Anonymity Services Tor, I2P, JonDonym: Classifying in the Dark (Web)

Antonio Montieri, Domenico Ciuonzo, *Senior Member, IEEE,* Giuseppe Aceto, and Antonio Pescapé, *Senior Member, IEEE*

Abstract—Traffic Classification (TC) is an important tool for several tasks, applied in different fields (security, management, traffic engineering, R&D). This process is impaired or prevented by privacy-preserving protocols and tools, that encrypt the communication content, and (in case of *anonymity tools*) additionally hide the source, the destination, and the nature of the communication. In this paper, leveraging a public dataset released in 2017, we provide classification results with the aim of investigating to which degree the specific anonymity tool (and the traffic it hides) can be identified, when compared to the traffic of other considered anonymity tools, using five machine learning classifiers. Initially, flow-based TC is considered, and the effects of feature importance and temporal-related features to the network are investigated. Additionally, the role of finer-grained features, such as the (joint) histogram of packet lengths (and inter-arrival times), is determined. Successively, "early" TC of anonymous networks is analyzed. Results show that the considered anonymity networks (Tor, I2P, JonDonym) can be easily distinguished (with an accuracy of 99.87% and 99.80%, in case of flow-based and early-TC, respectively), telling even the specific application generating the traffic (with an accuracy of 73.99% and 66.76%, in case of flow-based and early-TC, respectively).

Index Terms—Dark web; Dark net; Tor; I2P; JonDonym; traffic classification; anonymity; privacy; security.

1 Introduction

The increasing amount of people's online activities over last years has lead to a growing concern on their privacy and anonymity. As a consequence, Anonymity Tools (ATs) have been increasingly employed by Internet users to achieve privacy in varying degrees, i.e. to hide the source, the destination, and the nature of the communication, besides encrypting the content itself. Additionally, in many cases they are even capable of hiding the users' identity to the final destination (i.e. the web-server). These services provide anonymity to the users by forwarding their traffic through multiple stations, encrypting and decrypting it multiple times, until the users' data reach their destination. By doing so, users' data preserve their anonymity since each station being part of the path knows *only part of the information*. Hence, tracing of users' data within these networks is extremely difficult. Indeed, from users' perspective, these tools allow browsing the web circumventing provider-enforced restrictions or running applications without revealing users' identity and location to any intermediary observing the traffic. Among the several ATs developed in recent years, The Onion Router (Tor) [1], the Invisible Internet Project (I2P) [2] and JonDonym (formerly known as Java Anon Proxy, JAP, or Web-Mix) [3] are the *most popular*. In recent years, ATs have been investigated in several studies from disparate perspectives, collectively covering a wide spectrum of topics. These included very narrow aspects of anonymity "realm", such as the design improvement of a specific AT, its performance and delay analysis, development of effective attacks to be performed on it, users' behavior analysis and identity disclosure risk, censoring policies for ATs [4], so to name a few.

Although many important aspects of ATs equally deserve attention, a key issue is to *understand whether their (encrypted) traffic data can be classified and, if so, to which depth*. More specifically, it is interesting to ascertain to which degree an external observer can recognize an AT and how fine would be the fingerprinting granularity achievable, that is, whether traffic types and/or services hidden into them could be inferred. This investigation is equally useful to designers of anonymity networks, as it suggests how privacy of anonymity networks could be further *robustified*. Indeed, Traffic Classification (TC) is an important part of Internet traffic engineering and has applications in several fields such as network monitoring, security, application identification, anomaly detection, accounting, advertising, and service differentiation [5], [6]. From an operational standpoint, TC mechanisms consist in associating (labeling) traffic flows with specific application types. TC has gained on importance in recent years due to growing incentives to disguise certain applications [7], comprising those generating *anonymous traffic*. Therefore, TC of ATs traffic is an appealing (and open) research field.

TC methods range from (earlier) port-based methods, to those based on payload inspection (termed Deep Packet Inspection techniques, DPI [8], [9]) and, more recently, to those based on Machine Learning (ML) classifiers (either supervised and unsupervised). The latter class of TC approaches provides decisions based on the sole observation of traffic-flow [10] or packet-based features [11], [12]. Thus, ML-based

- *Antonio Montieri is with the University of Napoli Federico II (Italy). E-mail: antonio.montieri@unina.it*
- *Domenico Ciuonzo is with NM2 s.r.l. (Italy). E-mail: ciuonzo@nm-2.com*
- *Giuseppe Aceto and Antonio Pescapé are with the University of Napoli Federico II (Italy) and with NM2 s.r.l. (Italy). E-mail: giuseppe.aceto@unina.it; pescape@unina.it*

techniques *uniquely suit* to anonymous (encrypted) traffic analysis.

Exacerbating the lack of data for experimenting with TC approaches, it is worth mentioning that one of the main issues of research efforts in anonymity field is given by the fact that real data are *hardly publicly available*, which prevents experiments repeatability and, as a matter of fact, precludes unanimous and shared conclusions. Indeed, previous works on ATs have been based on: (*i*) data collected within a simulated environment [13]; (*ii*) data generated from private anonymous networks [14]; (*iii*) data generated from real traffic on anonymous networks by researchers[1] themselves [15], [16], [17], [18]. Unfortunately, in the latter case, researchers have been reluctant to making the collected data publicly available for reasons of users' privacy.

A fundamental opportunity in this direction, allowing to answer the question constituting the basis of the present study (i.e. whether identifiability of anonymous networks is possible), is represented by the recently released Anon17 dataset [19]. Indeed, this public dataset consists of a collection of traces gathered by different anonymity networks, as well as related services and applications running inside them. To the best of authors' knowledge, no similar datasets have been made available online up to date. Hence, Anon17 represents an important (shared) workbench for research studies on the topic.

In view of these reasons, the main contribution of this paper is a detailed study on whether anonymity networks (such as Tor, JonDonym, and I2P) can be discerned[2]. Our analysis is carried out at different levels of granularity, as we try to infer whether the *Anonymity Network* being observed (referred to as L1 in what follows) can be classified and, in affirmative case, whether the *Traffic Type* (L2) and *Application* (L3) transported *hidden* within them could be inferred. To this end, we consider five ML classifiers: three of them are based on the *Bayesian approach* (i.e. Naïve Bayes, Multinomial Naïve Bayes, and Bayesian Networks), whereas the other two on the well-known *decision trees* (i.e. C4.5 and Random Forest). The adoption of a pool of classifiers (as opposed to a single one) is not only motivated by an obvious need for comparison, but also to investigate minutely the (number and nature of) relevant features needed for an accurate classification, as well as the need to reach conclusions not coupled to a specific classifier. The present analysis is devoted both to the development and analysis of classifiers based on flow-originated features or pertaining to a short (early) sequence of packets. To the best of our knowledge, there is no similar classification-based analysis of ATs in the literature, both in terms of a similar viewpoint and detail of the analysis. The obtained results show that anonymity networks *can* be easily discerned, and the traffic type and the service running within it can be reasonably inferred as well (by a judicious use of the appropriate classifier and optimized set of features).

The remainder of the paper is organized as follows. Sec. 2 discusses related works, whereas Sec. 3 describes the considered TC framework of ATs; experimental results are reported in Sec. 4; finally, Sec. 5 provides conclusions and future directions.

2 Related Work

Up to authors' knowledge, there are no studies focused on the classification and identification of different anonymity services at various levels of granularity, that is discerning in the ET they generate: the specific Anonymous Network (L1), the Traffic Type (L2), and the Application (L3). The principal reason is the lack of suitable and available datasets. As a consequence, henceforth, we provide a wider discussion on the state-of-the-art of anonymous (encrypted) traffic investigation, being subject of many works in last years. Then, in the ending part of the section, we specifically focus on works dealing with anonymous TC to underline their difference with respect to the present study.

First attempts of analyzing anonymity networks are made by means of *emulation tools* [13] or in *private networks* [14], focusing on Tor case. In detail, Tor network requires all its traffic to be relayed *at least* through three nodes before reaching its destination. The entry point of Tor network is usually denoted as *entry node*, whereas the last relay before reaching the destination (i.e. the *exit node*) "lends" its IP address, which is interpreted as the source of the traffic by the destination. On the other hand, middle nodes add to the speed and robustness of the Tor network without making the owner of the relay look like the source of the traffic. When using Tor in its standard setting, Tor utilization can be inferred by monitoring connections to (known) entry nodes. To this end, *Tor bridges* are alternative entry points to the Tor network that are not all listed publicly. Nonetheless, since Tor communications operate on a cell-basis of a fixed length, they can be still identified even in the presence of bridges. For this reason, Tor *Pluggable Transports* (PTs) have been recently developed to disguise identification of traffic generated by the users connected to a certain Tor bridge, making it look like random (or something different from Tor traffic).

Specifically, in [13] a network emulator (called *ExperimenTor*) is presented, representing a test environment which allows modeling of relevant "actors" (i.e. Tor routers, bandwidth, users, and applications). On the other hand, a private network environment is set up in [14], with the intent of discriminating between (encrypted) HTTPS and Tor traffic. The reported comparison is based on the collection of the following traffic types: (*i*) regular HTTPS traffic; (*ii*) HTTP over a private Tor network; (*iii*) HTTPS over a private Tor network. Three ML classifiers (i.e. Random Forest, J48/C4.5 and AdaBoost) are there adopted, and shows that HTTP/HTTPS traffic over Tor can be detected with a score $\geq 93\%$ (when a 3.7% false-positive rate is ensured).

More recently, Rao et al. [21] propose an unsupervised approach, based on gravitational clustering, to detect flows of Tor network from mixed anonymous/non-anonymous traffic data. The experimental analysis is based on a dataset obtained by mixing (real-traffic) flows generated from different services (e.g., DNS, HTTP, HTTPS and SSH) and

[1] As mentioned earlier, the traffic on anonymity networks relies on passing the users' data through multiple nodes on the network. Since these nodes relay traffic for multiple users, collecting the data from these nodes will include traffic from several other users. This means that data are collected running a node and filtering its data so as to include only the "desired" traffic.

