 International Conference on Computer, Information and Telecommunication Systems (CITS)*, 2016, pp. 1–5.
[25] C.-H. Chen, "Feature selection for clustering using instance-based learning by exploring the nearest and farthest neighbors," *Information Sciences*, vol. 318, pp. 14 – 27, 2015.
[26] S. Maldonado, E. Carrizosa, and R. Weber, "Kernel penalized k-means: A feature selection method based on kernel k-means," *Information Sciences*, vol. 322, pp. 150 – 160, 2015.
[27] Y. Wu, C. Wang, J. Bu, and C. Chen, "Group sparse feature selection on local learning based clustering," *Neurocomputing*, vol. 171, pp. 1118 – 1130, 2016.
[28] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognition Letters*, vol. 10, no. 5, pp. 335 – 347, 1989.
[29] A. Klepaczko and A. Materka, *Artifical Intelligence and Soft Computing: 10th International Conference, ICAISC 2010, Zakopane, Poland, June 13-17, 2010, Part II*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, ch. Combining Evolutionary and Sequential Search Strategies for Unsupervised Feature Selection, pp. 149–156.
[30] A. Y. Ng, "Feature selection, L1 vs. L2 regularization, and rotational invariance," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 78.
[31] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Variable selection using random forests," *Pattern Recognition Letters*, vol. 31, no. 14, pp. 2225–2236, 2010.
[32] G. Katz, E. C. R. Shin, and D. Song, "Explorekit: Automatic feature generation and selection," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, Dec 2016, pp. 979–984.
[33] Y. Kim, H. Lee, and E. M. Provost, "Deep learning for robust feature generation in audiovisual emotion recognition," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 3687–3691.
[34] M. H. Dunham, *Data Mining: Introductory and Advanced Topics*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2002.
[35] I. Witten and E. Frank, *Data mining: practical machine learning, tools and techniques*. Boston: Morgan Kaufman, 2005.
[36] T. Hastie, R. Tibshirani, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction, 2nd Edition*, ser. Springer series in statistics. Springer, 2009.
[37] M. Pimentel, D. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215 – 249, 2014.
[38] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.
[39] V. Losing, B. Hammer, and H. Wersing, "Incremental on-line learning: A review and comparison of state of the art algorithms," *Neurocomputing*, vol. 275, no. Supplement C, pp. 1261 – 1274, 2018.
[40] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.
[41] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427 – 437, 2009.
[42] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861 – 874, 2006.
[43] D. M. W. Powers, "Evaluation : from precision , recall and f-measure to roc," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37 – 63, 2006.
[44] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.
[45] F. Pacheco, J. V. de Oliveira, R.-V. Snchez, M. Cerrada, D. Cabrera, C. Li, G. Zurita, and M. Arts, "A statistical comparison of neuroclassifiers and feature selection methods for gearbox fault diagnosis under realistic conditions," *Neurocomputing*, vol. 194, no. Supplement C, pp. 192 – 206, 2016.

[46] J. Khalife, A. Hajjar, and J. Diaz-Verdejo, "A multilevel taxonomy and requirements for an optimal traffic classification model," *International Journal of Network Management*, vol. 24, pp. 101 – 120, 2014.

[47] B. Claise, B. Trammell, and P. Aitken, "Specification of the IP flow information export (IPFIX) protocol for the exchange of flow information," Internet Engineering Task Force (IETF), Tech. Rep. 7011, 2013.

[48] R. Antonello, S. Fernandes, C. Kamienski, D. Sadok, J. Kelner, I. Gdor, G. Szab, and T. Westholm, "Deep packet inspection tools and techniques in commodity platforms: Challenges and trends," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1863 – 1878, 2012.

[49] T. Bujlow, V. C.-E. nol, and P. Barlet-Ros, "Independent comparison of popular DPI tools for traffic classification," *Computer Networks*, vol. 76, pp. 75 – 89, 2015.

[50] C. Xu, S. Chen, J. Su, S. M. Yiu, and L. C. K. Hui, "A survey on regular expression matching for deep packet inspection: Applications, algorithms, and hardware platforms," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2991–3029, 2016.

[51] A. Tongaonkar, R. Torres, M. Iliofotou, and Ram, "Towards self adaptive network traffic classification," *Computer Communications*, vol. 56, pp. 35 – 46, 2015.

[52] C. MU, X. hong HUANG, X. TIAN, Y. MA, and J. li Qi, "Automatic traffic signature extraction based on fixed bit offset algorithm for traffic classification," *The Journal of China Universities of Posts and Telecommunications*, vol. 18, pp. 79 – 85, 2011.

[53] C. MU, X. TIAN, X. hong HUANG, and Y. MA, "FlowAntEater: network traffic automatic signature generator," *The Journal of China Universities of Posts and Telecommunications*, vol. 20, pp. 69 – 74, 2013.

[54] E. Bocchi, L. Grimaudo, M. Mellia, E. Baralis, S. Saha, S. Miskovic, G. Modelo-Howard, and S.-J. Lee, "MAGMA network behavior classifier for malware traffic," *Computer Networks*, vol. 109, Part 2, pp. 142 – 156, 2016.

[55] D. Muelas, M. Gordo, J. L. Garca-Dorado, and J. E. L. de Vergara, "Dictyogram: A statistical approach for the definition and visualization of network flow categories," in *2015 11th International Conference on Network and Service Management (CNSM)*, 2015, pp. 219–227.

