Unveiling MIMETIC: Interpreting Deep Learning Traffic Classifiers via XAI Techniques

Alfredo Nascita, Antonio Montieri, Giuseppe Aceto, Domenico Ciuonzo, Valerio Persico, Antonio Pescapè
University of Napoli "Federico II" (Italy)
a.nascita@studenti.unina.it, {antonio.montieri, giuseppe.aceto, domenico.ciuonzo, valerio.persico, pescape}@unina.it

***Abstract*—The widespread use of powerful mobile devices has deeply affected the mix of traffic traversing both the Internet and enterprise networks (with bring-your-own-device policies). Traffic encryption has become extremely common, and the quick proliferation of mobile apps and their simple distribution and update have created a specifically challenging scenario for traffic classification and its uses, especially network-security related ones. The recent rise of Deep Learning (DL) has responded to this challenge, by providing a solution to the time-consuming and human-limited handcrafted feature design, and better classification performance. The counterpart of the advantages is the lack of interpretability of these black-box approaches, limiting or preventing their adoption in contexts where the reliability of results, or interpretability of polices is necessary. To cope with these limitations, eXplainable Artificial Intelligence (XAI) techniques have seen recent intensive research. Along these lines, our work applies XAI-based techniques (namely, Deep SHAP) to interpret the behavior of a state-of-the-art multimodal DL traffic classifier. As opposed to common results seen in XAI, we aim at a global interpretation, rather than sample-based ones. The results quantify the importance of each modality (payload- or header-based), and of specific subsets of inputs (e.g., TLS SNI and TCP Window Size) in determining the classification outcome, down to per-class (viz. application) level. The analysis is based on a publicly-released recent dataset focused on mobile app traffic.**

***Index Terms*—traffic classification; encrypted traffic; explainable artificial intelligence; deep learning; multimodal learning.**

I. Introduction

The knowledge of the mix of traffic traversing a network is instrumental to several management activities: Traffic Classification (TC) has a key role in defining a "normal" traffic profile for the purpose of anomaly detection, or to extracting (or inferring) fingerprints for intrusion detection and attack identification. Moreover, TC can be also exploited for defining technical boundaries for censorship enforceability, and assessing the effectiveness of surveillance and blocking countermeasures. For these reasons TC has seen consistent research and field adoption along the years, and is now seeing a renewed blossoming of interest due to recent evolution of network usage. Indeed, the widespread availability of well-equipped smartphones has impacted both the Internet and enterprise networks (due to bring-your-own-device policies), presenting a highly dynamic and extensively encrypted mix of traffic. On the other hand, new powerful Artificial Intelligence techniques (namely Deep Learning, "DL" in the following) have become available to face the new classification challenges. DL approaches are characterized by a fully-automated feature extraction phase (with reduced need of human experts in the loop) and a greater ability of learning from huge volumes of data, that provides better performance than the traditional Machine Learning (ML) approaches.

978-1-7281-5684-2/20/$31.00 ©2021 IEEE

The highly desirable characteristics of DL come at the cost of lack of *interpretability* of their results, as the black-box nature of DL techniques hides the reason behind specific classification outcomes. This impacts the understanding of classification errors and the evaluation of the resilience against adversarial manipulation of traffic to impair identification. Moreover, by understanding the behavior of the learned model, performance enhancements can be pursued with much more focused and efficient research, compared with a less-informed exploration of the (typically huge) hyper-parameters space. In fact, DL approaches keep naturally hidden the answers to basic questions like *"which parts of a complex architecture mostly contribute to the final decision?"*, *"which specific fields, packets, protocols are the most important in the classification process?"*, or *"which ones are responsible for classification errors or circumvention?"*.

The field of *eXplainable Artificial Intelligence* (XAI) constitutes the answer to these needs, as it provides approaches and techniques able to relate the structure of the model and the input to the respective classification outcome, partially revealing the (former) completely black box. The adoption of DL and (consequently) of XAI is relatively new, especially in the field of network traffic classification: with this work we contribute to this step forward in the understanding of DL-based network traffic classifiers.