[2] Preliminary results in the same framework of this study have been published as a conference publication [20].

Tor-emulated flows obtained via *ExperimenToR*. Results provided report an average accuracy of 88% for gravitational clustering, outperforming other unsupervised techniques, such as K-means, expectation-maximization, and DBSCAN.

We mention that several other works have analyzed Tor, JonDonym, and I2P based on *real data* [15], [16], [18], [22], [23], focusing however on some *other* (equally-important though) *aspects*, such as evaluating the volume of traffic run within [15], designing the attack type [16], [23], discovering "anonymous" routers [18] or providing guidelines for evaluating privacy level of a generic AT [22]. For example, Ling et al. [23] propose a combination of an active attack and a detection mechanism to break Tor users' privacy, based on the size of transmitted packet from the web server to the user, through the router. Since Tor has a fixed cell size (512 bytes) but accommodates varying packet sizes, the modulation of buffer size at the sender (and hence of payload size) will reflect on a different (known) number of Tor cells: using big and small buffer sizes, a binary signal is thus encoded with the connection (in the form of controlled number of cells) so as to recover it back at the receiver (web client) side, effectively identifying the client involved in the connection. Equally important, a delay is added in between buffered packets before transmission to ensure correct detection of the encoded bit at the receiver. Indeed, this step obviates effects due to packets congestion, retransmission or any normal traffic behavior during the server-client path. Reported results show that this method requires only 10 packets to reach a 90% detection rate (when a 4% false-positive rate is ensured). Differently, the authors in [22] define five different key factors for determining the anonymity level of any generic AT (taking however Tor, I2P, and JonDonym as specific case studies) from the user's perspective. Then, a synthetic measure is proposed as a weighted combination of these factors, with weights arising from a pairwise comparison technique. The analysis highlights that although these ATs aim to provide total users anonymity, some users info contained in these ATs is available to the operators of the services.

More recently, a few works have analyzed real traffic from anonymity networks, *focusing on TC aspect*; we now discuss them in detail.

First, there is a corpus of literature pertaining to a specific application of TC, known as website fingerprinting, whose aim is to identify a specific web-page accessed by a client of encrypted and anonymized connections by observing patterns of data flows such as packet size and direction. Herrmann et al. [24] tackle the problem of website fingerprinting in the context of different privacy-enhancing technologies based on both single-hop (OpenSSL, OpenVPN) and multi-hop (Tor, JonDonym) systems, by proposing a multinomial Naïve Bayes classifier (along with a few additional variants), that relies on the normalized frequency distribution of IP packet sizes. The proposed classifier correctly identifies (in closed world) up to 97% of requests on a sample of 775 sites (and over $300,000$ real-world traffic dumps recorded over a two-month period) in single-hop systems, while performs poorly with multi-hop systems. Nonetheless, this study highlights that the latter networks are not capable of providing perfect protection against the proposed technique (3% and 20% of average accuracy in Tor and JonDonym networks, respectively). The above finding is further explored in [25], where a Support Vector Classifier is employed for the same problem over Tor and JonDonym (*separately*), underlining their incomplete anonymity. The traffic features comprise those based on volume, time, and direction, such as the number of packets/transmitted bytes in both directions and the percentage of incoming packets. The training set consists of 15500 instances related to 775 websites (20 instances each) on either Tor or JonDonym. The results show that (over a known set of websites) the detection rate improves over [24] from 3% to 55% (resp. from 20% to 80%) in Tor (resp. JonDonym) network. On the other hand, in the open-world (unknown websites) scenario, the training set includes 4000 URLs chosen from the 1 million most popular websites in Alexa ranking and other 1000 URLs (not included in the training set) are added to the test data. In this case, the detection rate is 73% (with 0.05% false-positive rate).

More recently, in [26] an improved (repeatable) website fingerprinting approach is proposed, and showed to be superior both in terms of detection accuracy and computational efficiency to existing alternatives. To provide a closer analysis to a realistic case, a huge representative dataset is collected, with the intent of avoiding simplified assumptions made in the previous works, allowing for the first time the evaluation of the website fingerprinting attack against Tor using realistic background noise. Using this data, the practical limits of website fingerprinting at Internet scale are explored (in an open-world setup), showing that web-page fingerprinting effectiveness does not scale for any considered page in the considered datasets and any state-of-the-art classifier. Specifically, this is underlined by a decrease of recall/precision pair with the size of the background sites to distinguish the monitored pages from. Almubayed et al. [27] also use supervised classifiers (e.g., Naïve Bayes, Bayesian Network, C4.5, Random Forest, and Support Vector Machine), to develop an identification method for discovering Tor-encrypted web-pages (belonging to the top 5 from Alexa web-site) from HTTPS (background) traffic, represented by top 100 sites from Alexa. Results (pertaining to identification of each single web page from those considered as HTTPS background) report very high (resp. low) TP (resp. FP) rates for all the identification problems analyzed and all the classifiers there employed ($\geq$ 99% and $\leq$ 1% for TP and FP rates, respectively).

Differently, Springall et al. [28] present two novel methods (pertaining to HTTP and SSH traffic) to identify, at a Tor exit node, network connections originated behind the Tor network from those that have not, so as to allow a content provider filtering on a per-connection basis rather than per-IP basis. The proposed methods identify Tor inbound connections through the use of delay and round-trip time features, respectively. In order to evaluate performance, results are presented for two small-scale experiments (testing performance with HTTP and SSH traffic, respectively), showing very high identification rates (100% and 98.99% respectively) when partitioning network connections into Tor and non-Tor originating connections.

Similarly, Bai at al. [29] propose a fingerprinting method to identify Tor and Web-Mix (viz. JonDonym) networks. Their method uses specific strings, packet length, and frequency of the packets. The proposed approach is tested on simulated networks and achieves $\geq 95\%$ of accuracy in identifying both systems (Tor and Web-Mix). In [30] the authors propose a method based on Hidden Markov Models (HMMs) to classify encrypted Tor traffic in 4 categories: P2P, FTP, IM, and Web. The features employed are based on burst volumes and directions, extracted from Tor flows. Then, HMMs are employed to build inbound and output models of the application types considered. The proposed method is reported to obtain a maximum accuracy of 92%.

AlSabah et at. [17] propose another ML-based approach for recognizing applications (browsing, streaming, and BitTorrent) used by Tor's users based on the adoption of different classifiers (Naïve Bayes, Bayesian Network, functional and logistic model trees), leveraging *circuit-level* (circuit lifetime and the corresponding amount of data transferred) and *cell-level* info (inter-arrival time of the cells, including their statistics). Both *online* (cell-level info is used to classify the circuit while it is in use) and *offline* (cell- and circuit-level info is both capitalized to classify the circuit) approaches are considered, with the highest accuracy achieved for online (resp. offline) case equal to 97.8% (resp. 91%). Then, a similar setup is considered in [31], where four classifiers (Naïve Bayes, Bayesian Network, Random Forest, and C4.5) based on *traffic-flow* features are exploited to recognize user activities, and compared with classification based on *circuit-level* features. The results underline near-ideal accuracy (up to 100%) with both approaches, flow-based classification being nevertheless *less demanding*.[3]

Similarly, Shahbar and Zincir-Heywood [32] investigate whether Tor PTs can evade censorship systems based on flow-based (statistical) traffic analysis. Sadly, PTs are designed to hide the content of Tor connections only; thus, a flow-based analysis, in principle, can identify Tor traffic even when these obfuscation techniques are applied. The authors adopt a C4.5 classifier and demonstrate that PT-based obfuscation changes the content shape in a distinct way from Tor, conferring them their own unique fingerprints, hence making them recognizable via a statistical-based traffic classifier. The aforementioned analysis is then elaborated in [33], where the aim is effective flow-based recognition of Tor PTs in terms of describing the proper features, the sufficient amount of data, and the effect of data collection. The same authors analyze the effects of bandwidth sharing on I2P in [34], investigating both application and user profiling achievable by an attacker. The analysis resorts to a C4.5 classifier fed with flow-based features. The experimental results highlight that users and applications on I2P *can* be profiled. More specifically, a detrimental (resp. beneficial) effect of the shared bandwidth increase on applications (resp. users) profiling accuracy is observed. Moreover, it is noticed that the avoidance of shared client tunnels for all applications seems to boost applications profiling.

More recently, *Anon17* dataset is presented in [19]. As anticipated in Sec. 1, the dataset is composed by directional traffic-flows obtained by gathering data from three different ATs, namely Tor, I2P, and JonDonym. Furthermore, it provides information at increasing detail (i.e. traffic type and application levels), by providing labels for traffic flows pertaining to applications running on Tor and I2P (in different flavors), such as "Browsing" and "EEpsites", respectively, as well as the PTs employed on the Tor network. Up to authors' knowledge, no similar datasets are available publicly up to date. The sole exception is represented by the dataset described/analyzed in [35], containing however *only* Tor-traffic, belonging to eight different applications (Browsing, Audio, CHAT, Mail, P2P, FT, VOIP, and Video), and providing *only* time-related features.

3 Traffic Classification

In the following, first terms and concepts regarding traffic objects are introduced (Sec. 3.1), together with an overview of the classification features available in the Anon17 dataset (Sec. 3.2) and how these are capitalized, when automatic feature selection is exploited (Sec. 3.3); then the last part (Sec. 3.4) describes the classification algorithms adopted (with corresponding feature sets employed) for anonymous traffic analysis.

3.1 Traffic View

According to [19], the anonymous traffic contained in Anon17 is split into different *flows* [6], obtained as result of the application of the flow-exporting tool *Tranalyzer2* [36]. The direction of each flow is then marked as a feature (see details in Sec. 3.2), i.e. "A" and "B" for client-to-server and server-to-client, respectively. According to Tranalyzer2 documentation, the termination (segmentation) of an active flow depends on the activity or the lifetime of a connection [36].