[56] G. Levchuk, "Function and activity classification in network traffic data: existing methods, their weaknesses, and a path forward," vol. 9850, 2016, pp. 9850 – 9850 – 13.

[57] M. Iliofotou, H. chul Kim, M. Faloutsos, M. Mitzenmacher, P. Pappu, and G. Varghese, "Graption: A graph-based P2P traffic classification framework for the internet backbone," *Computer Networks*, vol. 55, no. 8, pp. 1909 – 1920, 2011.

[58] J. Jusko and M. Rehak, "Identifying peer-to-peer communities in the network by connection graph analysis," *International Journal of Network Management*, vol. 24, no. 4, pp. 235–252, 2014.

[59] Y. Jin, E. Sharafuddin, and Z.-L. Zhang, "Unveiling core network-wide communication patterns through application traffic activity graph decomposition," *SIGMETRICS Perform. Eval. Rev.*, vol. 37, no. 1, pp. 49–60, Jun. 2009.

[60] K. Xu, F. Wang, and L. Gu, "Behavior analysis of internet traffic via bipartite graphs and one-mode projections," *IEEE/ACM Transactions on Networking*, vol. 22, no. 3, pp. 931–942, June 2014.

[61] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 626–688, May 2015.

[62] N. Secchi, R. Bonelli, S. Giordano, and G. Procissi, "BRUTE: a high performance and extensible traffic generator," in *Proceedings of the SPECTS 05*, 2005, p. 839845.

[63] "iPerf - the ultimate speed test tool for TCP, UDP and SCTP," https://iperf.fr/, accessed: 2017-07-18.

[64] "OSTINATO network traffic generator and analizer," http://ostinato.org/, accessed: 2017-07-18.

[65] C. V. Wright, C. Connelly, T. Braje, J. C. Rabek, L. M. Rossey, and R. K. Cunningham, "Generating client workloads and high-fidelity network traffic for controllable, repeatable experiments in computer security," in *Recent Advances in Intrusion Detection*, S. Jha, R. Sommer, and C. Kreibich, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 218–237.

[66] A. Dainotti, A. Pescap, P. S. Rossi, F. Palmieri, and G. Ventre, "Internet traffic modeling by means of hidden markov models," *Computer Networks*, vol. 52, no. 14, pp. 2645 – 2662, 2008.

[67] A. Botta, A. Dainotti, and A. Pescap, "A tool for the generation of realistic network workload for emerging networking scenarios," *Computer Networks*, vol. 56, no. 15, pp. 3531 – 3547, 2012.

[68] P. Megyesi, G. Szabó, and S. Molnár, "User behavior based traffic emulator: A framework for generating test data for DPI tools," *Computer Networks*, vol. 92, pp. 41 – 54, 2015.

[69] S. Molnar, P. Megyesi, and G. Szabo, "Multi-functional traffic generation framework based on accurate user behavior emulation," in *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2013, pp. 13–14.

[70] L. Vassio, I. Drago, and M. Mellia, "Detecting user actions from HTTP traces: Toward an automatic approach," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Sept 2016, pp. 50–55.

[71] "Openbach," https://www.openbach.org/, accessed: 2017-12-18.

[72] R. Hofstede, P. Celeda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras, "Flow monitoring explained: From packet capture to data analysis with netflow and IPFIX," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 2037–2064, 2014.

[73] S. J. M. C., P. Carvalho, and L. Solange Rito, "Inside packet sampling techniques: exploring modularity to enhance network measurements," *International Journal of Communication Systems*, vol. 30, no. 6, pp. e3135–n/a, 2017.

[74] V. Moreno, J. Ramos, P. M. S. del Ro, J. L. Garca-Dorado, F. J. Gomez-Arribas, and J. Aracil, "Commodity packet capture engines: Tutorial, cookbook and applicability," *IEEE Communications Surveys Tutorials*, vol. 17, no. 3, pp. 1364–1390, 2015.

[75] P. Velan and V. Pus, "High-density network flow monitoring," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2015, pp. 996–1001.

[76] T. Wellem, Y. K. Lai, C. Y. Huang, and W. Y. Chung, "A hardware-accelerated infrastructure for flexible sketch-based network traffic monitoring," in *2016 IEEE 17th International Conference on High Performance Switching and Routing (HPSR)*, June 2016, pp. 162–167.

[77] A. Neumann, M. Ehrlich, L. Wisniewski, and J. Jasperneite, "Towards monitoring of hybrid industrial networks," in *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, May 2017, pp. 1–4.

[78] Z. Aouini, A. Kortebi, and Y. Ghamri-Doudane, "Traffic monitoring in home networks: Enhancing diagnosis and performance tracking," in *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Aug 2015, pp. 545–550.

[79] A. Kortebi, Z. Aouini, M. Juren, and J. Pazdera, "Home networks traffic monitoring case study: Anomaly detection," in *2016 Global Information Infrastructure and Networking Symposium (GIIS)*, Oct 2016, pp. 1–6.

[80] I. Ghafir, V. Prenosil, J. Svoboda, and M. Hammoudeh, "A survey on network security monitoring systems," in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, Aug 2016, pp. 77–82.

[81] W. Robitza, A. Ahmad, P. A. Kara, L. Atzori, M. G. Martini, A. Raake, and L. Sun, "Challenges of future multimedia QoE monitoring for internet service providers," *Multimedia Tools and Applications*, 2017.

[82] T. Jirsik, M. Cermak, D. Tovarnak, and P. Celeda, "Toward stream-based IP flow analysis," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 70–76, 2017.