To this aim, we perform the behavior interpretation of a state-of-the-art DL architecture for TC we recently proposed [1], analyzing the relative importance of inputs at fine grain (i.e. per-class) in the challenging task of classifying mobile apps. More specifically, we apply state-of-the-art XAI tools (namely, Deep SHAP [2]) to quantify and understand the importance of payload-derived and header-based inputs, further deepening the analysis to specific subsets of the inputs (i.e., TLS-SNI in the payload, and TCP Window Size and Payload Length, and packet Inter-Arrival Time and Direction for the header-based). To perform our experimental evaluation, we leverage the public traffic dataset MIRAGE-2019 that focuses on mobile-app traffic and is human-generated [3].

The paper is organized as follows. Section II surveys first

attempts of XAI application to networking and TC, positioning our work against related literature. Section III describes the considered XAI-based TC methodology, while the dataset employed and the experimental results are discussed in Sec. IV. Finally, Sec. V provides conclusions and future perspectives.

II. Background and Related Work

The huge success in several tasks such as image classification and speech recognition has paved the way to the use of AI for activities related to decision making, cost reduction, risk management, etc. However, models resulting from complex AI algorithms have a black-box nature which leads to an undesired *lack of interpretability*. As a result, in the last years many efforts have focused on assessing the transparency, causality, bias, fairness, and safety of the obtained solutions [4]. Investigating the working behavior of already-trained models allows to (i) assess the robustness of AI systems; (ii) evaluate their resilience to drifting data perturbations; (iii) verify legal, safety, and security requirements; (iv) complement human-expertise in decision making and even provide scientists with novel insights.

A particularly important separation of interpretability methods is based on the type of algorithm that could be applied: *model-specific* methods can be applied only to some specific models, whereas *model-agnostic* methods can be virtually applied to every possible ML/DL algorithm. Another complementary taxonomy of these methods is based on the *interpretation scale*: *local* methods provide an explanation only for a specific sample, whereas *global* methods attempt to explain the overall model behavior. Accordingly, different *families of approaches* have emerged with the aim of shedding light on ML/DL outcomes via *post-hoc explanations* [5].

Hereinafter, we discuss recent works investigating the interpretation of computer network tasks via the aforementioned XAI techniques. Indeed, in recent years several data-driven solutions based on either ML or DL have been proposed for various network-related problems such as resource allocation, routing, video-rate selection, congestion control [6], traffic classification [7, 8] and prediction [9]. However, the main reasons hindering these solutions to be widely adopted in production environments [10] is their general lack of interpretability as well as potential behavioral uncertainties. Nevertheless, an initial corpus of works has provided a first effort towards the interpretation of data-driven black-box models developed for networking problems as briefly discussed in the following.

Meng et al. [11] propose *Metis*, a framework for interpretability of *local and global control* networking problems. *Metis* exploits decision-trees- and hypergraph-based distillation to obtain interpretable rule-based controllers and support design, debugging, deployment, and *ad-hoc* adjustment of Deep Neural Networks (DNNs). Similarly, Zheng et al. [10] face the *resource allocation* problem by inspecting the behavior of a DNN trained to minimize the average job duration. They utilize *saliency maps* (showing impact of each input feature on the output) and *inspect the activation of intermediate neurons in hidden layers* to reveal what features a DNN depends on. Morichetta et al. [12] define the *EXPLAIN-IT* methodology for the explainability of unlabeled data. *EXPLAIN-IT* provides explanations for clustering results related to the analysis of QoE in YouTube video streaming. The proposed approach uses clustering results to train a classifier for *video quality prediction*, which is then explained via black-box XAI approaches. Dethise et al. [6] analyze the behavior of a model based on reinforcement-learning agents and aiming at *video bit-rate adaptation*. The investigation observes agent's decisions to understand how input features contribute to the decisions. The study exploits data inspection and visualization and leverages *LIME* [13] as XAI strategy. Focusing on *network cyber security*, Amarasinghe et al. [14] propose a framework for *anomaly detection* based on DNNs which provides *post-hoc* explanations for the detected anomalies to improve users' trust. The Layer-wise Relevance Propagation (LRP) method [15] is used to compute the relevance of input features.

More recently, a number of research contributions have investigated *interpretability in traffic identification and classification*. Beliard et al. [7] demonstrate how simple visualization tools (e.g., to represent the 1D original space and feature projections at intermediate layers) can be helpful in clarifying the inference process of Convolutional Neural Networks (CNNs). Wang et al. [8] explore DL methods (autoencoders and convolutional and recurrent networks) for mobile-app TC and use Deep SHAP [2] to explain the outcomes obtained through a 1D-CNN. The analysis focuses only on WeChat app and is limited to four (representative) outcomes, being the approach sample-dependent (viz. a local explanation method). In [9], we leverage Markovian distillation for interpreting traffic prediction results. In more detail, we compare Markov Chains and ML concordant/discordant predictions to highlight and interpret ML predictive patterns (focusing on outcome disagreements) by observing Markovian transition probabilities.