3.2 Classification Features

The traffic features available in Anon17 dataset are obtained starting from Tranalyzer2 [36]. More specifically, the latter is an open source tool that generates flows from a captured traffic dump or directly by working on the network interface, based on the *libpcap* library. Tranalyzer2 tool is bundled with different basic plugins, being able to extract a plethora of features per flow.[4] However, the dataset provides only a subset of these features, since some of them have been removed (such as ICMP and VLAN features) because they do not provide useful fingerprinting information. Additionally, aiming at protecting users' privacy (and simulate a *true* ET scenario), IP addresses and payloads of the packets have also been removed from the dataset.

Therefore, Anon17 is provided in the form of a subset of 81 fields corresponding to features per flow extracted by the aforementioned tool [19]. For our classification problem we have removed the fields `min_pl`, `max_pl`, and `mean_pl`, as they seem repeated with respect to `minPktSz`, `maxPktSz`, and `avePktSize`, respectively, considering the specific configuration adopted in Tranalyzer2 for capturing the traffic. Additionally, as opposed to [20], we have discarded the

[3] Indeed, circuit-level classification uses the data collected at Tor's relay, whereas flow-level classification is based on data that could be captured *anywhere* between the user and the Tor's relay.

[4] Tranalyzer2 also enables development of user-defined plugins, thus virtually allowing to extract any desired feature.

Table 1: Classification Levels: Anonymous network (L1), Traffic Type (L2) and Application (L3), with total number of samples per class, and class label for L3 granularity.

L1 - Anonymous Network	L2 - Traffic Type	L3 - Application
Tor **(358919, a)**	Normal Tor Traffic **(5283, a)**	Tor **(5283, a)**
	Tor Apps **(252, b)**	Streaming **(84, b)**, Torrent **(84, c)**, Browsing **(84, d)**
	Tor Pluggable Transports **(353384, c)**	Flash proxy **(172324, e)**, FTE **(106237, f)**, Meek **(43152, g)**, Obfs3 **(14718, h)**, Scramble suit **(16953, i)**
I2P **(645708, b)**	I2P Apps Tunnels with other Tunnels [0% Bandwidth] **(195081, d)**	I2PSnark **(127349, j)**, jIRCii **(29357, k)**, Eepsites **(38375, l)**
	I2P Apps Tunnels with other Tunnels [80% Bandwidth] **(449987, e)**	I2PSnark **(149992, m)**, jIRCii **(149998, n)**, Eepsites **(149997, o)**
	I2P Apps **(640, f)**	I2PSnark **(62, p)**, jIRCii **(221, q)**, Eepsites **(145, r)**, Exploratory Tunnels **(86, s)**, Participating Tunnels **(126, t)**
JonDonym **(6335, c)**	JonDonym **(6335, g)**	JonDonym **(6335, u)**

initial/final timestamps of each flow (`time_first` and `time_last`) to avoid biased results, as this pair of features may be influenced by a sequential collection of the traffic traces belonging to different ATs and/or application types. Therefore, we have considered a reduced set of 76 fields, that has been exploited to extract *four different types of features set*, with the aim of providing a comprehensive analysis of TC of ATs.

The *first* set of features considered comprises 74 summarizing *flow-based* statistics, such as:

- Flow direction (A/B) and duration;
- No. of bytes/packets Tx/Rx (including bytes/packets Tx rate and stream asymmetry measures);
- Packet Length (PL) statistics (mean, min, max, median, quartiles, etc.);
- Inter-Arrival Time (IAT) statistics (mean, min, max, median, quartiles, etc.);
- TCP header-related features (window size, sequence number, TCP options, etc.);
- IP header-related features (type-of-service, time-to-live, IP flags, etc.);
- No. of connections (i) from source (destination) IP to different hosts and (ii) between source and destination IP during the lifetime of the flow.

As underlined in [19], since I2P network works on both TCP and UDP, for UDP connections over I2P the TCP-related features may have zero value.

The *second* and *third* sets of features are based on a (finer) histogram representation of PL and joint PL-IAT, respectively. These sets are obtained through an appropriate format conversion and/or marginalization [37] from the field `ps_iat_histo`. This field, as provided by Tranalyzer2, contains precise (non-binned) PL info, whereas applies a non-uniform binning to the (continuous-valued) IAT information.[5] Their use will be investigated similarly to [11] to understand whether finer-grained features can improve classification performance.

Finally, the *fourth* set of features corresponds to the sequence of pairs (Payload Length, IAT) of the first K packets of each flow. This set is extracted from the field `nfp_pl_iat`[6], containing the info corresponding to the first $K = 20$ packets, as set by Tranalyzer2 default options [38]. This set will be later employed to investigate the design of effective algorithms for *early* TC [39], [40] of ATs.

Note that each set of M features adopted by each classifier will be generically indicated with $f_1, \dots, f_M$ (or collectively as $\boldsymbol{f} \triangleq \begin{bmatrix} f_1 & \cdots & f_M \end{bmatrix}^T$) and the set of classes as $\Omega \triangleq \{c_1, \dots, c_L\}$. Finally, relative importance (based on statistical rankings) of features' within the *first set* will be later analyzed in Sec. 4.3.

3.3 Feature Selection

In what follows, we will adopt the classifiers being considered along with *feature selection techniques*, allowing to extract only the most informative features from a larger set (in our case, the *first* set of features considered, being composed of 74 flow-based statistics). The aim is to (possibly) improve further their performance, while reducing their computational complexity. To this end, in this study, we will consider feature selection based on a *filtering approach*, since *wrapper methods* may be considerably more complex and coupled to a specific classification algorithm [41]. More specifically, the approach adopted ranks the elements within the set based on the relative importance of each feature, evaluated as the *Pearson's correlation* with the class (random) variable. We remark that other feature selection *measures* have been also tried (e.g., the normalized/unnormalized mutual information or the symmetric uncertainty). Nonetheless, since we have obtained similar trends (and slightly worse performance), those are not included in this study for the sake of brevity.

3.4 Classification Algorithms

In this sub-section we review five supervised classification algorithms successfully employed in several works tackling TC of anonymous traffic [17], [31], [32], [34], that are applied to the scenario investigated in this work: (i) Naïve Bayes, (ii) Multinomial Naïve Bayes, (iii) Bayesian Networks, (iv) C4.5, and (v) Random Forest.

Naïve Bayes (NB)

The NB is a simple probabilistic classifier that assumes class conditional *independence* of the features, being not the case

[5] More precisely, the IAT range is divided in 91 bins with the following ranges: bins $0-39$ covering $[0, 200)\,\mathrm{ms}$ with $5\,\mathrm{ms}$ width, bins $40-59$ covering $[200, 400)\,\mathrm{ms}$ with $10\,\mathrm{ms}$ width, bins $60-89$ covering $[400, 1000)\,\mathrm{ms}$ with $20\,\mathrm{ms}$ width and bin 90 for IAT values higher than $1\,\mathrm{s}$ [38].

[6] Only in this case, the `pl` acronym is referred to the Payload Length, in accordance to Tranalyzer2 nomenclature [38].

for real-world problems, but working well in practice and leading to reduced complexity.

More specifically, NB evaluates the probability that an unlabeled test instance $\boldsymbol{f}_T$ belongs to each class c_i, i.e. the posterior probability $P(c_i|\boldsymbol{f}_T)$, through the Bayes' theorem and returns the label corresponding to the maximum posterior among the classes, that is $P(c_i|\boldsymbol{f}_T) \propto P(c_i)\prod_{m=1}^{M} P(f_{T,m}|c_i)$. Here "$\propto$" means proportionality and $P(c_i)$ denotes the (prior) probability of class c_i (estimated from the training set). On the other hand, each distribution $P(f_m|c_i)$ is estimated by resorting to a PMF when the feature is *categorical*, whereas common alternatives for *numerical* features include: (i) Moment Matching to a Gaussian PDF (NB), (ii) Supervised Discretization (NB_SD), and (iii) Kernel-based Density Estimation (NB_KDE) [42].

In this work, each NB classifier will be fed either with the *first* set of features (flow-based statistics) or with the *fourth* set of features (sequence of (Payload Length, IAT) of the first K packets).

Multinomial Naïve Bayes (MNB)

The MNB classifier adopts sample histograms as a different set of features. Specifically, the MNB classifier treats the f_ms as frequencies of a certain value of a categorical random variable and compares the sample histogram of each test instance with the aggregated histogram of all training instances per class. Then, the evaluation of the conditional PMF $P(\boldsymbol{f}_T|c_i)$ is proportional to $\prod_{m=1}^{M}(\rho_m)^{f_{T,m}}$, where ρ_m denotes the probability of sampling the m^{th} feature. In particular, we will employ a variant of MNB classifier, adopting *term frequency transformation* with *cosine normalization*, as successfully exploited in Herrmann et al. [24] for website fingerprinting in anonymous networks.[7]

As introduced earlier, since the MNB classifier works on a set of features in the form of histograms, the *second* (PL histogram) and *third* (joint PL-IAT histogram) set of features described in Sec. 3.2 will be employed.

Bayesian Networks (BNs)

BNs are graphical representations which model dependence relationships between features and classes [43], collectively represented as the set of random variables $\boldsymbol{U} \triangleq \{f_1, \ldots, f_M, C\} = \begin{bmatrix} U_1 & \cdots & U_{M+1} \end{bmatrix}^T$. Unlike the NB classifier, they are *not* based on the conditional independence assumption for the features.

Formally, a BN for $\boldsymbol{U}$ is a pair $\mathcal{B} \triangleq \langle \mathcal{G}, \Theta \rangle$, which is learned during the training phase. The first component ($\mathcal{G}$) is a *Directed Acyclic Graph* that encodes a joint probability distribution over $\boldsymbol{U}$, where each *vertex* represents a random variable among $U_1, \ldots, U_{M+1}$ and *edges* represent their dependencies. The second component (Θ) represents the set of parameters modeling the BN, uniquely determining the local conditional distributions associated to the BN, which allow to encode the joint distribution $P_{\mathcal{B}}(f_1, \ldots, f_M, C)$. Finally, during the testing phase, for each instance $\boldsymbol{f}_T$, the BN classifier returns the label $\hat{c} \triangleq \arg\max_{c_i \in \Omega} P_{\mathcal{B}}(c_i|\boldsymbol{f}_T)$, based on Bayes' theorem.

