Anonymity Services Tor, I2P, JonDonym: Classifying in the Dark

Antonio Montieri[1], Domenico Ciuonzo[2], Giuseppe Aceto[1,2], Antonio Pescapé[1,2]
[1]University of Napoli "Federico II" (Italy), [2]NM2 s.r.l. (Italy)
{antonio.montieri, giuseppe.aceto, pescape}@unina.it, ciuonzo@nm-2.com

***Abstract*—Traffic classification, i.e. associating network traffic to the application that generated it, is an important tool for several tasks, spanning on different fields (security, management, traffic engineering, R&D). This process is challenged by applications that preserve Internet users' privacy by encrypting the communication content, and even more by anonymity tools, additionally hiding the source, the destination, and the nature of the communication. In this paper, leveraging a public dataset released in 2017, we provide (repeatable) classification results with the aim of investigating to what degree the specific anonymity tool (and the traffic it hides) can be identified, when compared to the traffic of the other considered anonymity tools, using machine learning approaches based on the sole statistical features. To this end, four classifiers are trained and tested on the dataset: (i) Naïve Bayes, (ii) Bayesian Network, (iii) C4.5, and (iv) Random Forest. Results show that the three considered anonymity networks (Tor, I2P, JonDonym) can be easily distinguished (with an accuracy of 99.99%), telling even the specific application generating the traffic (with an accuracy of 98.00%).**

***Index Terms*—dark web; Tor; I2P; JonDonym; traffic classification; anonymity; privacy; security.**

I. Introduction

The increase of people's online activities over last years has lead to a growing concern on people's privacy and anonymity. As a consequence, Anonymity Tools (ATs) are commonly employed by Internet users to achieve privacy at some extent, i.e. to hide the source, the destination, and the nature of the communication, other than encrypting the content itself. In addition, they are even capable of hiding the users' identity even to the final destination (i.e. the web-server). These services provide anonymity to the users by forwarding the users' traffic through multiple stations until the users' data reach their destination. During this journey, the data are encrypted multiple times. By doing so, users' data keep their anonymity since each station composing the path knows only part of the information. Hence, it is difficult tracing users' data within these networks. From users' perspective, such tools allow to browse the web "freely" or run applications without revealing their identity to any site observing the network (i.e. eavesdroppers or 3rd-party sniffing). Among the several ATs developed in last years, the Onion Router (Tor) [1], the Invisible Internet Project (I2P) [2] and JonDonym [3] are the most popular.

In recent years, ATs have been investigated from disparate perspectives by several studies, collectively covering a wide spectrum of topics. These included very narrow aspects of anonymity "realm", such as improving the design of a specific AT, studying its performance and delay, how to perform attacks on it, analyzing the behavior of its users, revealing its users' identity, censoring them [4] so to name a few.

Among the many important aspects of ATs, one fundamental issue is to *understand whether their traffic data can be classified and, if so, to which depth*. More specifically, it is interesting to ascertain to which degree an AT can be recognized from an external observer and how finer would be the fingerprinting granularity achievable, that is, whether traffic types and/or services hidden into them could be inferred. This investigation would also shed new lights on how privacy of anonymity networks could be further *robustified*. Therefore, Traffic Classification (TC) of ATs traffic is a recent open research field. TC is an important part of Internet traffic engineering and has applications in several fields such as network monitoring, application identification, anomaly detection, accounting, advertising and service differentiation [5]. TC has gained on importance in recent years due to increased interest in service differentiation and growing incentives to disguise certain applications [6], also the ones generating anonymous traffic. TC mechanisms consist in associating (labeling) traffic flows with specific application types, moving from earlier port-based methods, to those based on payload inspection (termed Deep Packet Inspection methods, DPI [7, 8]) and to those based on Machine Learning (ML) classifiers (either supervised and unsupervised), making decisions based on the sole observation of traffic-flow [9] or packet-based features [10]. Thus, ML-based techniques *perfectly suit* to anonymous (encrypted) traffic analysis.

Exacerbating the lack of data for experimenting with traffic classification 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*, thus precluding unanimous and shared conclusions, as well as they preclude experiments repeatability. Indeed, previous works on ATs have been based on either (i) data collected within a simulated environment [11, 12] or (ii) data generated from real traffic on anonymous networks by researchers[1] themselves [13, 14, 15, 16]. Unfortunately, in the latter case, researchers have been reluctant to making the collected data publicly

[1]As explained 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.

available for reasons of users' privacy.