[83] M. Cermák, T. Jirsk, and M. Latovika, "Real-time analysis of NetFlow data for generating network traffic statistics using Apache Spark," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, April 2016, pp. 1019–1020.

[84] F. Gringoli, L. Salgarelli, M. Dusi, N. Cascarano, F. Risso, and k. c. claffy, "Gt: Picking up the truth from the ground for internet traffic," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 5, pp. 12–18, Oct. 2009.

[85] 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.

[86] V. Carela-Español, T. Bujlow, and P. Barlet-Ros, "Is our ground-truth for traffic classification reliable?" in *Passive and Active Measurement*, M. Faloutsos and A. Kuzmanovic, Eds. Cham: Springer International Publishing, 2014, pp. 98–108.

[87] M. Stevanovic, J. M. Pedersen, A. D'Alconzo, S. Ruehrup, and A. Berger, "On the ground truth problem of malicious DNS traffic analysis," *Computers & Security*, vol. 55, pp. 142 – 158, 2015.

[88] A. Finamore, M. Mellia, M. Meo, M. M. Munafo, P. D. Torino, and D. Rossi, "Experiences of internet traffic monitoring with tstat," *IEEE Network*, vol. 25, no. 3, pp. 8–14, 2011.

[89] T. Bujlow, K. Balachandran, T. Riaz, and J. M. Pedersen, "Volunteer-based system for classification of traffic in computer networks," in *2011*

19th Telecommunications Forum (TELFOR) Proceedings of Papers, Nov 2011, pp. 210–213.

[90] A. Baer, P. Casas, A. DAlconzo, P. Fiadino, L. Golab, M. Mellia, and E. Schikuta, "DBStream: A holistic approach to large-scale network traffic monitoring and analysis," *Computer Networks*, vol. 107, pp. 5 – 19, 2016.

[91] P. Velan, "Eventflow: Network flow aggregation based on user actions," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, April 2016, pp. 767–771.

[92] J. J. Davis and A. J. Clark, "Data preprocessing for anomaly based network intrusion detection: A review," *Computers & Security*, vol. 30, no. 6, pp. 353 – 375, 2011.

[93] J. J. Davis and E. Foo, "Automated feature engineering for HTTP tunnel detection," *Computers & Security*, vol. 59, pp. 166 – 185, 2016.

[94] A. Moore, M. Crogan, A. W. Moore, Q. Mary, D. Zuev, D. Zuev, and M. L. Crogan, "Discriminators for use in flow-based classification," University of London, Tech. Rep., 2005.

[95] L. Peng, B. Yang, Y. Chen, and Z. Chen, "Effectiveness of statistical features for early stage internet traffic identification," *International Journal of Parallel Programming*, vol. 44, no. 1, pp. 181–197, 2016.

[96] 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.

[97] J.-H. H. Hyun-Min An, Su-Kang Lee and M.-S. Kim, "Traffic identification based on applications using statistical signature free from abnormal TCP behavior," *Journal of Information Science and Engineering*, vol. 31, pp. 1669–1692, 2015.

[98] M. Iliofotou, M. Faloutsos, and M. Mitzenmacher, "Exploiting dynamicity in graph-based traffic analysis: Techniques and applications," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '09, 2009, pp. 241–252.

[99] M. Berlingerio, D. Koutra, T. Eliassi-Rad, and C. Faloutsos, "Netsimile: A scalable approach to size-independent network similarity," *CoRR*, vol. abs/1209.2684, 2012. [Online]. Available: http://arxiv.org/abs/1209.2684

[100] A. Jakalan, J. Gong, Q. Su, X. Hu, and A. M. Abdelgder, "Social relationship discovery of IP addresses in the managed IP networks by observing traffic at network boundary," *Computer Networks*, vol. 100, pp. 12 – 27, 2016.

[101] F. Iglesias and T. Zseby, "Time-activity footprints in IP traffic," *Computer Networks*, vol. 107, Part 1, pp. 64 – 75, 2016.

[102] A. Hajjar, J. Khalife, and J. Daz-Verdejo, "Network traffic application identification based on message size analysis," *Journal of Network and Computer Applications*, vol. 58, pp. 130 – 143, 2015.

[103] V. Bandara, A. Pezeshki, and P. J. Anura, "Modeling spatial and temporal behavior of internet traffic anomalies," in *IEEE Local Computer Network Conference*, 2010, pp. 384–391.

[104] H. Shi, H. Li, D. Zhang, C. Cheng, and W. Wu, "Efficient and robust feature extraction and selection for traffic classification," *Computer Networks*, vol. 119, pp. 1 – 16, 2017.

[105] J. Zhang, Y. Xiang, Y. Wang, W. Zhou, Y. Xiang, and Y. Guan, "Network traffic classification using correlation information," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 104–117, 2013.

[106] Y. Wang, Y. Xiang, J. Zhang, and S. Yu, "A novel semi-supervised approach for network traffic clustering," in *2011 5th International Conference on Network and System Security*, 2011, pp. 169–175.

[107] D. M. Divakaran, L. Su, Y. S. Liau, and V. L. L. Thing, "SLIC: Self-learning intelligent classifier for network traffic," *Computer Networks*, vol. 91, pp. 283 – 297, 2015.

[108] A. Dainotti, W. de Donato, A. Pescape, and P. S. Rossi, "Classification of network traffic via packet-level hidden markov models," in *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, 2008, pp. 1–5.