In this study, we will either consider a BN classifier with (default) *K2 search* (BN_K2) for structure learning, or impose the network to have a *tree-augmented form* (BN_TAN) where the tree is formed by calculating the maximum weight spanning tree. Each BN classifier will be fed either with the *first* set of features (flow-based statistics) or with the *fourth* set of features (sequence of (Payload Length, IAT) of the first K packets).

C4.5

C4.5 is an algorithm employed to generate a decision tree used (mainly) for classification purposes [44], based on the concept of *entropy* of a distribution [37]. The training algorithm obviates to the NP-hardness of optimal tree search by means of a *greedy* procedure, based on a top-down recursive construction, with all the data of the training set in the root as the init. Then, instances are partitioned recursively based on the chosen feature whose values most effectively split so as to maximize a purity[8] measure in the data, such as the "gain ratio", that avoids bias toward features with a larger support [44]. Thus, the splitting criterion is triggered by the feature ensuring the highest gain ratio (i.e. purity). C4.5 recurs on the smaller sublists, until the following termination criteria are met: (i) all the instances in the list belong to the same class (a leaf node is here created with a label associated to that class); (ii) there are no remaining features for further partitioning (in such case, each leaf is labeled with the majority class in the subset); (iii) there are no examples left.[9]

In our analysis, C4.5 will be fed either with the *first* feature set (flow-based statistics) or with the *fourth* set of features (sequence of (Payload Length, IAT) of the first K packets).

Random Forest (RF)

RF is a classification method based on an ensemble of B several decision trees (the number of trees is a free parameter tuned by cross-validation or via the "out-of-bag" error), built at training time exploiting the ideas of "bootstrap aggregating" (bagging) and random-feature selection to mitigate over-fitting [45]. Specifically, during the training phase, each decision tree in the RF classifier is grown based on a bootstrap (i.e. a uniformly random sampling procedure with replacement) sample set of the training data available. RF adds to the above scheme a modified tree learning algorithm named "feature bagging" (to further reduce overfitting) that selects, at each candidate split in the learning process, only a random subset (whose size is another free tunable parameter) of the features. Finally, after training, decision on testing samples can be made by taking the majority vote or soft combination of the responses of B trees.

In this work, RF will be fed either with the *first* feature set (flow-based statistics) or with the *fourth* set of features (sequence of (Payload Length, IAT) of the first K packets).

[7] It is worth noting that the numerical analysis reported in Sec. 4 has also underlined highest performance of the considered variant with respect to the others analyzed in [24].

[8] A subset of data is said "pure" if all instances belong to the same class.

[9] Refinements introduced by C4.5 to reduce over-fitting include (a) *pre-pruning* (i.e. stop growing a branch when information becomes unreliable) and (b) *post-pruning* (i.e. growing a decision tree that correctly classifies all training data and then simplify it later by replacing some nodes with leaves), with the latter preferred in practice.

4 Experimental Results

This section reports details about the Anon17 dataset and the pre-processing operations carried out on it (Sec. 4.1), introduces the performance metrics employed for evaluation (Sec. 4.2) and shows the results of the (anonymous) TC investigations performed (Sec. 4.3).

4.1 Dataset Description

Anon17 was collected at the Network Information Management and Security Lab [46] between 2014 and 2017 in a *real* network environment. The dataset is labeled based on the information available on the anonymity services themselves (e.g., IP addresses of the Tor nodes) without relying on any application classification tool. The data are stored in ARFF format used in the data mining software tool *Weka* [42] and report features (discussed in Sec. 3.2) either on a per-flow basis or pertaining to the IAT/Payload-Length sequence of the first K packets of each flow. Unfortunately, the unavailability of the *whole sequence* of payload lengths and IATs prevents additional interesting analyses, such as the evaluation of the impact of packet sampling on the classification accuracy [47]. We refer to [19] for further details on Anon17 dataset.

Given the available dataset, we tackle classification of anonymity networks (as well as traffic types and applications) by making the assumption that we are in presence of anonymous traffic only, based on a two-fold motivation. First, this refers to an application context in which a traffic classifier tool has been able to provide accurate screening of clear and standard encrypted traffic, as demonstrated, for example by Barker et al. [14] and more recently by Rao et al. [21] for Tor network. Once the instances of anonymous traffic have been labeled, the aim of the proposed approach is to assess potential discrimination of different anonymity services within such instances. Therefore, the current study represents a further step toward the development of a hierarchical classification framework for traffic analysis of clear/encrypted/anonymous data. Secondly, the results of the present analysis can be intended as an upper bound on the classification performance of anonymity networks in the case of an *open-world* assumption. Indeed, a negative answer to our question (i.e. an unsatisfactory performance in classifying anonymous traffic only) would lead to the conclusion that anonymous traffic, even though perfectly screened from the remaining traffic bulk, would still remain an unobservable black-box to an eavesdropping user. Our results will show that this is not the case, and confirm that there is room for classification of ATs in an open-world assumption.

As explained in Sec. 1, our analysis of ATs is conducted at different levels of granularity, that is *Anonymous Network Level* (L1), *Traffic Type Level* (L2), and *Application Level* (L3). More specifically, we try to ascertain the granularity of the identifiability of these tools by performing classification. The hierarchical categorization of L1, L2, and L3 is reported in detail in Tab. 1. The total number of applications (L3 classes) identified for each anonymous network (three L1 classes) and traffic type (seven L2 classes) is 21 and constitutes the finest level of our TC task. Specifically, (normal) Tor Traffic includes the circuit establishment and the user activities, whereas Tor Apps refer to flows running three applications on the Tor network (i.e. L3 classes: Browsing, Video streaming, and Torrent file sharing). On the other hand, Tor PTs contain flows for five different obfuscation techniques (i.e. L3 classes: Flash proxy, FTE, Meek, Obfs3, and Scramble suit). Flows belonging to L2 I2P Apps Tunnels with other Tunnels are collected by running three applications (L3 classes) on the I2P network: I2Psnark (file sharing), jIRCii (Internet Relay Chat), and Eepsites (websites browsing). The difference between 0% and 80% bandwidth is in the amount of sharing rate of the user bandwidth. I2P Apps contain traffic flows for the same three applications. However, in the latter case, management tunnels belong to separate L3 classes (i.e. Exploratory Tunnels and Participating Tunnels). Lastly, JonDonym dataset contains flows for the whole free mixes on the JonDonym network.

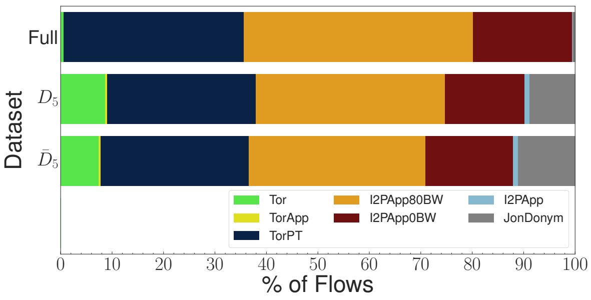


Figure 1: Down-sampling of Anon17 dataset: upper barplot (original *Full* dataset), middle barplot (down-sampling to 5%, D_5), lower barplot (removal of "all-zero-payload" flows, $\bar{D}_5$).

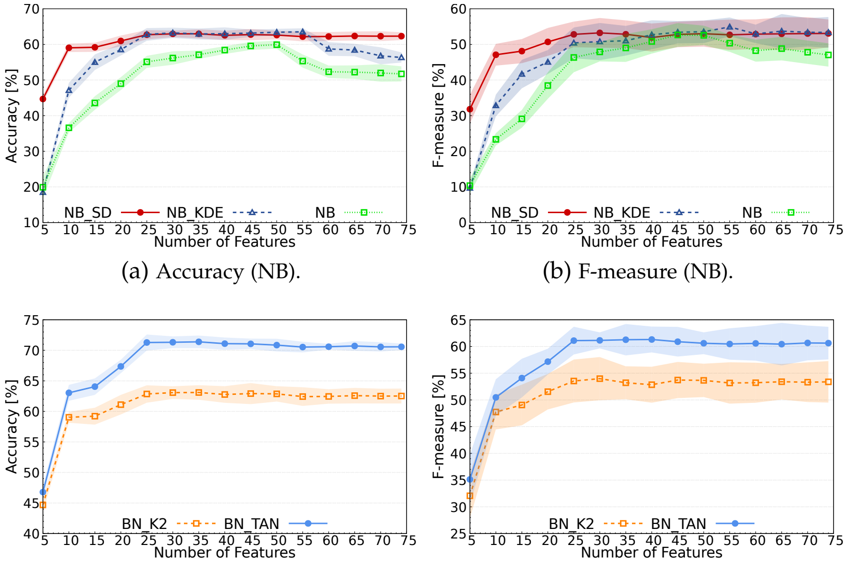
Anon17 exhibits a (majority) class imbalance problem, as shown by the total number of samples in Tab. 1 (and graphically depicted in the top bar of Fig. 1). To cope with it, we have randomly down-sampled[10] (without replacement) by applying a pre-processing filter[11] to the instances of the following highly-populated traffic types (so as to keep their number comparable with the others): (i) Tor Pluggable Transports, (ii) I2P Apps Tunnels with other Tunnels [0% BW], and (iii) I2P Apps Tunnels with other Tunnels [80% BW]. The considered filter also preserves the proportions of the contained L3 applications.

In this study, we will consider a configuration corresponding to the down-sampling to 5% of the original dataset of each traffic type set.[12] Fig. 1 shows the percentage of flows labeled with different traffic types after performing the aforementioned down-sampling (D_5). We underline that we have chosen to down-sample the *whole* dataset, as opposed to the sole training set, since the latter choice would have biased the overall accuracy measure (evaluated from the test set) toward the performance of the majority classes.

[10] Over-sampling methods (e.g., SMOTE, ROSE, etc.) are not considered here as Anon17 dataset does not show a minority class imbalance problem.

