XAI for Interpretable Multimodal Architectures with Contextual Input in Mobile Network Traffic Classification

Francesco Cerasuolo, Idio Guarino, Vincenzo Spadari, Giuseppe Aceto, Antonio Pescapé
University of Napoli Federico II
{francesco.cerasuolo, idio.guarino}@unina.it, v.spadari@studenti.unina.it, {giuseppe.aceto, pescape}@unina.it

***Abstract*—Understanding network traffic patterns is crucial in today's interconnected world, particularly with the increasing use of communication and collaboration apps (CC apps). These allow different *user activities* (chat, audio, video), resulting specifically hard to classify. Despite promising results of DL for traffic classification (TC), its opacity poses challenges in understanding the decision-making process and hinders its widespread adoption.**

To cope with these limitations, we leverage eXplainable AI (XAI) techniques to *open the "black box"* and interpret Mimetic-All, a state-of-art multimodal DL architecture. Our goal is to analyze joint app-activity classification in an *early fashion*—viz. with the first packets of each bidirectional communication—while evaluating the contribution of each traffic input—*packet header fields* (`SEQ`), *payload bytes* (`PAY`), and *contextual inputs* from concurrent biflows (`Context`)—to the final prediction. Our findings show that although the inclusion of `Context` enhances performance, its importance is relatively lower w.r.t. `SEQ` and `PAY` in TC. Moreover, within `PAY`, specific byte subsets are identified as more influential for TC compared to others, whereas in `SEQ`, the length of transport-level payload holds greater importance than other header fields.

***Index Terms*—communication apps; deep learning; encrypted traffic classification; multimodal approaches; XAI.**

I. Introduction

In today's interconnected world, where digital communication permeates every aspect of our lives, understanding the related network traffic has become paramount. Recently, communication and collaboration apps (CC apps) have attested as one of the primary categories contributing to $\approx 23\%$ of uplink traffic [1]. This phenomenon arises from the recent COVID pandemic and the associated lock-downs, when these services became essential for communication. Now, they have become integral parts of daily routines. Furthermore, a 2023 market analysis predicts that global users of CC apps will reach about 7 billion by 2030 (+92% w.r.t. 2022) [2]. This growth, along with the diversification of applications and services, has increased the complexity and volume of network traffic, necessitating advanced management techniques.

Traffic Classification (TC)—i.e., the process of categorizing traffic based on its characteristics—has emerged as a crucial endeavor for various purposes ranging from network management to security enforcement. In such a scenario, dealing with CC apps is even more complex because of their *multi-activity* nature—i.e., they commonly generate video, voice, chat, or game content traffic [3]. Hence, the research landscape in this domain has experienced significant expansion, evolving from Machine Learning (ML) to Deep Learning (DL) methodologies and further advancing into multi-modal architectures, leveraging various *network traffic views* to optimize performance [4]. Despite their efficacy, a notable hindrance to the widespread adoption of such approaches is their *"black box"* nature, which renders the decision-making process inscrutable. This issue is also underscored by established guidelines for AI systems [5] and European AI Act [6], which aim to ensure the safe and reliable use of AI for societal advancement.

ISBN 978-3-903176-63-8 ©2024 IFIP

To tackle this challenge, eXplainable Artificial Intelligence (XAI) offers a solution to elucidate AI-based models, demystify their operations, and dismantle their "black box" structure, fostering transparency and trust in these systems.

The **main contributions** of this paper lie in employing XAI techniques to explain multimodal architectures. Specifically, we employ Deep SHAP, a widely used feature attribution method to assign an *importance* to each input in determining model output. We leverage `MIRAGE-COVID-CCMA-2022`—a recently collected and publicly-available dataset encompassing traffic of 9 popular CC apps and providing app-activity ground truth—and Mimetic-All—a multimodal architecture tailored for joint app-activity classification.

The rest of the paper is organized as follows. Section II summarizes current advancements in multimodal architecture for TC and XAI methods for DL models. Next, Sec. III outlines our methodology, while Sec. IV details the experimental setup. Results are presented in Sec. V. Finally, Sec. VI concludes the paper, also discussing potential future directions.