[109] C. Yin, S. Li, and Q. Li, "Network traffic classification via HMM under the guidance of syntactic structure," *Computer Networks*, vol. 56, no. 6, pp. 1814 – 1825, 2012.

[110] T. Yildirim and P. Radcliffe, "VoIP traffic classification in IPSec tunnels," in *2010 International Conference on Electronics and Information Engineering*, vol. 1, Aug 2010, pp. V1–151–V1–157.

[111] R. Alshammari and A. N. Zincir-Heywood, "Identification of VoIP encrypted traffic using a machine learning approach," *Journal of King Saud University - Computer and Information Sciences*, vol. 27, no. 1, pp. 77 – 92, 2015.

[112] E. Adi, Z. Baig, and P. Hingston, "Stealthy denial of service (DoS) attack modelling and detection for HTTP/2 services," *Journal of Network and Computer Applications*, vol. 91, pp. 1 – 13, 2017.

[113] R. Bar Yanai, M. Langberg, D. Peleg, and L. Roditty, "Realtime classification for encrypted traffic," in *Proceedings of the 9th International Conference on Experimental Algorithms*, ser. SEA'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 373–385.

[114] P. Branch and J. But, "Rapid and generalized identification of packetized voice traffic flows," in *37th Annual IEEE Conference on Local Computer Networks*, 2012, pp. 85–92.

[115] Y. Okada, S. Ata, N. Nakamura, Y. Nakahira, and I. Oka, "Application identification from encrypted traffic based on characteristic changes by encryption," in *2011 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*, May 2011, pp. 1–6.

[116] K. Goseva-Popstojanova, G. Anastasovski, A. Dimitrijevikj, R. Pantev, and B. Miller, "Characterization and classification of malicious web traffic," *Computers & Security*, vol. 42, pp. 92 – 115, 2014.

[117] M. Soysal and E. G. Schmidt, "Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison," *Performance Evaluation*, vol. 67, no. 6, pp. 451 – 467, 2010.

[118] M. Korczynski and A. Duda, "Classifying service flows in the encrypted skype traffic," in *2012 IEEE International Conference on Communications (ICC)*, June 2012, pp. 1064–1068.

[119] Y. ning Dong, J. jie Zhao, and J. Jin, "Novel feature selection and classification of internet video traffic based on a hierarchical scheme," *Computer Networks*, vol. 119, pp. 102 – 111, 2017.

[120] R. Singh, H. Kumar, and R. Singla, "An intrusion detection system using network traffic profiling and online sequential extreme learning machine," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8609 – 8624, 2015.

[121] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescap, "Multi-classification approaches for classifying mobile app traffic," *Journal of Network and Computer Applications*, 2017.

[122] T. Bakhshi and B. Ghita, "On Internet traffic classification: A two-phased machine learning approach," *Journal of Computer Networks and Communications*, vol. 2016, no. 2048302, 2016.

[123] S. E. Middleton and S. Modafferi, "Scalable classification of QoS for real-time interactive applications from IP traffic measurements," *Computer Networks*, vol. 107, pp. 121 – 132, 2016.

[124] M. Mohammadi, B. Raahemi, A. Akbari, H. Moeinzadeh, and B. Nasersharif, "Genetic-based minimum classification error mapping for accurate identifying peer-to-peer applications in the internet traffic," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6417 – 6423, 2011.

[125] M. feng SUN and J. tao CHEN, "Research of the traffic characteristics for the real time online traffic classification," *The Journal of China Universities of Posts and Telecommunications*, vol. 18, no. 3, pp. 92 – 98, 2011.

[126] R. Yuan, Z. Li, X. Guan, and L. Xu, "An SVM-based machine learning method for accurate internet traffic classification," *Information Systems Frontiers*, vol. 12, no. 2, pp. 149–156, 2010.

[127] P. Wang, S. C. Lin, and M. Luo, "A framework for qos-aware traffic classification using semi-supervised machine learning in SDNs," in *2016 IEEE International Conference on Services Computing (SCC)*, 2016, pp. 760–765.

[128] R. Alshammari and A. N. Zincir-Heywood, "How robust can a machine learning approach be for classifying encrypted VoIP?" *Journal of Network and Systems Management*, vol. 23, no. 4, pp. 830–869, Oct 2015.

[129] G. L. Sun, Y. Xue, Y. Dong, D. Wang, and C. Li, "A novel hybrid method for effectively classifying encrypted traffic," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, Dec 2010, pp. 1–5.

[130] Y. Lai, Y. Chen, Z. Liu, Z. Yang, and X. Li, "On monitoring and predicting mobile network traffic abnormality," *Simulation Modelling Practice and Theory*, vol. 50, pp. 176 – 188, 2015.

[131] R. Lin, O. Li, Q. Li, and Y. Liu, "Unknown network protocol classification method based on semi-supervised learning," in *2015 IEEE International Conference on Computer and Communications (ICCC)*, 2015, pp. 300–308.

[132] L. Peng, B. Yang, and Y. Chen, "Effective packet number for early stage internet traffic identification," *Neurocomputing*, vol. 156, pp. 252 – 267, 2015.

[133] A. Rizzi, A. Iacovazzi, A. Baiocchi, and S. Colabrese, "A low complexity real-time internet traffic flows neuro-fuzzy classifier," *Computer Networks*, vol. 91, pp. 752 – 771, 2015.

[134] M. H. Bhuyan, D. Bhattacharyya, and J. Kalita, "A multi-step outlier-based anomaly detection approach to network-wide traffic," *Information Sciences*, vol. 348, pp. 243 – 271, 2016.