A fundamental opportunity in this direction, allowing to answer the main question behind the present study (i.e. whether identifiability of anonymous networks is possible), is represented by the recently released Anon17 dataset [17]. Indeed, this public dataset consists of a collection of traces gathered by Tor, I2P and JonDonym, as well as related services and applications running inside these networks (such as Tor pluggable transports or EEpsites on I2P). To the best of authors' knowledge, no similar datasets have been made available online up to date. Therefore, Anon17 represents an important (shared) workbench for research studies on the topic.

In view of the aforementioned reasons, the main contribution of this paper is an investigation on whether anonymity networks (such as Tor, JonDonym, and I2P) can be discerned. Our analysis is carried out at different levels of granularity, that is, we try to analyze whether the *Anonymity Network* being observed (briefly referred to as L1 in what follows) can be classified and, in affirmative case, whether the *Traffic Type* (L2) and *Application* (L3) running *hidden* within them could be inferred. To the best of our knowledge, there is no similar classification-based analysis of anonymity tools in the literature, both in terms of a similar viewpoint and detail of the analysis. We consider four classifiers: two based on the *Bayesian approach* (i.e. Naïve Bayes and Bayesian Networks) and other two based on *decision trees* (i.e. C4.5 and Random Forest). The obtained results show that anonymity networks *can* be easily discerned, and the traffic type and the service running within it can be accurately inferred as well (by a judicious use of the appropriate classifier).

The remainder of the paper is organized as follows. Sec. II discusses related work on the topic, whereas Sec. III describes the considered TC framework of anonymity services; experimental results are reported in Sec. IV; finally, Sec. V provides conclusions and future directions.

II. Related Work

To the best of our knowledge, due to the lack of available datasets, there are no studies focused on the classification and identification of different anonymity services at various levels of granularity, that is discerning in the encrypted traffic they generate: the specific Anonymous Network (L1), Traffic Type (L2), and Application (L3). Therefore, in this section, we report a larger discussion of the state-of-the-art of anonymous (encrypted) traffic investigation, which has been the subject of many works in last years. Then, in the latter part, we focus on works dealing with anonymous traffic classification (underlining also their difference to the present study).

First attempts of analyzing anonymity networks are provided in *emulation tools* [11, 12], focusing on Tor network. Specifically, in [11] a Tor network emulator (named *ExperimenTor*) is presented, providing a test environment which allows modeling of relevant "actors", such as Tor routers, bandwidth, users, and applications. Similarly, a simulated environment with the intent of differentiating between (encrypted) HTTPS and Tor traffic is developed in [12]. To perform the comparison, the following traffic types are gathered: (i) true HTTPS traffic; (ii) HTTP over the simulated Tor network; (iii) HTTPS over the simulated network. Random Forest, J4.8 (C4.5), and AdaBoost classifiers are employed, showing that $\geq 90\%$ of the HTTP and HTTPS traffic over Tor can be detected (subject to 3.7% of false-positive rate).

Some other works have focused on Tor, JonDonym, and I2P based on *real data* [13, 14, 18, 16, 19], concentrating however on some other aspects, such as designing the attack type, evaluating the volume of traffic run within, discovering "anonymous" routers or providing guidelines for evaluating privacy level of a generic AT. For example, Ling et al. [18] propose an active attack and a detection mechanism to degrade users' privacy within Tor, based on transmitted packet size from the web server through the router to the user. Since Tor has a fixed cell size (512 bytes) and varying packet sizes, one-bit padding is used to mark the packets in order to trace them back at the receiver side. Indeed, the padding length forces the Tor router to use a known number of cells. Then, if the data exceed the cell size, fragmentation is employed, allowing the attacker to mark the client who receives these cells as the client has access to the server. Since packets may face congestion, retransmission or any normal traffic behavior during the server-client path, a delay is added in between buffered packets before transmission to ensure correct detection of the encoded bit at the receiver. It is shown that 10 packets are sufficient for this method to achieve a 90% detection rate (subject to a false-positive rate of 4%). Differently, in [19] five factors for measuring the anonymity level of any generic AT from the user's perspective are proposed, taking Tor, I2P, and JonDonym as case studies. Also, a weighted combination of these factors is proposed, with weights obtained via a pairwise comparison technique. The analysis shows that though these ATs claim to provide total users anonymity, some information (about the users) contained in these ATs is available to the operators of the services.