[11] Adopted filter is implemented in the Weka environment by means of `weka.filters.supervised.instance.Resample` Java class.

[12] We recall that in our previous work [20], we have considered two down-sampling configurations (corresponding to 5% and 10%) of each traffic type set, showing a non-relevant difference in performance between them. However, aiming at a fairer investigation, here we have opted for the more balanced configuration.



(a) Accuracy (NB).

(b) F-measure (NB).

(c) Accuracy (BN).

(d) F-measure (BN).

Figure 2: Accuracy and F-measure of NB (a-b) and BN (c-d) classifiers for different subsets of features (from 5 to 74 with increments of 5) for L3 (Application) level. Average on 10-folds and corresponding $\pm 3\sigma$ confidence interval are shown.

Finally, to perform TC of sole *informative* flows, we have discarded from D_5 all the instances containing only zero-payload packets. The latter filtering procedure has been conducted implicitly by the inspection of the field `maxPktSz`, as Anon17 does not provide the complete sequence of packet lengths (see [19]). The bottom barplot in Fig. 1 shows the final resulting dataset (here denoted with $\bar{D}_5$) considered in the following flow-based TC. On the other hand, for the early-TC, a different filtering procedure has been pursued, as explained in detail later in Sec. 4.3.

4.2 Performance Measures

Our comparison will be based on the following performance measures [41]: *overall accuracy*, *precision*, and *recall*. Since these last two metrics are defined on a per-class basis, their averaged (macro) versions (denoted with "prec" and "rec", respectively) will be employed as synthetic measures and, to account for the effects of both in a concise fashion, we will consider the *F-measure* ($\mathrm{F} \triangleq (2 \cdot \mathrm{prec} \cdot \mathrm{rec})/(\mathrm{prec} + \mathrm{rec})$). Moreover, to provide a complete performance "picture" of each classifier, we will also show their confusion matrices so as to identify the most frequent misclassification patterns. Clearly, a higher concentration of the confusion matrix toward the diagonal (where predicted class equals the actual one) implies better performance of the generic classifier.

Finally, for each considered analysis, our evaluation will be based on a (stratified) 10-fold cross-validation.[13] Indeed, K-fold validation represents a stable performance evaluation framework as it produces less variance in the results. For completeness, we will report both the mean and the variance (in the form of a $\pm 3\sigma$ interval, corresponding to 99.7% confidence under a Gaussian assumption) of each performance measure as a result of the evaluation on the ten different folds.

4.3 Classification Results

In this section, we show results pertaining to several sets of experiments aimed at investigating TC of ATs. More specifically, the first part of the numerical analysis is focused on flow-based TC (even in the case of "finer" histogram-based features), while the second part pertains to early TC. In both cases, feature relevance is also assessed for all the classifiers considered. The section ends with a finer-grained analysis of the pattern errors of the two classification "philosophies".

Flow-based classification

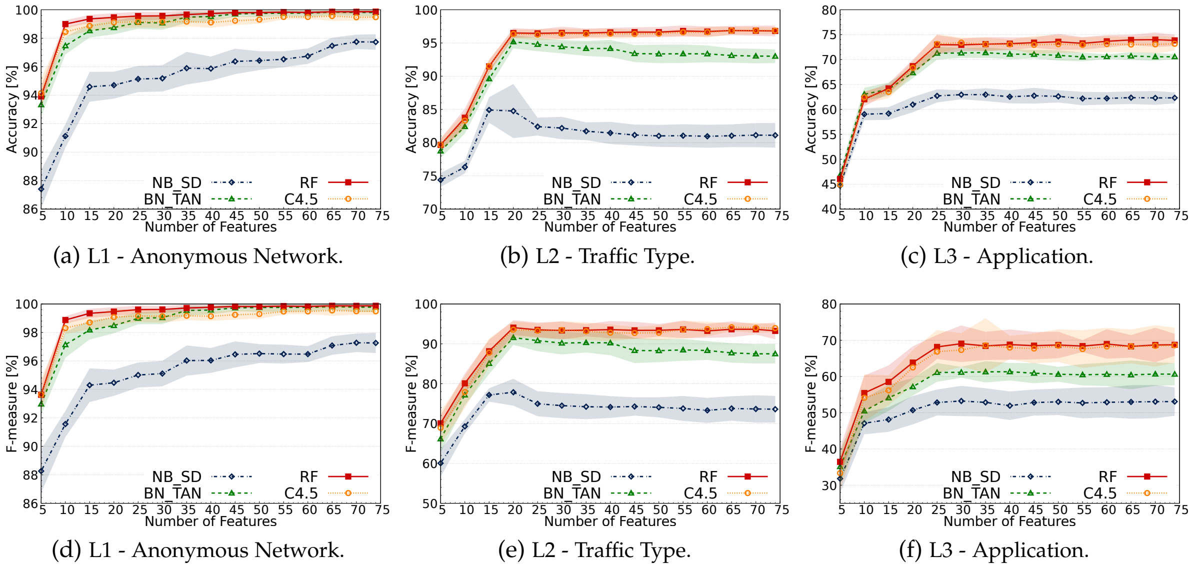
First, we investigate flow-based TC based on the *first* feature set (i.e. comprising summarizing flow-based statistics) described in Sec. 3.2. However, before proceeding with a rigorous comparison of the classifiers here considered, we first focus on relative performance evaluation of different variants of NB and BN considered in this paper (cf. Sec. 3.4).

To this end, Fig. 2 shows the accuracy and F-measure of `NB`, `NB_SD`, and `NB_KDE` (resp. `BN_K2` and `BN_TAN`) in top (resp. bottom) plots. Note that the results pertain to L3, being the *hardest* classification task, but similar trends have been observed also for the other (two) levels. Furthermore, the performance has been evaluated by training/testing the classifiers with a varying subset of features, ranked (in decreasing importance) by resorting to the Pearson's correlation (cf. Sec. 3.2) so as to draw general conclusions.[14]

From inspection of the figure, it is apparent that both `NB_SD` and `NB_KDE` outperform `NB` over all the range of feature subsets, with `NB_SD` achieving higher performance even in the case of a *smaller* set. Similarly, `BN_TAN` *significantly* outperforms `BN_K2`. The former result can be explained as density estimation of each feature, either in a discretized (`NB_SD`) or "kernelized" (`NB_KDE`) fashion, is beneficial since the Gaussian assumption represents an

[13] Although preliminary results in [20] have shown negligible difference in performance with respect to a random training-test set splitting, the present evaluation is intended to highlight statistically-significant trends.

[14] Using the Weka filter `CorrelationAttributeEval`, employed in conjunction with a `Ranker` utility which allows obtaining the top $M_\star$ (most informative) features, with $M_\star$ as input parameter.



(a) L1 - Anonymous Network.

(b) L2 - Traffic Type.

(c) L3 - Application.

(d) L1 - Anonymous Network.

(e) L2 - Traffic Type.

(f) L3 - Application.

Figure 3: Accuracy (a-c) and F-measure (d-f) of flow-based classifiers for different subsets of features (from 5 to 74 with increments of 5) for each classification level. Average on 10-folds and corresponding $\pm 3\sigma$ confidence interval are shown.

Table 2: Best overall accuracy and macro F-measure for dataset $\bar{D}_5$ obtained with different [optimal number of features] employed. Highlighted values: **maximum accuracy** and maximum F-measure for each level.

Flow-Based Classifier	Metric	L1	L2	L3
NB_SD	*Accuracy*	97.74% [70]	84.92% [15]	62.97% [35]
	F-measure	97.28% [70]	77.86% [20]	53.23% [30]
BN_TAN	*Accuracy*	99.83% [65]	95.18% [20]	71.39% [35]
	F-measure	99.81% [65]	91.57% [20]	61.31% [40]
C4.5	*Accuracy*	99.56% [65]	**96.87%** [70]	73.41% [30]
	F-measure	99.55% [65]	94.17% [65]	68.60% [74]
RF	*Accuracy*	**99.87%** [74]	**96.87%** [65]	**73.99%** [70]
	F-measure	99.87% [74]	94.06% [20]	69.05% [30]

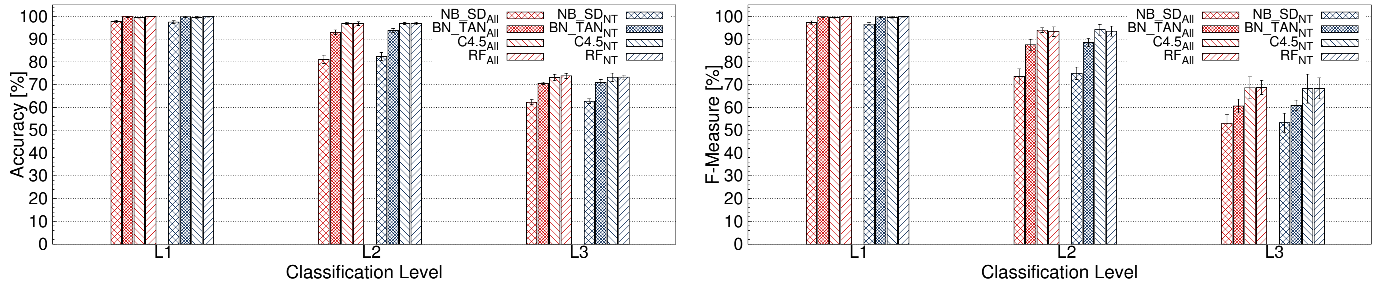
overly simplified assumption, whereas in the latter case it is apparent that constraining BN structure learning to a tree-augmented form (as opposed to a greedy sub-optimal learning) provides improved generalization capabilities.

Thus, in the remainder of this section, only the variants NB_SD and BN_TAN will be considered in our comparison, being the best-performing Naïve Bayes and Bayesian Networks classifiers variants observed, respectively.