[135] X. Li, F. Qi, D. Xu, and X. s. Qiu, "An internet traffic classification method based on semi-supervised support vector machine," in *2011 IEEE International Conference on Communications (ICC)*, 2011, pp. 1–5.

[136] A. Raghuramu, P. H. Pathak, H. Zang, J. Han, C. Liu, and C.-N. Chuah, "Uncovering the footprints of malicious traffic in wireless mobile networks," *Computer Communications*, vol. 95, pp. 95 – 107, 2016.

[137] J. Zhang, Y. Xiang, W. Zhou, and Y. Wang, "Unsupervised traffic classification using flow statistical properties and IP packet payload," *Journal of Computer and System Sciences*, vol. 79, no. 5, pp. 573 – 585, 2013.

[138] J. Zhang, C. Chen, Y. Xiang, W. Zhou, and Y. Xiang, "Internet traffic classification by aggregating correlated naive bayes predictions," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 5–15, 2013.

[139] J. Yang, L. Yuan, Y. He, and L. ying Chen, "Timely traffic identification on P2P streaming media," *The Journal of China Universities of Posts and Telecommunications*, vol. 19, no. 2, pp. 67 – 73, 2012.

[140] G.-Z. Lin, Y. Xin, X.-X. Niu, and H.-B. Jiang, "Network traffic classification based on semi-supervised clustering," *The Journal of China Universities of Posts and Telecommunications*, vol. 17, pp. 84 – 88, 2010.

[141] Y. Wang, Y. Xiang, J. Zhang, W. Zhou, and B. Xie, "Internet traffic clustering with side information," *Journal of Computer and System Sciences*, vol. 80, no. 5, pp. 1021 – 1036, 2014.

[142] R. Raveendran and R. Menon, "An efficient method for internet traffic classification and identification using statistical features," *International Journal of Engineering Research & Technology (IJERT)*, vol. 4, pp. 297–303, 2015.

[143] Z. Liu, R. Wang, M. Tao, and X. Cai, "A class-oriented feature selection approach for multi-class imbalanced network traffic datasets based on local and global metrics fusion," *Neurocomputing*, vol. 168, no. Supplement C, pp. 365 – 381, 2015.

[144] F. Ertam and E. Avc, "A new approach for internet traffic classification: GA-WK-ELM," *Measurement*, vol. 95, pp. 135 – 142, 2017.

[145] R. Dubin, A. Dvir, O. Pele, O. Hadar, I. Richman, and O. Trabelsi, "Real time video quality representation classification of encrypted HTTP adaptive video streaming - the case of safari," *CoRR*, vol. abs/1602.00489, 2016. [Online]. Available: http://arxiv.org/abs/1602.00489

[146] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Information Sciences*, vol. 378, pp. 484 – 497, 2017.

[147] A. Juvonen, T. Sipola, and T. Hmlinen, "Online anomaly detection using dimensionality reduction techniques for HTTP log analysis," *Computer Networks*, vol. 91, pp. 46 – 56, 2015.

[148] N.-F. Huang, G.-Y. Jai, H.-C. Chao, Y.-J. Tzang, and H.-Y. Chang, "Application traffic classification at the early stage by characterizing application rounds," *Information Sciences*, vol. 232, pp. 130 – 142, 2013.

[149] A. Callado, J. Kelner, D. Sadok, C. A. Kamienski, and S. Fernandes, "Better network traffic identification through the independent combination of techniques," *Journal of Network and Computer Applications*, vol. 33, no. 4, pp. 433 – 446, 2010.

[150] L. Wang, *Data Mining with Computational Intelligence*. Berlin, Heidelberg: Springer-Verlag, 2009.

[151] H. Zhang, G. Lu, M. T. Qassrawi, Y. Zhang, and X. Yu, "Feature selection for optimizing traffic classification," *Computer Communications*, vol. 35, no. 12, pp. 1457 – 1471, 2012.

[152] A. Fahad, Z. Tari, I. Khalil, I. Habib, and H. Alnuweiri, "Toward an efficient and scalable feature selection approach for internet traffic classification," *Computer Networks*, vol. 57, no. 9, pp. 2040 – 2057, 2013.

[153] A. Fahad, Z. Tari, I. Khalil, A. Almalawi, and A. Y. Zomaya, "An optimal and stable feature selection approach for traffic classification based on multi-criterion fusion," *Future Generation Computer Systems*, vol. 36, pp. 156 – 169, 2014.

[154] L. Zhen and L. Qiong, "A new feature selection method for internet traffic classification using ml," *Physics Procedia*, vol. 33, pp. 1338 – 1345, 2012, 2012 International Conference on Medical Physics and Biomedical Engineering (ICMPBE2012).

[155] H. Shi, H. Li, D. Zhang, C. Cheng, and X. Cao, "An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification," *Computer Networks*, vol. 132, pp. 81 – 98, 2018.

[156] Z. Liu and Q. Liu, "Studying cost-sensitive learning for multi-class imbalance in Internet traffic classification," *The Journal of China Universities of Posts and Telecommunications*, vol. 19, no. 6, pp. 63–72, 2012.

[157] M. C. Belavagi and B. Muniyal, "Performance evaluation of supervised machine learning algorithms for intrusion detection," *Procedia Computer Science*, vol. 89, pp. 117 – 123, 2016.