Positioning of our contribution: similarly to the final works previously reported, we focus on TC and utilize XAI techniques to obtain insights regarding the behavior of a (multimodal) DL architecture. Also, we exploit Deep SHAP to infer the *importance* of a set of inputs for certain samples (a *local* approach), as in [8]. However, differently from Wang et al. [8], we use this sample-dependent information to extract *global* explanations, which allows to explain the general correct behavior of the classifier, also in terms of varying input granularity. Moreover, our analysis is not limited to one app, but is related to a selection of the 41 mobile apps encompassing the Mirage-2019 dataset. Finally, for the experimental evaluation, we adopt an *open* dataset, focused on human-operated *mobile* apps. This choice sets our work apart from almost the totality of other works (saved for our own previous contribution on traffic prediction) using private (or partially-private) datasets or simulated data.

III. Multimodal Deep Learning–based Explainable Traffic Classification

In this section, we describe the proposed contribution. Specifically, in Sec. III-A, we describe the Mimetic classifier,

particularly its architecture and training procedure; then, in Sec.III-B, we introduce the concept of interpretability in DL architectures and describe our approach for interpretability based on the Deep SHAP technique.

A. Mobile Traffic Classification via MIMETIC

MIMETIC is a multimodal DL traffic classifier that exploits multiple modalities (viz. views) of traffic data, to capitalize their heterogeneous nature via *intermediate fusion* [16] of (automatically extracted) features.[1] Herein, we leverage XAI techniques to interpret the behavior of the aforementioned state-of-the-art traffic classifier [17]. This classifier operates at *biflow* level: a biflow (viz. a bidirectional flow) is a traffic object encompassing all the packets sharing the same 5-tuple (i.e. source and destination IP address, source and destination port, and transport-level protocol) in both upstream and downstream directions [18]. Hence, the mobile TC task consists of assigning to each biflow a class within the set $\{1, \ldots, L\}$, with L denoting the number of different apps.

MIMETIC consists of two *single-modality* (viz. input-specific) branches that extract the corresponding intra-modality features. Each branch is fed with one input type chosen among *unbiased inputs* [18]. The input of the first branch (henceforth named PAY-modality) is constituted by the first N_b bytes of transport-layer payload arranged in byte-wise format, whereas for the second branch (HDR-modality) we consider some informative header fields of the first N_p packets.[2] In detail, we consider the first $N_b = 576$ bytes and $N_p = 12$ packets, respectively.[3] To capture inter-modality dependencies, the abstract features extracted by the single-modality branches are joined via a concatenation layer and fed to a *shared* dense layer before performing the classification through a softmax. MIMETIC is trained via a two-phase procedure: (i) an independent *pre-training* of each single-modality branch and (ii) a subsequent *fine-tuning* of the whole architecture.

B. Interpreting DL-based Traffic Classifiers via Deep SHAP

The starting point for interpreting complex DL architectures is to consider a simpler explanation model $g(\cdot)$, which is designed to closely-approximate the original model $f(\cdot)$.

In this paper, we focus on *local methods*, which try to explain the original model in the neighborhood of each particular instance $\boldsymbol{x}$, using the so-called *simplified inputs* $\boldsymbol{x'}$ that map to the original ones through a mapping function.

Most of the interpretability techniques assume a peculiar functional form for the explanation model $g(\cdot)$, which led to the definition of Additive Feature Attribution (AFA) methods which are linear functions of binary variables, i.e.

$$g(\boldsymbol{z}^{'}) = \phi_0 + \sum_{m=1}^{M} \phi_m \, z_m^{'} \tag{1}$$

where $\boldsymbol{z}^{'} \in \{0,1\}^M$, M denotes the number of simplified inputs, and $\phi_m \in \mathbb{R}$. AFA methods provide an explanation model attributing an "effect" ϕ_m to each input, and summing the effects of all input attributions approximates the original model output $f(\boldsymbol{x})$. The most used interpretability techniques (LIME, DeepLIFT, etc.) belong to this family of explanation models. Additionally, when the functional form in Eq. (1) is required to satisfy (i) local accuracy, (ii) missingness, and (iii) consistency properties, there exists a unique AFA solution satisfying them [2]. Such solution coincides with the computation of the well-known *Shapley values*.