Then, with the aim of selecting the best subset (viz. an optimized number) of features and provide a comparison of the supervised techniques considered, in Fig. 3 we show the performance of all the classifiers described in Sec. 3.4 (except for MNB, whose performance relies on histogram-based features and will be thus discussed later) when varying the (ranked) subset of features for both training and test sets. As apparent from the results, all the classifiers obtain excellent results in L1 classification, i.e. all achieving both $> 95\%$ accuracy and F-measure when approximately the top 25 features are employed. On the other hand, performance metrics generally degrade with the increasing granularity of the classification task (i.e. moving from L1 to L3). This intuitive trend can be attributed to the increasing difficulty of the classification task being tackled. Indeed, the discrimination of anonymous traffic at L3 is harder than trying to discern merely the anonymity network. Interestingly, the degradation level varies with the classifier and it is observed to be milder for C4.5 and RF, whereas it is higher for NB_SD. This finding can be explained as the conditional independence assumption of the features for NB_SD is *limiting* when tackling harder classification tasks (i.e. L3). Overall, from figures inspection, the performance of all classifiers (approximately) reaches a steady value around the top 30 features (as already observed for NB_SD and BN_TAN).

Collectively, the highest performance (with an optimized number of features) is obtained by RF and C4.5, corresponding to 99.87% (resp. 99.87%), 96.87% (resp. 94.17%) and 73.99% (resp. 69.05%) at L1, L2, and L3, respectively, in terms of accuracy (resp. F-measure), as shown by the summarizing results reported in Tab. 2.

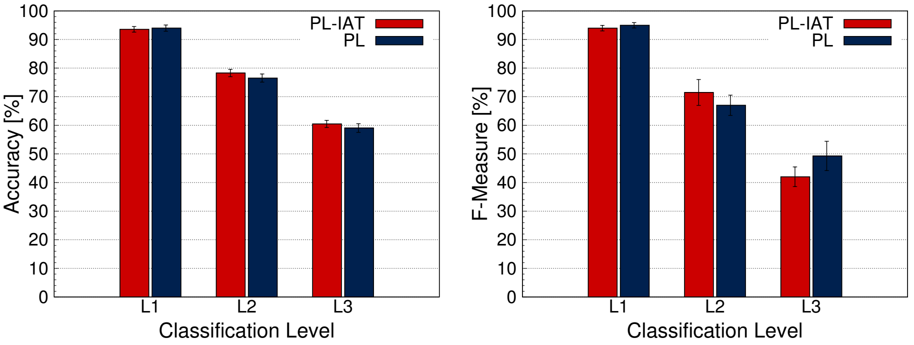
Additionally, to investigate whether the most relevant features are mostly related to packet lengths or are also *time-related*, in Fig. 4 we compare the accuracy (Fig. 4a) and the F-measure (Fig. 4b) of the same pool of classifiers when (*a*) *all* the (74) features of the *first* feature set are employed and (*b*) *Not Time-related* (NT) features (52) are considered. The analysis is conducted at the three different levels of classification granularity allowed by Anon17. From inspection of both figures, it is apparent that the contribution of



(a) Accuracy.

(b) F-measure.

Figure 4: Accuracy (a) and F-measure (b) of considered flow-based classifiers fed with the full set of (All) features (74) and Non Time-related (NT) features only (52). Average on 10-folds and corresponding $\pm 3\sigma$ confidence interval are shown.



(a) Accuracy.

(b) F-measure.

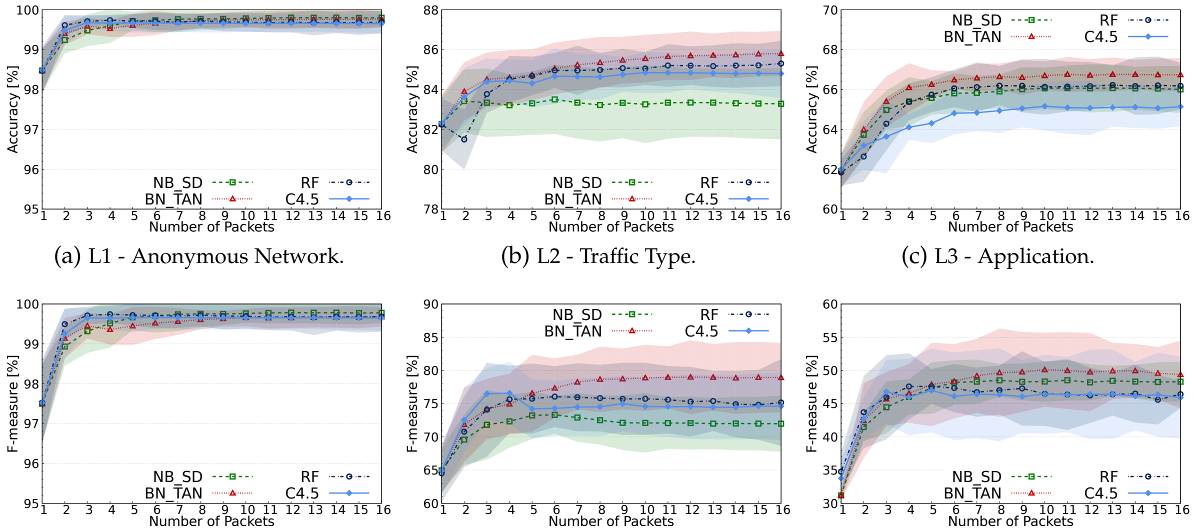
Figure 5: Accuracy (a) and F-measure (b) of MNB classifier [24] leveraging PL and PL-IAT histograms as features. Average on 10-folds and corresponding $\pm 3\sigma$ confidence interval are shown.

Table 3: Best overall accuracy and macro F-measure for early traffic classification obtained with different [optimal value of the first N packets] employed. Highlighted values: **maximum accuracy** and maximum F-measure for each level.

Early Classifier	Metric	L1	L2	L3
NB_SD	*Accuracy*	**99.80%** [14]	83.50% [06]	66.07% [11]
	F-measure	99.78% [12]	73.31% [06]	48.52% [11]
BN_TAN	*Accuracy*	99.76% [16]	**85.80%** [16]	**66.76%** [11]
	F-measure	99.68% [16]	78.96% [13]	50.10% [10]
C4.5	*Accuracy*	99.68% [03]	84.85% [10]	65.16% [10]
	F-measure	99.66% [06]	76.58% [04]	46.98% [05]
RF	*Accuracy*	99.74% [04]	85.30% [16]	66.22% [13]
	F-measure	99.74% [04]	76.05% [06]	47.59% [05]

time-related features (within first dataset) to classification performance is only *marginal*. This observation applies to all the levels of granularity and to all the classifiers being employed in our investigation, with a maximum improvement of just $+1.21\%$ (resp. $+1.40\%$) in terms of accuracy (resp. F-measure) obtained for NB_SD at L2 using NT features.

We now focus on investigating whether (i) finer-grained features, such as histograms, would improve performance and (ii) whether these finer-grained features would require time-related features. We recall that the appeal of histogram-based features has been highlighted by different works on TC [7], [48]. Based on this reason, in Fig. 4 we report the performance (in terms of both accuracy and F-measure) of MNB in conjunction with the use of the *second* (PL histogram) and *third* (joint Payload Length-IAT histogram) feature sets described in Sec. 3.2. Similar to the previous analyses, the performance is evaluated at the three levels of granularity for the sake of a complete comparison. First, it is apparent that considering histogram-based features *does not improve* classification performance, as evident from comparison of Fig. 3 and Fig. 4. The lack of improved performance may be due to a two-fold reason: (i) in MNB case, features pertaining to IP/TCP headers and number of connections are not taken into consideration and (ii) the histogram discretization provided by Anon17 may be not adequate for developing an accurate fingerprint. Interestingly, time-related features do not improve appreciably classification performance at first two levels. This trend is similar to that observed for classifiers fed with the first feature set, see Fig. 4. The only exception is represented by the increase of F-measure at L3



(a) L1 - Anonymous Network.

(b) L2 - Traffic Type.

(c) L3 - Application.

(d) L1 - Anonymous Network.

(e) L2 - Traffic Type.

(f) L3 - Application.

Figure 6: Accuracy (a-c) and F-measure (d-f) considering the first N non-zero payload packets in terms of Payload Length & IATs (from 1 to 16) for each classification level.

($+6.85\%$) when the PL histogram feature set. This trend may be attributed to the harder classification task to be solved.

Early Traffic Classification

We now investigate the possibility of performing "early" classification of anonymous traffic. To this end, we consider the *fourth* set of features extracted by Anon17, i.e. the sequence of the first $K = 20$ Payload Lengths & IATs (as described in Sec. 3.2).

For the present analysis, we remove from dataset D_5 (see Fig. 1) all the instances whose first $K = 20$ packets have *all zero payload*. We recall that these correspond to a superset of those removed in the case of flow-based TC (i.e. the resulting dataset $\bar{D}_5$), as we are removing *also* the instances with some payload exchange after the first 20 (zero-payload) packets. The reason for a different filtering procedure is to avoid submitting "non-informative" (referring to the first $K = 20$ packets) instances to the considered classifiers.

This allows us training the classifiers considered in this paper[15] with the Payload Lengths & IATs of the first $N < K$ (non-zero payload) packets so as to assess the possibility of accurately classifying ATs with a few packets, as shown for the Internet traffic scenario [39], [40]. We underline that, in case of flows having less than N payload-carrying packets, the remaining packets are treated as *missing values* [41] by the supervised classifiers.

For this reason, Fig. 6 reports the accuracy and F-measure at the three considered levels when the classifiers are trained/tested on the Payload Lengths & IATs of the first $N = 1, \ldots, 16$ (non-zero payload) packets, so as to assess to which degree ATs (and services running within them) can be identified "on-the-fly". As apparent from the results shown, $5 \div 7$ packets are usually *sufficient* to (approximately) allow the classifiers to achieve their highest performance. This consideration applies to each of the three levels of granularity considered. Remarkably, the reported results agree qualitatively with those in [39], [40] pertaining to non-anonymous traffic. Additionally, we remark that a similar analysis (not shown for brevity) has been conducted by feeding the classifiers with Payload Lengths only (without considering IATs). In the latter case, results have shown almost equal performance at the first level, whereas a rough 5% drop has been observed for F-measure at L2 and L3.