[158] K. Lalitha and V. Josna, "Traffic verification for network anomaly detection in sensor networks," *Procedia Technology*, vol. 24, pp. 1400 – 1405, 2016, international Conference on Emerging Trends in Engineering, Science and Technology (ICETEST - 2015).

[159] R. Alshammari and A. N. Zincir-Heywood, "Can encrypted traffic be identified without port numbers, IP addresses and payload inspection?" *Computer Networks*, vol. 55, no. 6, pp. 1326 – 1350, 2011.

[160] M. M. U. Rathore, "Threshold-based generic scheme for encrypted and tunneled voice flows detection over IP networks," *Journal of King Saud University - Computer and Information Sciences*, vol. 27, no. 3, pp. 305 – 314, 2015.

[161] D. Zhang, C. Zheng, H. Zhang, and H. Yu, "Identification and analysis of skype peer-to-peer traffic," in *2010 Fifth International Conference on Internet and Web Applications and Services*, 2010, pp. 200–206.

[162] J. A. Caicedo-Muoz, A. L. Espino, J. C. Corrales, and A. Rendn, "Qos-classifier for vpn and non-vpn traffic based on time-related features," *Computer Networks*, vol. 144, pp. 271 – 279, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128618307321

[163] S. Miller, K. Curran, and T. Lunney, "Multilayer perceptron neural network for detection of encrypted vpn network traffic," in *CiberSA*, 2018.

[164] G. Draper-Gil, A. Habibi, M. Saiful, and A. Ghorbani, "Characterization of encrypted and vpn traffic using time-related features," in *2nd International Conference on Information Systems Security and Privacy (ICISSP 2016)*, 2016, pp. 407–414.

[165] S. E. Gmez, B. C. Martnez, A. J. Snchez-Esguevillas, and L. H. Callejo, "Ensemble network traffic classification: Algorithm comparison and novel ensemble scheme proposal," *Computer Networks*, vol. 127, pp. 68 – 80, 2017.

[166] Y. Du and R. Zhang, "Design of a method for encrypted P2P traffic identification using k-means algorithm," *Telecommunication Systems*, vol. 53, no. 1, pp. 163–168, 2013.

[167] D. J. Arndt and A. N. Zincir-Heywood, "A comparison of three machine learning techniques for encrypted network traffic analysis," in *2011 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, April 2011, pp. 107–114.

[168] H. Singh, "Performance analysis of unsupervised machine learning techniques for network traffic classification," in *2015 Fifth International Conference on Advanced Computing Communication Technologies*, 2015, pp. 401–404.

[169] P. V. Amoli and T. Hmlinen, "A real time unsupervised NIDS for detecting unknown and encrypted network attacks in high speed network," in *2013 IEEE International Workshop on Measurements Networking (M N)*, Oct 2013, pp. 149–154.

[170] M. Lotfollahi, R. S. H. Zade, M. J. Siavoshani, and M. Saberian, "Deep packet: A novel approach for encrypted traffic classification using deep learning," *CoRR*, vol. abs/1709.02656, 2017. [Online]. Available: http://arxiv.org/abs/1709.02656

[171] T. Bujlow, T. Riaz, and J. M. Pedersen, "A method for classification of network traffic based on C5.0 machine learning algorithm," in *2012 International Conference on Computing, Networking and Communications (ICNC)*, Jan 2012, pp. 237–241.

[172] B. Huifeng, Z. Xianlong, Z. Hongfeng, and Z. Likun, "Traffic-load prediction based on echo state network improved by bayesian theory in 10G-EPON," *The Journal of China Universities of Posts and Telecommunications*, vol. 22, no. 2, pp. 69 – 73, 2015.

[173] G. Münz, H. Dai, L. Braun, and G. Carle, "Tcp traffic classification using markov models," in *Traffic Monitoring and Analysis*, F. Ricciato, M. Mellia, and E. Biersack, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 127–140.

[174] E. M. R. Oliveira, A. C. Viana, K. Naveen, and C. Sarraute, "Mobile data traffic modeling: Revealing temporal facets," *Computer Networks*, vol. 112, pp. 176 – 193, 2017.

[175] K.-W. Lim, S. Secci, L. Tabourier, and B. Tebbani, "Characterizing and predicting mobile application usage," *Computer Communications*, vol. 95, pp. 82 – 94, 2016, mobile Traffic Analytics.

[176] M.-Y. Su, "Using clustering to improve the KNN-based classifiers for online anomaly network traffic identification," *Journal of Network and*

Computer Applications, vol. 34, no. 2, pp. 722 – 730, 2011, efficient and Robust Security and Services of Wireless Mesh Networks.

[177] N. Kheir, "Behavioral classification and detection of malware through HTTP user agent anomalies," *Journal of Information Security and Applications*, vol. 18, no. 1, pp. 2 – 13, 2013.

[178] B. Schmidt, A. Al-Fuqaha, A. Gupta, and D. Kountanis, "Optimizing an artificial immune system algorithm in support of flow-based internet traffic classification," *Applied Soft Computing*, vol. 54, pp. 1 – 22, 2017.

[179] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, July 2017, pp. 43–48.

[180] V. Carela-Español, P. Barlet-Ros, A. Cabellos-Aparicio, and J. Solé-Pareta, "Analysis of the impact of sampling on netflow traffic classification," *Comput. Netw.*, vol. 55, no. 5, pp. 1083–1099, 2011.

[181] R. Keralapura, A. Nucci, and C.-N. Chuah, "A novel self-learning architecture for P2P traffic classification in high speed networks," *Computer Networks*, vol. 54, no. 7, pp. 1055 – 1068, 2010.