Shapley values originate from cooperative game theory [19] and identify the contribution of player m to the payoff $v(\mathcal{P})$ achieved by the overall coalition $\mathcal{P}$. To do so, this method assesses the payoff of *every subset* of cooperating players $\mathcal{S} \subset \mathcal{P}$ and tests the effect of removing/adding the player m to $\mathcal{S}$ on the total payoff $v(\mathcal{S})$ obtained by $\mathcal{S}$ if they cooperate.

When transposing the method to the task of explaining a DL-based model, the input data maps into the players of the cooperative game, whereas the DL architecture output $f(\boldsymbol{x})$ corresponds to the payoff function. Unluckily, the time required for the exact computation of Shapley values *grows exponentially* with the input size M.

Conversely, SHapley Additive exPlanation (SHAP) *approximates* these quantities via the conditional expectation [2]

$$f(\boldsymbol{h_x}(\boldsymbol{z}^{'})) \approx \mathbb{E}\left\{f(\boldsymbol{z}) | \boldsymbol{z}_{\mathcal{S}}\right\} \tag{2}$$

where $\mathcal{S}$ denotes the set of non-zero indices within $\boldsymbol{z}^{'}$. This approximation allows to compute the above quantity in a computationally-efficient fashion and eliminates the need to re-train the models. The computation is also simplified by assuming statistical independence among the inputs and linearity of the model [2], i.e. $f(\boldsymbol{x}) = \sum_{m=1}^{M} w_m x_m + b$. Indeed, when both these assumptions hold, it can be shown that ϕ_m's are in *closed-form* and equal to $\phi_m(f, \boldsymbol{x}) = w_m \, (x_m - \mathbb{E}\{x_m\})$. The aforementioned simplification for ϕ_m's is capitalized as described next.

Herein, we rely on DeepLIFT [20] for the explicit and recursive computation of SHAP values. Indeed, the latter is an AFA DL-explanation method which attributes to each input x_m a value $C_{\Delta x_m \Delta o}$ that represents the effect of that input being set to a reference value (an uninformative background value) as opposed to its original value. Accordingly, DeepLIFT can be used to obtain a compositional approximation of SHAP values (i.e. using output expectation as a reference value and resorting to explicit Shapley equations, for consistent linearization), leading to *DeepSHAP* [2], a fast-approximation of SHAP values. Specifically, this approach will be used to assess the importance of inputs selected from the raw traffic data (e.g., related to PAY or HDR modalities of MIMETIC) of a given biflow in classifying the app generating it.

Therefore, to explain the predictive behavior of DL-based traffic classifiers, the prediction model $f(\boldsymbol{x})$ is chosen as the soft-output associated to the generic i^{th} app, i.e. $p_i(\boldsymbol{x})$. Hence, we interpret the SHAP value ϕ_m as the *importance value* of the m^{th} input in forming the confidence associated to labeling the

[1] This peculiar procedure optimizes the less sophisticated *early* (or *data*) *fusion* and *late* (or *score/decision*) *fusion* that are not able to fully exploit the potentiality of multi-modality.

[2] Number of bytes in transport-layer payload, TCP window size (set to zero for UDP packets), inter-arrival time, and packet direction $\in \{-1, 1\}$.

[3] We underline that these choices have been driven by both our past experience [3, 18] and further preliminary analyses (not shown for brevity).

biflow (whose overall input is $\boldsymbol{x}$) with the i^{th} app. We recall that, since ϕ_m can be also negative, they should be interpreted as follows: positive (negative) values increase (decrease) the confidence $p_i(\boldsymbol{x})$ in the i^{th} app with respect to its average value $\mathbb{E}\{p_i\}$.[4] Herein, for each biflow, we focus on explaining the soft-output associated to the predicted app $\hat{p}(\boldsymbol{x})$, as this represents the most relevant (and highest) output for TC.