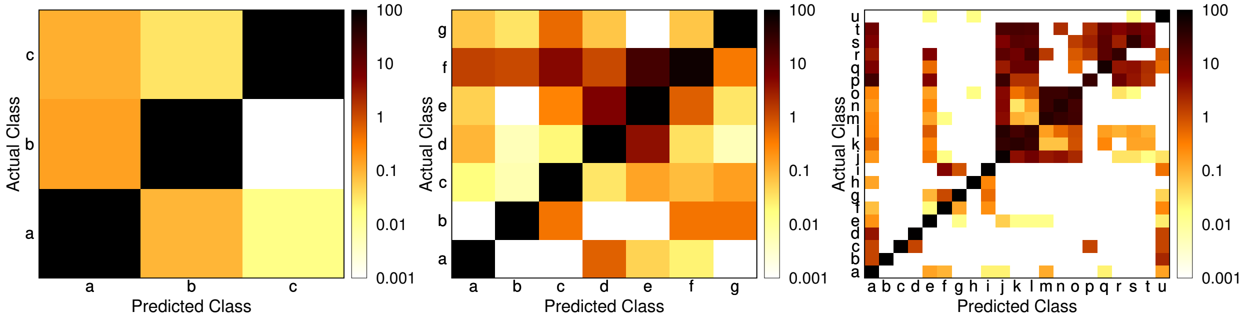
In detail, performance at L1 is extremely satisfactory ($\geq 99\%$ with only 3 payload-carrying packets), whereas there is a significant degradation with respect to flow-based TC (see Fig. 3) in the case of L2 and L3. Specifically, as summarized in Tab. 3, the highest accuracy (resp. F-measure) achieved at L2 is 85.80% (resp. 78.96%), as opposed to 96.87% (resp. 94.17%) in the case of flow-based TC. Similarly, at L3 the highest accuracy (resp. F-measure) is 66.76% (resp. 50.10%), as opposed to 73.99% (resp. 69.05%). Interestingly, the highest performance for early-TC is achieved by *Bayesian methods* (i.e. NB_SD and BN_TAN) as opposed to decision tree-based classifiers for the flow-based approach.

The observed performance loss corresponds to the price paid for trying to classify *with only a few packets at an extremely thin level of detail*. The degradation at L2/L3 underlines the need for further investigations on early-TC, and the need for developing *deeper* (viz. structured) representations of the sequence of the first few packets.

Fine-grained Performance

Since classification at increasing level of granularity reveals to be a challenging task (but also the most interesting from a user's privacy perspective), henceforth we analyze the confusion matrices of flow-based and "early" classifiers, shown in Figs. 7 and 8, respectively, so as to highlight interesting error patterns. We recall that for these matrices the higher the concentration toward the main diagonal, the better the overall performance. More specifically, for each classification level, we report the confusion matrices of the optimal (in terms of *F-measure*) combination of classifier and

[15] Similar results to flow-based TC have been observed when comparing NB and BN variants. Therefore, in what follows, we will again report performance of the sole NB_SD and BN_KDE, respectively.

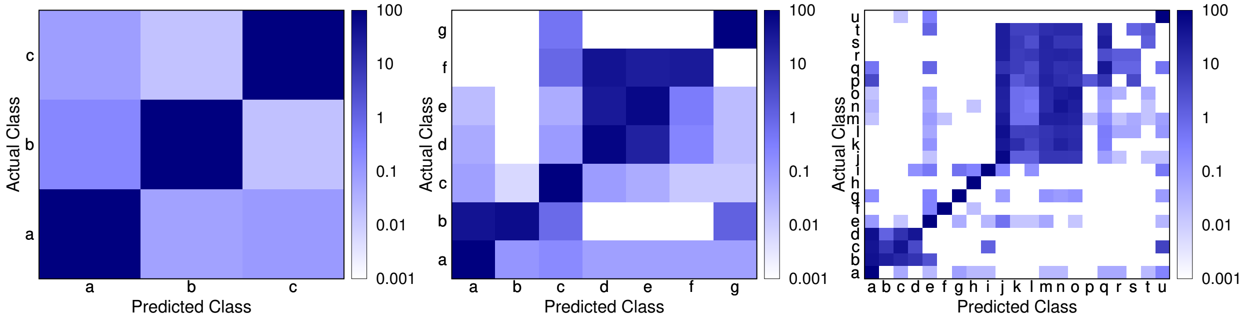


(a) L1 - Anonymous Network.

(b) L2 - Traffic Type.

(c) L3 - Application.

Figure 7: Confusion matrices (percentage accuracy and log scale) of the best flow-based classifier at L1 (`RF` with 74 features), L2 (`C4.5` with 65 features), and L3 (`RF` with 30 features).



(a) L1 - Anonymous Network.

(b) L2 - Traffic Type.

(c) L3 - Application.

Figure 8: Confusion matrices (percentage accuracy and log scale) of the best "early" classifier at L1 (`NB_SD` with 12 packets), L2 (`BN_TAN` with 13 packets), and L3 (`BN_TAN` with 10 packets).

number of employed features (or of the first N packets), based on previous analyses.

First, by looking at L1 performance, error patterns look scattered for both approaches. However, in case of flow-based classification, it is interesting to note that `I2P` is never misclassified with `JonDonym`, as opposed to the corresponding best early classifier. Secondly, by looking at L2 confusion matrices, it is apparent that the classifiers based on both approaches (with different quantitative outcomes) present error patterns which almost entirely lead to a misclassification of the traffic type *within the same* anonymous network. Nonetheless, it is worth noticing that while this applies to `I2P` traffic types ("**d**", "**e**", and "**f**") for both approaches, misclassification of `Tor` traffic types ("**a**", "**b**", and "**c**") is *only observed in early classifiers*. In the latter case, the error-patterns are mostly due to misclassification of `Tor Apps` ("**b**") with `Normal Tor Traffic` ("**a**").

Moving to a further detail, we discuss in what follows L3 confusion matrices. From inspection of the results at the finest level of ATs granularity, it can be concluded that the best flow-based classifier has the same high discrimination capability when trying to discern applications running within `Tor`. Such result qualitatively agrees with [31], where a high accuracy is achieved in classifying Tor applications (i.e. browsing, streaming, and BitTorrent, corresponding to "**b**", "**c**", and "**d**", respectively) via *flow-based* classification based on Tranalyzer2. The results shown in Fig. 7 are compatible with the above work (interestingly, also in our *harder* classification task, tree-based classifiers performed the best) and further show that Tor app can be hardly misclassified with apps from other anonymous networks (such as I2P).

On the other hand, the best early classifier achieves low discrimination power for applications contained within the traffic types `Normal Tor Traffic` and `Tor Apps`. An opposite trend is instead apparent for the case of `Tor Pluggable Transports`, which are *easily discerned* with *both* flow-based and early classifiers. Differently, in [32] classification of Tor PTs was demonstrated successful in comparison to background traffic, achieving with a `C4.5` classifier a 97% accuracy with a 10-fold validation. Here, we assume that the background traffic has been already screened out, therefore results obtained in the two cases cannot be directly compared. However, the results in this section confirm the unique fingerprint generated by PTs when trying to obfuscate Tor traffic.

Finally, it is apparent how performance at L3 of both approaches are limited superiorly mostly by the error-prone recognition of applications running within `I2P`, representing (for the considered dataset), the least discernible AT in terms of its *carried services and applications*. Indeed, the best flow-based classifier (being the approach leading to the highest performance observed), is not able to discern apps within `I2P Apps Tunnels with other Tunnels [0% Bandwidth]` and `I2P Apps Tunnels with other Tunnels [80% Bandwidth]`, as apparent from the two clusters within the confusion matrix in Fig. 7. More in detail, in [22] the effect of bandwidth participation on I2P is investigated, showing higher application profiling with less bandwidth sharing. This trend qualitatively agrees with the discussed results, where the best performing flow-based classifier is shown to be most prone to misclassification of `I2P Apps Tunnels with other Tunnels [80% bandwidth]` (i.e. "**m**", "**n**", and "**o**"). Therefore, the results of the present study agree with the literature. In addition, we observe that another (although less problematic) error-cluster for the flow-based classifier is also apparent for all the applications belonging to the traffic type `I2P Apps`.

Finally, we recall that the present work provides a more comprehensive study of traffic classification and identification of different ATs at a varying degree of granularity, underlining the narrowness of the above studies (i.e. focusing on a particular AT or a specific aspect of it).

5 Conclusions

This paper tackled TC of ATs, specifically Tor, I2P, and JonDonym, reasoning on which degree they can be told apart, considering different granularities (the *anonymity network* adopted, the *traffic type* tunneled in the network, and the *application* category generating such traffic). The analysis has been carried on the public dataset Anon17, processed with random down-sampling (to cope with its strong class imbalance) and by filtering out non-informative (all-zero-payload) flows. Different ML classifiers (Naïve Bayes, Multinomial Naïve Bayes, Bayesian Networks, C4.5, and Random Forest) have been applied to the processed dataset, based on different feature sets and by assessing the optimal number of features to consider, by means of feature selection.

Our analysis shows that Tor, I2P, and JonDonym anonymous networks can be hardly mistaken from each other. Indeed, results underline that all considered classifiers obtain extremely satisfactory performance (at least 97% accuracy when the number of features is chosen wisely) in discriminating the anonymity networks present in Anon17, in both cases of *flow-based* and *early* classifiers. Furthermore, it is shown that digging down in the specific type of traffic tunneled, and the specific type of application generating such traffic, is possible with up to 73.99% accuracy and 69.05% F-measure with a flow-based `RF`. Performance of "early" classifiers at application level is lower (66.76% and 50.10% in terms of accuracy and F-measure, respectively, with the early `BN_TAN`) and more structured feature representations should be conceived. Thanks to the public availability of Anon17 dataset and the detailed description of methods and (open-source) tools, our results are easily repeatable, comparable, and extensible by the research community.