[182] Z. Xu, L. Ma, and J. Sun, "Efficient tri-ary search tree based packet classification algorithm," *IET Conference Proceedings*, pp. 833–836(3), January 2007.

[183] W. Pak and Y. Choi, "High performance and high scalable packet classification algorithm for network security systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 1, pp. 37–49, Jan 2017.

[184] Z. Liu, R. Wang, and D. Tang, "Extending labeled mobile network traffic data by three levels traffic identification fusion," *Future Generation Computer Systems*, vol. 88, pp. 453 – 466, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17309937

[185] A. Satoh, T. Osada, T. Abe, G. Kitagata, N. Shiratori, and T. Kinoshita, "Traffic classification in mobile ip network," in *Proceedings of the 4th International Conference on Ubiquitous Information Technologies Applications*, 2009, pp. 1–6.

[186] D. Naboulsi, M. Fiore, S. Ribot, and R. Stanica, "Large-scale mobile traffic analysis: A survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 124–161, 2016.

[187] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *CoRR*, vol. abs/1803.04311, 2018. [Online]. Available: http://arxiv.org/abs/1803.04311

[188] J. Riihijarvi and P. Mahonen, "Machine learning for performance prediction in mobile cellular networks," *IEEE Computational Intelligence Magazine*, vol. 13, no. 1, pp. 51–60, Feb 2018.

[189] Y. Jin, N. Duffield, J. Erman, P. Haffner, S. Sen, and Z.-L. Zhang, "A modular machine learning system for flow-level traffic classification in large networks," *ACM Trans. Knowl. Discov. Data*, vol. 6, no. 1, pp. 4:1–4:34, Mar. 2012.

[190] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci, "Googling the internet: Profiling internet endpoints via the world wide web," *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 666–679, April 2010.

[191] M. Pietrzyk, J.-L. Costeux, G. Urvoy-Keller, and T. En-Najjary, "Challenging statistical classification for operational usage: The adsl case," in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '09, 2009, pp. 122–135.

[192] L. Grimaudo, M. Mellia, E. Baralis, and R. Keralapura, "Select: Self-learning classifier for internet traffic," *IEEE Transactions on Network and Service Management*, vol. 11, no. 2, pp. 144–157, June 2014.

[193] B. Ng, M. Hayes, and W. K. G. Seah, "Developing a traffic classification platform for enterprise networks with sdn: Experiences amp;amp; lessons learned," in *2015 IFIP Networking Conference (IFIP Networking)*, May 2015, pp. 1–9.

[194] L. Bertaux, S. Medjiah, P. Berthou, S. Abdellatif, A. Hakiri, P. Gelard, F. Planchou, and M. Bruyere, "Software defined networking and virtualization for broadband satellite networks," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 54–60, March 2015.

[195] E. Exposito, *Advanced Transport Protocols: Designing the Next Generation*. Wiley-ISTE Ltd, 2013.

[196] E. Exposito and C. Diop, *Smart SOA Platforms in Cloud Computing Architectures*. Wiley-ISTE Ltd, 2014.

[197] J. Aguilar, J. Torres, and K. Aguilar, "Autonomic communication system based on cognitive techniques," *KES Journal*, vol. 22, pp. 17–37, 2018.

[198] "ULAKNET data," http://ulakbim.tubitak.gov.tr/en/hizmetlerimiz/data-warehouse, accessed: 2017-09-27.

[199] K. Cho, K. Mitsuya, and A. Kato, "Traffic data repository at the wide project," in *Proceedings of the Annual Conference on USENIX Annual Technical Conference*, ser. ATEC '00. Berkeley, CA, USA: USENIX Association, 2000, pp. 51–51.

[200] M. W. G. of the WIDE Project, "MAWI data," http://mawi.wide.ad.jp/mawi/, accessed: 2017-09-27.

[201] CAIDA, "CAIDA data," http://www.caida.org/data/, accessed: 2017-09-27.

[202] The University of Waikato, "Traffic traces," http://wand.net.nz/wits/waikato/1/20040507-233830-64.php, accessed: 2017-09-27.

[203] The University of Auckland, "Traffic traces," http://wand.net.nz/wits/auck/8/20031202-090000.php, accessed: 2017-09-27.

[204] RIPE Network Coordination Centre, "Traffic traces," https://labs.ripe.net/datarepository/data-sets/nlanr-pma-data, accessed: 2017-09-27.

[205] "TSTAT telecommunication networks group - politecnico di torino," http://ostinato.org/, accessed: 2017-07-18.

[206] "DATCAT internet measurement data catalog," http://imdc.datcat.org/collection/1-070W-6=tcrsg-collection, accessed: 2017-03-10.

[207] "The telecommunication networks group unibs," http://netweb.ing.unibs.it/~ntw/, accessed: 2017-07-18.

[208] "Lbnl/icsi enterprise tracing project," http://www.icir.org/enterprise-tracing/Overview.html, accessed: 2017-07-18.

[209] "Kdd cup 1999 data," http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, accessed: 2017-09-27.

[210] "DARPA intrusion detection data sets," https://ll.mit.edu/ideval/data/, accessed: 2017-09-27.

XI. Appendix

A. Appendix A

This section lists in Table IV the most common public available datasets found for traffic analysis.

B. Appendix CC

Table V depicts the trends for the observation process. Table VI depicts the trends for the feature engineering process. Table VII depicts the trends for the algorithm selection process. Table VIII depicts the trends for the model construction process.