Once we have obtained a local explanation for a single instance, our proposed *global explanation* approach relies on aggregating (viz. pooling) explanations over different samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$. The aggregation step is however carried out on *normalized* SHAP values, obtained by dividing each SHAP value by their overall sum, namely $\widetilde{\phi}_m \triangleq \phi_m \,/\, \sum_{m=1}^{M} \phi_m$. Considering $\widetilde{\phi}_m$ allows focusing on the relative *importance* of each input (indeed, for each sample, the sum of the importance values equals one). Additionally, as in [8], we aggregate only on *correctly-classified samples* to focus on the correct behavior of MIMETIC and to allow to interpret its counter-intuitive (while right) decisions a posteriori.

IV. EXPERIMENTAL ANALYSIS

This section describes the experimental setup and interpretability results. Specifically, in Sec. IV-A a brief description of the MIRAGE-2019 dataset is provided. Then, the interpretability of MIMETIC TC-results is discussed, focusing on relative contribution of each modality, and per-modality investigation of corresponding inputs (Sec. IV-B). Finally, referring to the PAY-modality, an in-depth TLS-based analysis is performed (Sec. IV-C).

A. Dataset Description and Experimental Setup

This study leverages MIRAGE-2019, an open dataset containing mobile-app traffic.[5] 280+ human experimenters[6] (mimicking the typical usage of each app by testing most-used functionalities) contributed to the collection of the dataset at the ARCLAB laboratory of the University of Napoli "Federico II" within May'17–May'19, exploiting the MIRAGE architecture [3]. As a whole, MIRAGE-2019 encompasses the traffic generated by 41 Android apps (newest app versions available at time of capture were used) belonging to 16 categories according to the Google Play Store[7]. The dataset contains $\geq$ 4.6k traffic traces, each generated in a capture session of $5 \div 10$ mins, and $\approx$ 96.5k biflows.

All the results in the following refer to a random (stratified) dataset split with 90% (resp. 10%) instances allocated for training (resp. testing). We highlight that, in the considered case, the TC performance attained by MIMETIC were 88.74%, 87.83%, and 93.43% in terms of accuracy, F-measure, and G-mean, respectively.

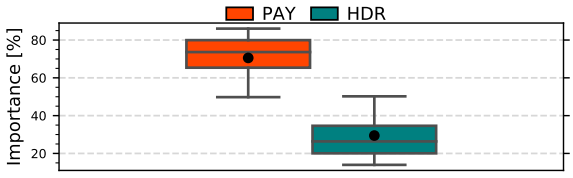


Figure 1. Modality contributions of MIMETIC in terms of importance $\widetilde{\phi}_m$. PAY-modality contributes with remarkably higher importance values than HDR-modality.

B. Interpretability Analysis for TC Decisions

In Fig. 1, we report the (relative) contribution of each traffic modality (HDR or PAY) of MIMETIC. The results are obtained by aggregating (over all the correctly-classified tested samples), via a median statistic, the pooled SHAP values $\widetilde{\phi}_{\mathcal{M}}$ for each modality, obtained via DeepSHAP (cf. Sec. III-B). It is apparent that PAY-modality *always* contributes with higher importance values. Finally, we would also underline that the complementary analysis performed on a per-app basis (whose results are not shown for brevity), highlights analogous patterns, remarking the major importance of PAY-modality over the HDR one.

Hereinafter we firstly focus on the importance of inputs associated with the HDR-modality of MIMETIC architecture. Specifically, this modality is fed with 4 header fields extracted from the first 12 packets of each biflow, namely inter-arrival time (IAT), direction (DIR), TCP window size (TCP_WS), and transport-layer payload length (PL). With this analysis, we provide global explanations at app granularity. To this aim, in the following we discuss the results obtained for some exemplifying apps.

Figure 2 depicts the median importance values for each element of the 4×12 matrix constituting the input of HDR-modality. For Facebook (Fig. 2a), we can notice that the importance is mainly related to PL and TCP_WS with a decreasing importance of the packets after the very first ones (this latter observation applies to all the fields but IAT which is characterized by a more flat distribution). Overall, we can notice that the first four packets are the most important. A similar situation is observed for OneDrive (Fig. 2b): TCP_WS and PL are the most important fields, whereas the other two fields expose very minimal median scores, regardless of the packet number. For the other apps, it is more challenging to recognize the most important fields: for instance, matrix elements of Hangouts (Fig. 2c) have no remarkable higher values, beyond TCP_WS of the 2^{nd} packet and PL of the initial ones (i.e the first 5 packets). Similarly, the matrix of ilMeteo (Fig. 2d) results scattered without fields with remarkably major importance. A different situation is observed for Skype and Trello. In the former case, DIR of the packets around the 7^{th} is useful for the prediction. On the other hand, IAT plays an essential role together with DIR for Trello. These results

[4]The sum of the SHAP values equals the considered soft-output value minus the *base output*. The latter represents the average of the same soft-output obtained in correspondence of the samples associated to the background set.