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

In [20], a Support Vector Classifier is employed for website fingerprinting over Tor and JonDonym (*separately*), underlining their not complete 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 a known set of 775 websites (each with 20 instances) on either Tor or JonDonym. The results show classification improvement (over a known set of websites) from 3% to 55% (resp. from 20% to 80%) in Tor (resp. JonDonym) network over previous works. 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 provided by Alexa and other 1000 URLs (disjunct from the training set) are added to the test data. In this case, the detection rate is 73% (with 0.05% false-positive rate). Another ML-based approach is proposed by AlSabah et al. [15] to recognize applications (browsing, streaming, and BitTorrent)

used by Tor's users by means 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 (both cell- and circuit-level info is used to classify the circuit) classification is considered, with the best accuracy obtained for online (resp. offline) case equal to 97.8% (resp. 91%). A similar setup is studied in [21], where four classifiers (Naïve Bayes, Bayesian Network, Random Forest and C4.5) are used to recognize user activities based on *traffic-flow* features and compared it with classification based on *circuit-level* features. The results show a high accuracy (up to 100%) with both approaches, flow-based classification being however *less demanding*.[2]

Similarly, Shahbar and Zincir-Heywood [22] investigate whether Tor Pluggable Transports (PTs) can evade a flow-based traffic analysis by blocking systems. Indeed, Tor PTs have been 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. Unfortunately, PTs are designed to hide only the content of Tor connections; thus, a flow-based analysis can potentially identify Tor traffic even in the presence of such obfuscation techniques. By adopting a C4.5 classifier, results show 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 analysis. The above work is extended in [23], where the aim is the recognition of Tor PTs in terms of describing the proper features, the sufficient amount of data, and the effect of data collection on flow-based classification of Tor PTs. The same authors in [24] analyze the effects of bandwidth sharing on I2P, investigating both application and user profiling achievable by an attacker. The analysis relies on a C4.5 classifier built on flow-based features. Results show that users and applications on I2P *can* be profiled, with a detrimental (resp. beneficial) effect of the shared bandwidth increase on applications (resp. users) profiling accuracy. Additionally, not using the shared client tunnels for all applications seems to increase applications profiling.

Finally, *Anon17* dataset is presented in [17]. As anticipated in Sec. I, the latter is a directional traffic-flow dataset collecting data from three different ATs (i.e. Tor, I2P, and JonDonym). Additionally, it provides information at finer granularity (i.e. traffic type and application levels), by providing labels for traffic flows pertaining to applications running on Tor and I2P (in different flavors), as well as the PTs employed on the Tor network. To the best of our knowledge, no similar datasets are available publicly at the date.

[2]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.

III. Traffic Classification

In the following, terms and concepts regarding traffic objects are introduced, together with an overview of the classification features available in the Anon17 dataset. The last part describes the classification algorithms adopted for anonymous traffic analysis.

A. Traffic View

According to [17], the anonymous traffic contained in Anon17 is split into different *flows* [5], obtained as result of the application of the flow-exporting tool *Tranalyzer2* [25]. The direction of each flow is then marked as a feature (see details in Sec. III-B), 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 [25].

B. Classification Features

The traffic features available in Anon17 dataset are obtained starting from Tranalyzer2 [25]. More specifically, this 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 provides a total of 92 features per flow. However, the dataset provides only a subset of these features, since some of them are removed such as ICMP and VLAN features, since they do not provide useful fingerprinting information. Aiming at protecting users' privacy, IP addresses and payloads of the packets are also removed from the dataset. Therefore, Anon17 is provided in the form of a subset of 81 features per flow extracted by the aforementioned tool, comprising:

- Flow direction (A/B), starting/ending timestamps, and duration of the flow;
- No. of bytes/packets Tx/Rx (including bytes/packet 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.);
- Joint PL-IAT statistics (such as histograms);
- 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 [17], since I2P network works on both TCP and UDP, for UDP connections over I2P the TCP-related features may have zero value. Furthermore, for our classification problem we have made the following choices:

- We have not considered the joint PL-IAT statistics (namely the features `nfp_pl_iat` and `ps_iat_histo`), as it is apparent that these features require further processing (and investigation) on how

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

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

they could be effectively (and efficiently) exploited in any of the classification approaches being considered. In any case, their exploitation appears extremely interesting, given the usefulness demonstrated in traffic modeling and classification [10].

- We have removed the features `minPktSz`, `maxPktSz`, and `avePktSize`, as they seem repeated with respect to `min_pl`, `max_pl` , and `mean_pl`, respectively, considering the specific configuration adopted in Tranalyzer2 for capturing the traffic.

Therefore, in view of the aforementioned considerations, the classifiers being compared will be all fed with a reduced set of 76 features. This set of M features adopted by each classifier will be generically indicated with $f_1, \ldots, f_M$ (or collectively as $\boldsymbol{f} \triangleq \begin{bmatrix} f_1 & \cdots & f_M \end{bmatrix}^T$) and the set of classes as $\Omega \triangleq \{c_1, \ldots c_L\}$. Finally, features' relative importance (based on statistical rankings) will be later analyzed in Sec. IV-B.