TABLE IV: Most common publicly available datasets found in the papers reviewed.

Dataset source	Description	Year
ULAKNET data [198]	Category and application classification	2008
ISP [106]	Traffic identification	2010
Wide [199]	Traffic identification	2009
Cambridge [94]	11 datasets for traffic analysis	2005
MAWI [200]	Different dataset for traffic analysis	2010-2014
CAIDA [201]	Different dataset for traffic analysis	2016
University of Waikato [202]	Different dataset for traffic analysis	2004
University of Auckland [203]	Different dataset for traffic analysis	2003
RIPE NCC organization [204]	Different dataset for traffic analysis	2010
ITALY [205]	Several applications	2008-2017
DatCat [206]	Several applications	2013-2017
UNIBS [207]	Several applications	2009
LBNL [208]	Several applications	2004
KKDCup[209]	Anomaly detection	1999
DARPA [210]	Anomaly detection	1998-2000

TABLE V: Trends of the selected papers for the data collection phase.

Characteristic	**Papers**							
Traffic	Real: [96], [54], [107],[119],[144], [102], [147] , [135], [130], [131], [136],[132],[133], [138], [134], [146], [101], [106], [141], [128], [129], [105], [142], [143], [121]				Emulated: [122],[116], [148],[123], [124],[120],[117],[125],[109] , [139], [126], [100], [110],[137], [140], [127], [145],[111], [112],[114], [118], [115], [113], [149]			
Public available	Yes: [96], [107], [144],[102], [130], [131],[132],[142], [133], [138],[105], [134], [146], [101], [106], [141], [128], [129], [143]		No: [54], [119], [147], [135], [136], [121]		Yes: [120], [145]		No: [122], [116],[148],[123], [124],[117],[125],[109], [139],[126],[100], [110], [137], [140], [111], [112], [113], [114], [118], [115], [127], [149]	
Encrypted	Yes: [128], [129]	No: [96], [107], [144], [102], [130], [131],[132], [142], [133], [138], [105], [134], [146], [101], [106], [141], [143]	Yes: [121]	No: [54], [119], [147], [135], [136]	Yes: [145]	No: [120]	Yes: [110], [111], [112], [113],[114],[115]	No: [122],[116],[148],[123], [124],[117],[125],[109], [139], [126], [100], [137], [140], [127], [118], [149]

TABLE VI: Trends of the selected papers for the feature engineering phase.

Characteristic	**Papers**							
FE	Time: [144], [101], [109]		STATs: [110], [111], [112],[113],[114], [115], [116], [117],[118], [119], [120], [121], [122], [123], [124], [125],[126], [127], [128], [129], [96], [130], [131], [132], [133], [134], [106], [135], [136], [137], [138], [105], [139], [140], [107],[141], [142], [143]		Graph: [54], [100]		Hybrid: [145], [146], [147], [148], [102], [149]	
FS	Yes: [101]	No: [144], [109]	Yes: [112], [116], [122], [130], [132], [133], [134], [135], [136], [143]	No: [110], [111], [113], [114], [115], [117], [118], [119], [120], [121], [123], [124], [125], [139], [126], [127], [128], [129], [96], [131], [106], [141], [137], [105], [140], [107], [138], [142]	Yes:	No: [54], [100]	Yes: [147]	No: [145], [146], [148], [102], [149]

TABLE VII: Trends of the selected papers for the algorithm selection phase.

Characteristic	**Papers**															
FE	CC: [116], [112], [147], [130], [136], [132], [148], [123], [124], [117], [125], [109], [126], [110],[114], [118],[115], [96], [54], [107], [119], [138], [128], [129], [142]				MClass&E: [143], [139], [111], [102], [105], [149], [121]				Clust: [134], [101], [120], [100], [137], [140], [145]				H&A: [122], [135], [133],[127], [144], [131], [146], [106], [113], [141]			
Objective	AppC: [123], [124], [117]	AppP: [132], [125], [126], [110], [114], [118], [107], [119], [128], [142], [148], [109], [115], [96], [138], [129]	AD: [116], [112], [130], [136], [54], [147]	OTHERS:	AppC: [143], [139], [111], [105], [149]	AppP: [102], [121]	AD:	OTHERS:	AppC: [145]	AppP: [137]	AD: [120], [134]	OTHERS: [100], [101]	AppC: [122], [135], [127], [113]	AppP: [133], [144], [131], [106], [141]	AD: [146]	OTHERS:

TABLE VIII: Trends of the selected papers for the model deployment trend.

Characteristic	**Papers**					
On-line implementation	YES: [110], [123], [125], [139],[128], [145],[122],[113], [127], [133], [112], [147], [100]			NNS: [111], [114], [117], [118], [119], [124], [135], [126], [107], [105], [142], [149], [143], [132], [115], [129], [96], [138], [144], [148], [102], [131], [106], [140], [141], [109], [137], [121], [116], [136], [130], [120], [54], [134], [146],[101]		
Reconfiguration	RTraining: [123], [139], [127]	SLE:	ONS: [110], [125], [128], [145],[122], [113], [133],[112], [147],[100]	RTraining: [119], [107]	SLE:	ONS: [111], [114], [117], [118], [124],[135], [126], [105], [142], [149], [143], [121], [132], [115], [129], [96], [138], [144], [148], [102], [131], [106], [140], [141], [109], [137], [116], [136], [130], [120], [54], [134], [146]