[5]http://traffic.comics.unina.it/mirage

[6]Experimenters have been informed about the objectives of their activities and the public release of the corresponding traces for research purposes. Also, no personal information was involved at any time of MIRAGE-2019 collection (i.e. non-personal mobile devices, private IPv4 address space, purposedly-created app accounts) [3].

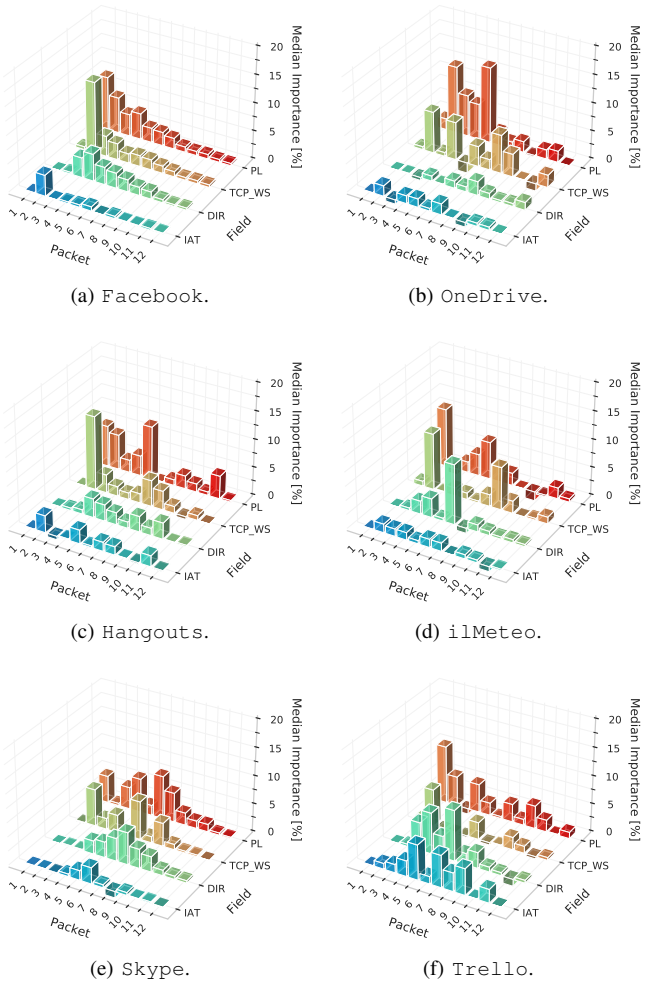
[7]https://play.google.com/store/apps

confirm that all the considered fields can play a crucial role in identifying the correct app generating the observed biflows, even if with some differences among the apps.

Now we discuss the importance of inputs associated to the PAY-modality, which relies on the first $N_b = 576$ bytes of each biflow. In all the plots, sample-wise positive and negative SHAP values are highlighted with red and blue colors, respectively. These distributions are integrated with the median importance value of each byte (over different samples), which is reported as a solid black line. This helps understanding regions that are more consistently influential for predictions and assessing their variability.

In Fig. 3, we analyze the global explanations associated with single apps, focusing on some remarkable examples which are discussed in the following. For `Dropbox` (Fig.3a), we notice an initial region attaining high per-sample positive values witnessing that the first 200 bytes play, on average, a crucial role in correctly identifying this app. Such an outcome is also confirmed by the median value. On the opposite, the median of the distribution for the other bytes is very low, suggesting their poor relevance. However, the situation is not always so neat: for a small group of apps (e.g., `Spotify`—Fig. 3b), it is hard to identify regions that are consistently influential, with the median that does not highlight specific groups of bytes. For these apps, we often observe more dense blue distributions, reporting the presence of input values that confuse the network. We remark that this situation concerns a limited group of apps.