C. Classification Algorithms

In this sub-section we review four supervised classification algorithms successfully employed in several works tackling TC of anonymous traffic [15, 21, 22, 24], that are applied to the TC scenario investigated in this work: (i) Naïve Bayes, (ii) Bayesian Networks, (iii) C4.5, and (iv) Random Forest.

Naïve Bayes (NB): The NB is a simple probabilistic classifier that assumes class-conditional independence of the features $\boldsymbol{f}$, i.e. $P(f_1, \ldots, f_M|c_j) = \prod_{m=1}^{M} P(f_m|c_j)$, being not the case for real-world problems, but working well in practice and leading to reduced complexity. It 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 $P(c_i|\boldsymbol{f}_T) \propto P(c_i) \prod_{m=1}^{M} P(f_{T,m}|c_i)$. 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 PDF $P(f_m|c_i)$ is estimated by resorting to multinomial PMFs when the features are *categorical*, whereas different choices may be pursued for *numerical* features. In this paper, we focus on (simple) moment matching to a Gaussian PDF [26].

Bayesian Networks (BNs): BNs are graphical representations which model dependence relationships between features and classes [27], 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 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.

C4.5: C4.5 is an algorithm employed to generate a decision tree used (mainly) for classification purposes [28], based on the concept of *entropy* of a distribution [29].

The training algorithm is based on a (greedy) top-down recursive tree 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[3] measure in the data, such as the "gain ratio", that avoids bias toward features with a larger support [28]. 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.

Random Forest (RF): RF is a classification method based on an ensemble of several decision trees, built at training time exploiting the ideas of "bootstrap aggregating" (bagging) and random-feature selection to control variance and thus avoid over-training [30].

Specifically, during the training phase, each decision tree in the RF classifier is grown based on a bootstrap (i.e. a uniformly

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

random sampling procedure with replacement) sample set of the training data available. The number of trees B denotes a free parameter which can be tuned by using cross-validation or by observing the out-of-bag error. Finally, after training, decision on testing samples can be made by taking the majority vote or soft combination of the responses of B trees.

IV. Experimental Results

This section reports details about the Anon17 dataset and the pre-processing operations carried out on it, and shows the results of the classification experiments performed.

A. Dataset Description

Anon17 was collected at the Network Information Management and Security Lab [31] 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 is formatted into ARFF format used in the data mining software tool *Weka* [26] and reports features (discussed in Sec. III-B) on per-flow basis (unfortunately this prevents additional analyses, as the evaluation of packet sampling impact on classification accuracy [32]). We refer to [17] for further details on the 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. [12] 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. Second, the results of the present analysis can be intended as an upper bound on 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. I, 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. I. The total number of applications (L3 classes) identified for each anon network (3 L1 classes) and traffic type (7 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

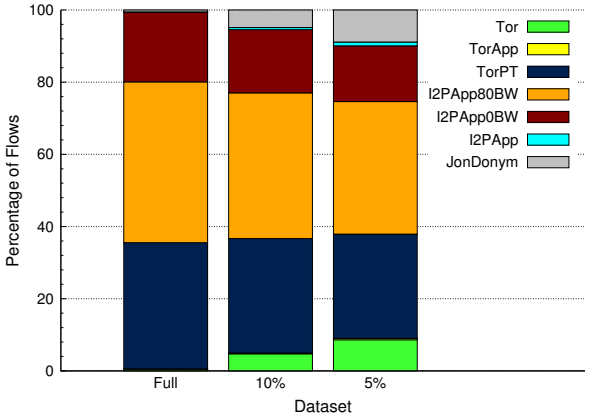


Figure 1: Intelligent down-sampling of Anon17 dataset: left chart (original dataset), middle chart (down-sampling to 10%, D_{10}), right chart (down-sampling to 5%, D_5).

to flows running 3 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 5 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 3 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 3 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.

Anon17 exhibits a (majority) class imbalance problem[4], as shown by the total number of samples in Tab. I. To cope with it, we randomly down-sample (without replacement) by applying a filter[5] 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 context, we will consider two configurations, corresponding to down-sampling to 5% and 10% of the original dataset of each traffic type set (referred to as D_5 and D_{10} in what follows). Fig. 1 shows the percentage of flows labeled with different traffic types after performing the above-mentioned down-sampling.

B. Classification Results

In this section, we show results pertaining to several sets of experiments, obtained through: (i) different classification

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

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

Table II: Overall accuracy and macro F-measure for dataset D_5 (dataset D_{10}) with the whole set of 76 features employed. Highlighted values: **maximum per Split** and maximum per 10-fold for each level.