As future work we will investigate (i) hierarchical classification, (ii) comparison with other public labeled datasets (possibly also in an open-world assumption), should they become available, (iii) development of classifier fusion techniques for anonymous TC, and (iv) implementation of features and classifiers in the open-source TC platform TIE [49] to allow researchers to evaluate them on live traffic traces.

Acknowledgments

This work is partially funded by art. 11 DM 593/2000 for NM2 s.r.l. (Italy). The authors would like to thank the Faculty of Computer Science of Dalhousie University, Halifax, Canada, for publicly releasing Anon17 dataset.

References

[1] P. Syverson, R. Dingledine, and N. Mathewson, "Tor: the second generation onion router," in *USENIX 13th Security Symposium (SSYM)*, 2004.

[2] "The Invisible Internet Project (I2P)," [Online] https://geti2p.net/en/, Jul. 2017.

[3] "Project: AN.ON - Anonymity," [Online] http://anon.inf.tu-dresden.de/index_en.html, Jul. 2017.

[4] G. Aceto and A. Pescapé, "Internet censorship detection: A survey," *Computer Networks*, vol. 83, pp. 381–421, 2015.

[5] A. Dainotti, A. Pescapé, and G. Ventre, "Worm traffic analysis and characterization," in *IEEE International Conference on Communications (ICC)*, 2007, pp. 1435–1442.

[6] A. Dainotti, A. Pescapé, and K. C. Claffy, "Issues and future directions in traffic classification," *IEEE Network*, vol. 26, no. 1, 2012.

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

[8] N. Cascarano, A. Este, F. Gringoli, F. Risso, and L. Salgarelli, "An experimental evaluation of the computational cost of a DPI traffic classifier," in *IEEE Global Communications Conference (GLOBECOM)*, 2009, pp. 1–8.

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

[10] 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.

[11] A. Dainotti, A. Pescapé, and H.-c. Kim, "Traffic classification through joint distributions of packet-level statistics," in *IEEE Global Communications Conference (GLOBECOM)*, 2011, pp. 1–6.

[12] A. Dainotti, F. Gargiulo, L. I. Kuncheva, A. Pescapè, and C. Sansone, "Identification of traffic flows hiding behind TCP port 80," in *IEEE International Conference on Communications (ICC)*, 2010, pp. 1–6.

[13] K. Bauer, M. Sherr, D. McCoy, and D. Grunwald, "ExperimenTor: a testbed for safe and realistic Tor experimentation," in *USENIX Workshop on Cyber Security Experimentation and Test (CSET)*, 2011.

[14] J. Barker, P. Hannay, and P. Szewczyk, "Using traffic analysis to identify the second generation onion router," in *9th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC)*, 2011, pp. 72–78.

[15] A. Chaabane, P. Manils, and M. A. Kaafar, "Digging into anonymous traffic: A deep analysis of the Tor anonymizing network," in *IEEE 4th International Conference on Network and System Security (NSS)*, 2010, pp. 167–174.

[16] B. Westermann and D. Kesdogan, "Malice versus AN. ON: Possible risks of missing replay and integrity protection," in *Springer International Conference on Financial Cryptography and Data Security (FC)*, 2011, pp. 62–76.

[17] M. AlSabah, K. Bauer, and I. Goldberg, "Enhancing Tor's performance using real-time traffic classification," in *ACM Conference on Computer and Communications security (CCS)*, 2012, pp. 73–84.

[18] P. Liu, L. Wang, Q. Tan, Q. Li, X. Wang, and J. Shi, "Empirical measurement and analysis of I2P routers," *Journal of Networks*, vol. 9, no. 9, pp. 2269–2279, 2014.

[19] K. Shahbar and A. N. Zincir-Heywood, "Packet momentum for identification of anonymity networks," *Journal of Cyber Security and Mobility*, vol. 6, no. 1, pp. 27–56, 2017.

[20] A. Montieri, D. Ciuonzo, G. Aceto, and A. Pescapé, "Anonymity services Tor, I2P, JonDonym: Classifying in the dark," in *IEEE 29th International TeleTraffic Congress (ITC)*, 2017, pp. 1–6.

[21] Z. Rao, W. Niu, X. Zhang, and H. Li, "Tor anonymous traffic identification based on gravitational clustering," *Peer-to-Peer Networking and Applications*, pp. 1–10, 2017.

[22] K. Shahbar and A. N. Zincir-Heywood, "Weighted factors for evaluating anonymity," in *International Symposium on Foundations and Practice of Security (FPS), Springer LNCS*, 2017, pp. 235–240.

[23] Z. Ling, J. Luo, W. Yu, and X. Fu, "Equal-sized cells mean equal-sized packets in Tor?" in *IEEE International Conference on Communications (ICC)*, 2011, pp. 1–6.

[24] D. Herrmann, R. Wendolsky, and H. Federrath, "Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial Naïve-Bayes classifier," in *ACM workshop on Cloud computing security (CCSW)*, 2009, pp. 31–42.

[25] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website fingerprinting in onion routing based anonymization networks," in *ACM 10th annual Workshop on Privacy in the Electronic Society (WPES)*, 2011, pp. 103–114.

[26] A. Panchenko, F. Lanze, J. Pennekamp, T. Engel, A. Zinnen, M. Henze, and K. Wehrle, "Website fingerprinting at internet scale," in *Network and Distributed System Security Symposium (NDSS)*, 2016.

[27] A. Almubayed, H. Ali, and J. Atoum, "A model for detecting tor encrypted traffic using supervised machine learning," *International Journal of Computer Network and Information Security*, vol. 7, no. 7, p. 10, 2015.

[28] A. Springall, C. DeVito, and S.-H. S. Huang, "Per connection server-side identification of connections via tor," in *IEEE 29th International Conference on Advanced Information Networking and Applications (AINA)*, 2015, pp. 727–734.

[29] X. Bai, Y. Zhang, and X. Niu, "Traffic identification of Tor and Web-mix," in *Eighth International Conference on Intelligent Systems Design and Applications (ISDA)*, vol. 1. IEEE, 2008, pp. 548–551.

[30] G. He, M. Yang, J. Luo, and X. Gu, "Inferring application type information from tor encrypted traffic," in *IEEE 2nd International Conference on Advanced Cloud and Big Data (CBD)*, 2014, pp. 220–227.

[31] K. Shahbar and A. N. Zincir-Heywood, "Benchmarking two techniques for Tor classification: Flow level and circuit level classification," in *IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, 2014, pp. 1–8.

[32] ——, "Traffic flow analysis of Tor pluggable transports," in *IEEE 11th International Conference on Network and Service Management (CNSM)*, 2015, pp. 178–181.

[33] ——, "An analysis of Tor pluggable transports under adversarial conditions," in *IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 2017.

[34] ——, "Effects of shared bandwidth on anonymity of the I2P network users," in *IEEE Symposium on Security and Privacy, Workshop on Traffic Measurements for Cybersecurity (WTMC)*, 2017, pp. 235–240.

[35] A. H. Lashkari, G. Draper-Gil, M. Mamun, I. Saiful, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in *3rd International Conference on Information System Security and Privacy (ICISSP)*, 2017, pp. 253–262.

[36] S. Burschka and B. Dupasquier, "Tranalyzer: Versatile high performance network traffic analyser," in *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2016, pp. 1–8.

[37] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[38] "Tranalyzer2," [Online] http://tranalyzer.com, Sep. 2017.

[39] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, "Traffic classification on the fly," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 2, pp. 23–26, 2006.

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

[41] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[42] M. Hall, F. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[43] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine learning*, vol. 29, no. 2-3, pp. 131–163, 1997.

[44] J. R. Quinlan, *C4.5: programs for machine learning*. Elsevier, 2014.

[45] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[46] "NIMS: Network Information Management and Security Group," [Online] https://projects.cs.dal.ca/projectx/, Jul. 2017.

[47] D. Tammaro, S. Valenti, D. Rossi, and A. Pescapè, "Exploiting packet-sampling measurements for traffic characterization and classification," *International Journal of Network Management*, vol. 22, no. 6, pp. 451–476, 2012.

[48] E. Hjelmvik and W. John, "Breaking and improving protocol obfuscation," *Chalmers University of Technology, Tech. Rep*, vol. 123751, 2010.

[49] W. De Donato, A. Pescapé, and A. Dainotti, "Traffic identification engine: an open platform for traffic classification," *IEEE Network*, vol. 28, no. 2, pp. 56–64, 2014.

Antonio Montieri is a PhD Student at the Department of Electrical Engineering and Information Technology (DIETI) of the University of Napoli "Federico II" since February 2017. He has received his MS Degree from the same University in July 2015. Currently, he is part of the TRAFFIC research group whose activities are carried out in the field of Computer Networks. In detail, his work is focused on network measurements, (encrypted and mobile) traffic classification and modeling, monitoring and assessment of cloud network performance. Antonio has co-authored 13 papers and 4 posters accepted for publication in international journals and conference proceedings.

Domenico Ciuonzo (S'11-M'14-SM'16) is Researcher at Network Measurement and Monitoring (NM2), Naples, Italy. He holds a Ph.D. in Electronic Engineering from the Second University of Naples, Italy and, from 2011, he has held a number of visiting researcher appointments. Since 2014 he is member of the editorial board of several IEEE, IET and ELSEVIER journals. His research interests include data fusion, statistical signal processing, wireless sensor networks, traffic analysis and machine learning.

Giuseppe Aceto is a Post Doc at the Department of Electrical Engineering and Information Technology of University of Napoli "Federico II". Giuseppe has a PhD in telecommunication engineering from the University of Napoli "Federico II'. His work falls in measurement and monitoring of network performance and security, with focus on censorship. Recently, he is working on bioinformatic and ICTs applied to health. Giuseppe is the recipient of a best paper award at IEEE ISCC 2010.

Antonio Pescapè (SM'09) is a Full Professor of computer engineering at the University of Napoli "Federico II". His work focuses on Internet technologies and more precisely on measurement, monitoring, and analysis of the Internet. Recently, he is working on bioinformatic and ICTs for a smarter health. Antonio has co-authored more than 200 conference and journal papers and is the recipient of a number of research awards.