Finally, for the other apps including `Trello` (Fig. 3c), the observed behavior is different; in such cases, we can clearly observe a central region (corresponding with bytes in the interval $[300; 400]$B) that supports correct predictions. This observation underlines the importance of considering a value for N_b no less than 400B (as in MIMETIC), i.e. not limited to the very first bytes of the payload.



(a) `Facebook`. (b) `OneDrive`. (c) `Hangouts`. (d) `ilMeteo`. (e) `Skype`. (f) `Trello`.

Figure 2. Importance $\tilde{\phi}_m$ of the header-field inputs (identified by the respective packet) for `HDR`-modality of MIMETIC for exemplifying apps.

C. Analysis of TLS Biflows

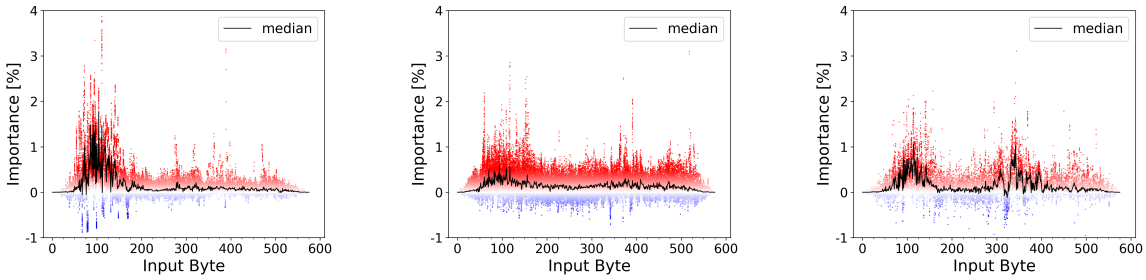
In the following analysis, we focus on TLS biflows (which altogether constitute $\approx 80\%$ of the dataset) and investigate the benefits to TC effectiveness deriving from *Server Name Indication (SNI)* extension. We focus on the latter because the bytes highlighted by DeepSHAP belong to this particular extension in many local (viz. per-sample) explanations.

In detail, the SNI allows to specify which hostname the client is attempting to connect to. Apart from a few biflows not using this extension (*None*), we can divide the biflows in our dataset into two groups: those (i) with *App-Specific SNI* (i.e. used only by a particular app) and those (ii) with *Common SNI* (found in more than one app, by definition). Concerning the latter, the most used is `www.google.it`, shared by 28 apps, while the majority of the Common SNIs are related to Google and Facebook domains. On the other hand, the presence of peculiar SNI-related information (i.e. App-Specific SNIs) is expected to carry remarkable benefits.

To provide a quantification, we have evaluated the performance of MIMETIC's PAY-modality for biflows with either App-Specific SNIs or Common SNIs. Results witness that TC accuracy appreciably differs between the two aforementioned groups: for the former, it is surprisingly high, i.e. $\approx 94\%$, whereas considering the latter, it decreases to $\approx 69\%$.

In order to highlight the distribution of different SNI-type-carrying biflows, Fig. 4 reports their subdivision for each app. In detail, four apps, namely `Waze`, `Dropbox`, `eBay`, and `Diretta`, show a very high App-Specific SNI rate ($> 80\%$), with `Waze` having $\approx 95\%$ of biflows exposing SNIs found only in biflows of the same app. For the other apps, this rate gradually decreases down to `Slither.io` which reports exclusively Common SNIs.

As expected, the share of biflows with App-Specific SNIs impacts the per-app classification performance. Focusing on the above-mentioned four apps exposing the major share of these biflows, the resulting accuracy lies in the range $[92\%; 98\%]$ (up to $+10\%$ w.r.t. the average accuracy over the whole dataset). On the other hand, considering the other side of the spectrum, `Slither.io` (whose traffic is characterized by the sole presence of Common-SNI biflows) results in 84%



(a) Dropbox. (b) Spotify. (c) Trello.

Figure 3. Importance $\widetilde{\phi}_m$ of the payload-byte inputs (identified by their position) for PAY-modality of MIMETIC for exemplifying apps.