Classifier		Accuracy L1	Accuracy L2	Accuracy L3	F-measure L1	F-measure L2	F-measure L3
NB	*Split*	99.72% (99.80%)	85.44% (83.34%)	66.76% (66.41%)	99.70% (99.80%)	80.90% (77.70%)	62.70% (62.00%)
	10-fold	99.75% (99.79%)	85.15% (85.95%)	65.23% (63.11%)	99.70% (99.80%)	80.50% (82.70%)	61.00% (58.60%)
BN	*Split*	98.20% (98.43%)	90.43% (89.94%)	82.41% (81.35%)	98.20% (98.40%)	91.40% (91.30%)	83.80% (83.20%)
	10-fold	98.30% (98.53%)	90.35% (89.83%)	82.15% (81.54%)	98.30% (98.50%)	91.40% (91.20%)	83.60% (83.30%)
C4.5	*Split*	**100.00% (100.00%)**	**99.95%** (99.96%)	97.72% **(97.97%)**	**100.00% (100.00%)**	**99.99% (99.99%)**	97.70% **(98.00%)**
	10-fold	99.99% (99.99%)	99.97% (99.97%)	98.03% (98.25%)	99.99% (99.99%)	99.99% (99.99%)	98.00% (98.30%)
RF	*Split*	99.98% (99.99 %)	99.65% (99.72%)	95.81% (96.53%)	99.99% (99.99%)	99.70% (99.70%)	95.80% (96.50%)
	10-fold	99.99% (99.99%)	99.72% (99.81%)	96.18% (96.69%)	99.99% (99.99%)	99.70% (99.80%)	96.20% (96.70%)

Table III: Least 5 discernible applications (L3) for each classifier, ranked by F-measure, with dataset D_5 and 10-fold validation.

Application L3 (NB)	Precision	Recall	F-Measure
I2PSnark (j)	64.50%	2.90%	5.50%
I2PSnark (m)	76.00%	4.30%	8.10%
jIRCii (n)	15.30%	27.50%	19.70%
Eepsites (l)	29.10%	19.00%	23.00%
I2PSnark (p)	40.50%	79.00%	53.60%

Application L3 (BN)	Precision	Recall	F-Measure
Eepsites (r)	8.70%	69.70%	15.50%
I2PSnark (p)	12.30%	75.80%	21.20%
Participating Tunnels (t)	24.50%	54.00%	33.70%
jIRCii (q)	25.40%	80.10%	38.60%
Exploratory Tunnels (s)	25.80%	79.10%	38.90%

Application L3 (C4.5)	Precision	Recall	F-Measure
jIRCii (n)	92.10%	92.30%	92.20%
I2PSnark (m)	94.80%	94.50%	94.70%
Browsing (d)	97.60%	95.20%	96.40%
Eepsites (o)	97.40%	97.40%	97.40%
Streaming (b)	97.60%	97.60%	97.60%

Application L3 (RF)	Precision	Recall	F-Measure
Participating Tunnels (t)	65.70%	34.90%	45.60%
Exploratory Tunnels (s)	60.50%	57.00%	58.70%
Eepsites (r)	72.80%	51.70%	60.50%
I2PSnark (p)	84.30%	69.40%	76.10%
jIRCii (q)	84.90%	78.70%	81.70%

Table IV: Overall accuracy and macro F-measure vs. feature set size $(19, 38, 76)$ for L3, with dataset D_5. Highlighted values: **maximum per Split** and maximum per 10-fold for each feature set size.