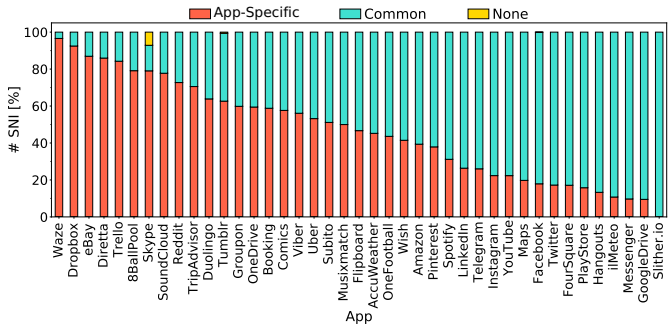


Figure 4. App-Specific and Common SNI distribution for the 41 apps composing the MIRAGE-2019 dataset.

accuracy (-4% w.r.t. the average).

While these results cannot be exclusively imputed to the SNI values observed in the biflows neither to the sole PAY-modality, they further highlight the centrality of SNI-related information, which are fruitfully leveraged by the TC models.

V. Conclusions and Future Directions

We focused on explaining the black-box behavior of DL-based traffic classifiers, exploiting our previous state-of-the-art proposal MIMETIC, using tools from XAI domain, and evaluating our methodology on an open dataset of mobile-app traffic (encompassing 41 apps). Based on DeepSHAP, we were able to obtain *global explanations* (from the interpretability viewpoint) which allowed us to assess the weight of each modality and the importance of input data fed to HDR (resp. PAY) modality, quantifying the importance of each header field (resp. payload byte). Delving into the latter aspect, we performed a domain-based TLS analysis focusing on SNI to reach a human-understandable interpretation of these results. Future avenues of research will include (i) investigating trustworthiness of traffic classifiers (via calibration analysis), (ii) comparing global explanations obtained via other XAI tools (e.g., LRP [15]), and (iii) applying XAI techniques to other network traffic analysis tasks (e.g., prediction).

References

[1] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Toward effective mobile encrypted traffic classification through deep learning," *Neurocomputing*, vol. 409, pp. 306–315, 2020.

[2] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *NeurIPS'17*.

[3] G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapé, "MIRAGE: Mobile-app traffic capture and ground-truth creation," in *4th IEEE ICCCS'19*.

[4] H. Hagras, "Toward human-understandable, explainable AI," *IEEE Computer*, vol. 51, no. 9, pp. 28–36, 2018.

[5] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K. R. Müller, "Explaining deep neural networks and beyond: A review of methods and applications," *Proc. IEEE*, vol. 109, no. 3, pp. 247–278, 2021.

[6] A. Dethise, M. Canini, and S. Kandula, "Cracking open the black box: What observations can tell us about reinforcement learning agents," in *ACM NetAI'19*.

[7] C. Beliard, A. Finamore, and D. Rossi, "Opening the deep pandora box: Explainable traffic classification," in *IEEE INFOCOM WKSHPS'20*, pp. 1292–1293.

[8] X. Wang, S. Chen, and J. Su, "Real network traffic collection and deep learning for mobile app identification," *Hindawi Wireless Communications and Mobile Computing*, 2020.

[9] G. Aceto, G. Bovenzi, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapé, "Characterization and prediction of mobile-app traffic using Markov modeling," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 907–925, 2021.

[10] Y. Zheng, Z. Liu, X. You, Y. Xu, and J. Jiang, "Demystifying deep learning in networking," in *ACM APNet'18*.

[11] Z. Meng, M. Wang, J. Bai, M. Xu, H. Mao, and H. Hu, "Interpreting deep learning-based networking systems," in *ACM SIGCOMM'20*.

[12] A. Morichetta, P. Casas, and M. Mellia, "EXPLAIN-IT: Towards explainable AI for unsupervised network traffic analysis," in *ACM CoNEXT Big-DAMA'19*.

[13] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *ACM SIGKDD KDD'16*.

[14] K. Amarasinghe, K. Kenney, and M. Manic, "Toward explainable deep neural network based anomaly detection," in *IEEE HSI'18*.

[15] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, pp. 1–46, 2015.

[16] D. Ramachandram and G. W. Taylor, "Deep multimodal learning: A survey on recent advances and trends," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 96–108, 2017.

[17] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapè, "MIMETIC: mobile encrypted traffic classification using multimodal deep learning," *Elsevier Computer Networks*, vol. 165, p. 106944, 2019.

[18] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using Deep Learning: Experimental evaluation, lessons learned, and challenges," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 2, pp. 445–458, 2019.

[19] L. S. Shapley, "A value for n-person games," *Contributions to the Theory of Games*, vol. 2, no. 28, pp. 307–317, 1953.

[20] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *ICML'17*.