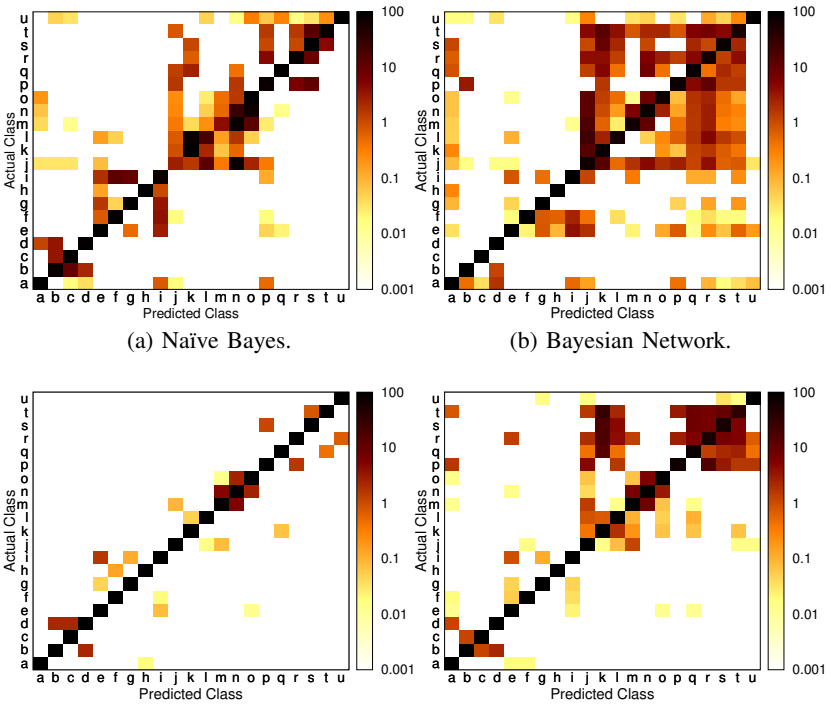
Classifier		Acc 100%	Acc 50%	Acc 25%	F-meas 100%	F-meas 50%	F-meas 25%
NB	*Split*	66.76%	66.61%	67.45%	62.70%	62.70%	63.80%
	10-fold	65.23%	65.62%	66.30%	61.00%	61.90%	62.80%
BN	*Split*	82.41%	88.33%	91.70%	83.80%	89.10%	92.10%
	10-fold	82.15%	88.58%	91.64%	83.60%	89.40%	92.10%
C4.5	*Split*	**97.72%**	**97.94%**	96.11%	**97.70%**	**97.90%**	96.10%
	10-fold	98.03%	97.97%	95.99%	98.00%	98.00%	96.00%
RF	*Split*	95.81%	96.90%	**96.16%**	95.80%	96.90%	**96.20%**
	10-fold	96.18%	97.08%	96.25%	96.20%	97.10%	96.30%

algorithms[6], (ii) different down-samplings of the dataset to cope with class imbalance problem, and (iii) varying the size of the (ranked) subset of features. Our comparison will be based on the following performance measures [33]: *overall accuracy*, *precision*, and *recall*. Since these last two metrics are defined on a per-class basis, their weighted averaged versions will be employed when a synthetic measure will be needed. Additionally, we will consider the *F-measure* ($\mathrm{F} \triangleq (2 \cdot \mathrm{prec} \cdot \mathrm{rec})/(\mathrm{prec} + \mathrm{rec})$), so that to account for both the effects of precision (prec) and recall (rec) in a concise fashion. Moreover, we will also consider confusion matrices of classifiers to provide their complete performance "pictures" and identify the most frequent misclassification patterns. In detail, for each considered analysis, we will evaluate (for completeness) two different *unbiased* evaluation setups:

- a random training-test set splitting (with corresponding percentages 70%-30%);
- a (stratified) 10-fold cross-validation analysis.

Firstly, in Tab. II we report the overall accuracy and the F-measure achieved by the considered classifiers for both the datasets D_{10} and D_5, and the two testing setups. Also, these performance measures are specialized for classification at different levels of granularity (i.e. L1, L2, and L3) being considered. From the inspection of Tab. II, it is apparent that all the classifiers achieve extremely satisfactory performance in discrimination of the three ATs contained in Anon17, with C4.5 achieving the best performance. The same consideration applies to BN, C4.5 and RF on L2 discrimination. On the

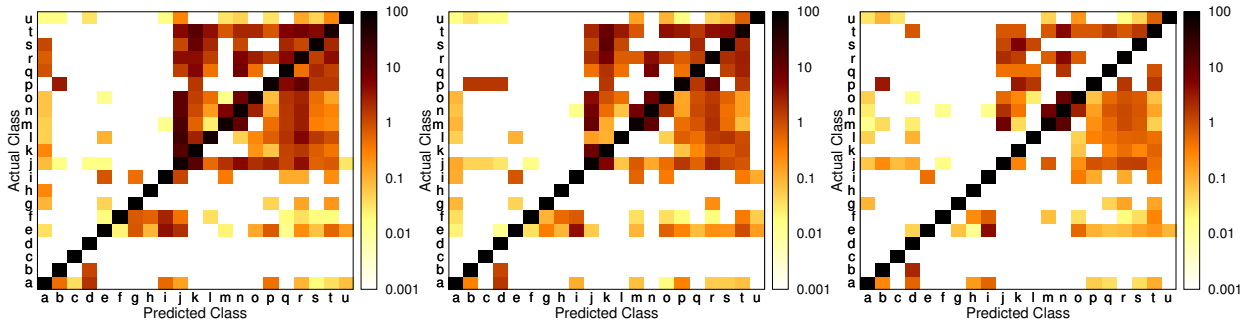
[6]The considered classifiers have been implemented within the well-known Weka framework with default options [26].



(a) Naïve Bayes. (b) Bayesian Network.

(c) C4.5. (d) Random Forest.

Figure 2: Confusion matrices (L3, percentage accuracy, log scale) for dataset D_5 and 10-fold validation (see Tab. I for class labels).



(a) Full set of features. (b) Top 38 features. (c) Top 19 features.

Figure 3: Confusion matrices (L3, percentage accuracy, log scale) for Bayesian Network, dataset D_5 and 10-fold validation.

other hand, performance metrics generally degrade with the increasing granularity of the classification task (i.e. moving from L1 to L3), which can be observed for both the under-sampled datasets (i.e. D_{10} and D_5) and the type of setup employed (i.e. random splitting or 10-fold validation). 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 simply 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 more significant for `BN` and penalizing for `NB`. This finding can be explained as the (strong) conditional independence assumption of the features is *limiting* when tackling harder classification tasks (i.e. L3).

Since classification at L3 is the most challenging task (but also the most interesting from a user's privacy perspective) and, given the satisfactory performance of (almost) all the classifiers at first two levels, in what follows we concentrate on L3. To this end, we show *per-class* precision, recall, and F-measure to highlight classifiers' per-application behavior. For brevity, we only focus on D_5 in what follows, since similar observations have been observed for D_{10}. More specifically, for each classifier we report in Tab. III the 5 applications with the lowest recognition score, ranked according to the F-measure (10-fold validation). Interestingly, `RF`, and `BN` are equally prone to errors related to `I2P apps`. Differently, `NB` has severe problems with classification of `I2PSnark` (i.e. `j`, `m`, and, `p`). Finally `C4.5`, which outperforms all

other classifiers, has its performance limited superiorly by I2P tunnels with 80% bandwidth sharing (i.e. `t` and `s`) and by `streaming` and `browsing` applications running over Tor (i.e. `b` and `d`, respectively).

To investigate in detail the error patterns of each classifier, in Fig. 2 we show the corresponding confusion matrices (for 10-fold validation setup). We recall that for these matrices the higher the concentration toward the main diagonal, the better the overall performance. Firstly, it is apparent that all the classifiers (with different quantitative outcomes) present error patterns which almost entirely lead to a misclassification of the traffic type and/or the application *within the same* anonymous network. For example, rarer are misclassifications of `Tor apps` and `Tor PTs` with applications running within I2P. This is apparent by looking at the error-pattern cluster for `NB`, `BN`, and `RF`, especially in relationship to I2P, whereas for `C4.5` error patterns are less frequent, but still confined to the same network or traffic type. In the latter case, error patterns in `C4.5` are mostly due to misclassification of applications belonging to `I2P apps tunnels with other tunnels` (in the presence of 80% of shared bandwidth) and to misclassification within `Tor apps`.

Finally, we investigate the relative importance of the set of features being considered by evaluating feature selection effect on classifiers performance at the application level L3. More specifically, we consider three scenarios: performance with (i) the whole set of feature, (ii) the 50% of the feature set (i.e. 38 features) and (iii) the 25% of the feature set (i.e. 19 features). The features have been ranked in decreasing *Information Gain*[7] (i.e. the mutual information between the class variable and the generic feature). To this end, in Tab. IV we report the overall accuracy and F-measure for the considered scenarios, for both the reduced datasets and both the employed setups. By looking at the results reported, a substantial insensitivity of `NB`, `C4.5` and `RF` to feature selection can be noted. This is explained as almost all the discrimination power for the above classifiers resides within the first 19 features. A different conclusion is drawn for `BN` instead, which enjoys a boost in classification accuracy (resp. F-measure) of 9.3% (resp. 8.3%) when reducing the number of features to one quarter of the original set. This is attributed to higher requirements for structure and parameter learning in the case of a larger set of features (needing a higher number of training samples), whereas in the case of a reduced set of features, a simplified structure has to be learned. The enhancement is not only due to the improvement in the discrimination accuracy for the most frequent class, as confirmed by the confusion matrices of `BN` (10-fold validation) for the three feature sets considered in Fig. 3. Indeed it is apparent that feature selection is able to provide a homogeneous reduction of error patterns, especially those related to the I2P "cluster".

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

C. Comparison with Literature

In this section, we compare closest work to ours. In [21], high accuracy is achieved in classifying Tor applications (i.e. browsing, streaming and BitTorrent) via flow-based classification based on Tranalyzer2. The results shown in Fig. 2 are compatible with the above work (interestingly, also in our harder classification task, `C4.5` performed the best) and further show that Tor app can be hardly misclassified with apps from other anonymous networks (such as I2P). Differently, in [22], 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. Finally, in [19] the effect of bandwidth participation on I2P is investigated, showing higher application profiling with less bandwidth sharing. This trend qualitatively agrees with Tab. III, where the best performing classifier (i.e. C4.5) 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. Nonetheless, the present work provides a more comprehensive study of traffic classification and identification of different ATs at different granularities, underlining the narrowness of the above studies (i.e. focusing on a particular AT or a specific aspect of it).

V. Conclusions

This paper tackled Traffic Classification of Anonymity Tools, 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 intelligent downsampling to cope with its strong class imbalance. Different classification algorithms (Naïve Bayes, Bayesian Network, C4.5, Random Forest) have been applied to the processed dataset, also varying the considered feature sets. Results show that all considered classifiers obtain extremely satisfactory performance in discriminating the anonymity networks present in Anon17, with C4.5 achieving virtually ideal results. Finally, our analysis shows that Tor, I2P and JonDonym anonymous networks can be hardly mistaken for each other, and that further digging down in the specific type of traffic tunneled, and the specific type of application generating such traffic, is possible with up to 98.03% accuracy and 98.00% F-measure with C4.5 (on 5%-downsampled dataset D_5). We found substantial insensitivity to feature selection for Naïve Bayes, C4.5, and Random Forest, while Bayesian Networks improve classification accuracy of $\approx 9\%$ with intelligent selection. 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) exploitation of histogram-based features [6], (iii) dif-

ferent feature selection methods and correlation analysis, (iv) comparison with other public labeled datasets (possibly also in an open-world assumption), should they become available, and (v) implementation of features and classifiers in the open-source TC platform TIE [34] 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 SSYM'04*.

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

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

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

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

[6] A. Dainotti, A. Pescapé, and C. Sansone, "Early classification of network traffic through multi-classification," in *TMA'11*. Springer, pp. 122–135.

[7] 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 GLOBECOM'09*.

[8] G. Aceto, A. Dainotti, W. De Donato, and A. Pescapé, "PortLoad: taking the best of two worlds in traffic classification," in *IEEE INFOCOM'10*, pp. 1–5.

[9] 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.

[10] A. Dainotti, A. Pescapé, and H.-c. Kim, "Traffic classification through joint distributions of packet-level statistics," in *IEEE GLOBECOM'11*, pp. 1–6.

[11] K. Bauer, M. Sherr, D. McCoy, and D. Grunwald, "ExperimenTor: a testbed for safe and realistic Tor experimentation," in *USENIX CSET'11*.

[12] J. Barker, P. Hannay, and P. Szewczyk, "Using traffic analysis to identify the second generation onion router," in *IEEE/IFIP EUC'11*, pp. 72–78.

[13] A. Chaabane, P. Manils, and M. A. Kaafar, "Digging into anonymous traffic: A deep analysis of the Tor anonymizing network," in *IEEE NSS'10*, pp. 167–174.

[14] B. Westermann and D. Kesdogan, "Malice versus AN.ON: Possible risks of missing replay and integrity protection," in *Springer FC'11*, pp. 62–76.

[15] M. AlSabah, K. Bauer, and I. Goldberg, "Enhancing Tor's performance using real-time traffic classification," in *ACM CCS'12*, pp. 73–84.

[16] 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.

[17] K. Shahbar and A. N. Zincir-Heywood, "Anon17: Network traffic dataset of anonymity services," Faculty of Computer Science Dalhousie University, Tech. Rep., Mar. 2017.

[18] Z. Ling, J. Luo, W. Yu, and X. Fu, "Equal-sized cells mean equal-sized packets in Tor?" in *IEEE ICC'11*.

[19] K. Shahbar and A. N. Zincir-Heywood, "Weighted factors for measuring anonymity services: A case study on Tor, JonDonym, and I2P," Faculty of Computer Science Dalhousie University, Tech. Rep., Mar. 2017.

[20] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website fingerprinting in onion routing based anonymization networks," in *ACM WPES'11*.

[21] K. Shahbar and A. N. Zincir-Heywood, "Benchmarking two techniques for Tor classification: Flow level and circuit level classification," in *IEEE CICS'14*, pp. 1–8.

[22] ——, "Traffic flow analysis of Tor pluggable transports," in *IEEE CNSM'15*, pp. 178–181.

[23] ——, "An analysis of Tor pluggable transports under adversarial conditions," Faculty of Computer Science Dalhousie University, Tech. Rep., Mar. 2017.

[24] ——, "Effects of shared bandwidth on anonymity of the I2P network users," Faculty of Computer Science Dalhousie University, Tech. Rep., Mar. 2017.

[25] S. Burschka and B. Dupasquier, "Tranalyzer: Versatile high performance network traffic analyser," in *IEEE SSCI'16*, pp. 1–8.

[26] 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.

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

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

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

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

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

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

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

[34] 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.