II. Related Work

In this section, we analyze various state-of-the-art (SOTA) works employing XAI in the domain of TC. We present these works in Tab. I, outlining for each work the *networking problem* addressed, the *interpretability* of the proposed solution, the use of *multimodal architecture* along with the *modalities* employed, the classification of *apps* and *activities*, and the *XAI methodology* used. The last row summarizes the present work, whose positioning w.r.t. SOTA is discussed at the end of this section.

Most works employ mobile datasets and perform classification or prediction tasks. Only a few employ multimodal ar-

Table I: Related work, listed in chronological order, employing XAI for various networking problems. Last row describes current proposal.

Paper	Year	Dataset	Networking Problem	Interpretability	MM	Modalities	App-TC	Act-TC	XAI Method
Amarasinghe et al. [7]	2018	🕵	AD	●	○	-	○	○	LRP
Dethise et al. [8]	2019	🎞	TP	●	○	-	○	○	LIME
Morichetta et al. [9]	2019	▶	VQP	●	○	-	○	○	LIME
Rezaei et al. [10]	2019	📱	TC	●	○	-	●	○	Occlusion
Beliard et al. [11]	2020	🌐	TC	●	○	-	○	○	t-SNE F.Map Visual.
Wang et al. [12]	2020	📱	TC	●	○	-	●	○	Deep SHAP
Aceto et al. [13]	2021	VPN	TC	○	●	PB, HF	●	◐	Calibration
Aceto et al. [14]	2021	📱	TP	○	○	-	○	○	Markov-Distill.
Akbari et al. [15]†	2021	🌐 G	TC	●	●	THB, FTS, FS	●	◐	Occlusion
Montieri et al. [16]	2021	📱	TP	○	○	-	○	○	Markov-Distill.
Nascita et al. [17]	2021	📱	TC	●	●	PB, HF	●	○	Deep SHAP, Calibration
Sadeghzadeh et al. [18]	2021	VPN	TC	●	○	-	○	◐	Perturbation
Fauvel et al. [19]	2022	🖧	TC	●	○	-	●	○	X-DL Arch.
Guarino et al. [20]	2022	📱	TC	○	●	PB, HF, CF	●	●	Calibration
Guarino et al. [21]	2024	📱	TP	●	○	-	○	○	Deep SHAP, Calibration
This Paper	2024	📱	TC	●	●	PB, HF, CF	●	●	Deep SHAP

Dataset: Malware (🕵), Video (🎞), Youtube (▶), Mobile (📱), VPN (VPN), Internet (🌐), Google (G), Wired (🖧); **Networking Problem:** Anomaly Detection (AD), Video Quality Prediction (VQP), Traffic Classification (TC), Traffic Prediction (TP); **Multimodal (MM)**; **XAI Method:** Layer-wise Relevance Propagation (LRP), Feature Map Visualization (F.Map Visual.), Markovian-Distillation (Markov-Distill.), Explainable-by-Design DL Architecture (X-DL Arch.); **Modalities:** Flow-Time Series (FTS), TLS Handshake Bytes (THB), Flow Statistics (FS), Payload Bytes (PB), Header Features (HF), Contextual Features (CF); ● present, ◐ partially present, ○ lacking; †: disjoint evaluation on non-mobile app and mobile service traffic.

chitectures, typically leveraging payload bytes, flow statistics, or features extracted from packet headers. While most focus on app classification, only a minority address a task similar to activity classification, with some specifically categorizing VPN and non-VPN services. Furthermore, many of these works employ interpretability techniques to explain the decision-making process. As for XAI methodology, several works focus on the trustworthiness of their proposed solution by performing a calibration analysis. Others perform various form of *post-hoc explanation*: (a) layer-wise relevance propagation (*LRP*), (b) interpretable local surrogates via *LIME*, (c) different type of *perturbation* analysis, (d) importance attribution based on *Shapley values*. Additionally, some use visual representations (e.g., *t-SNE*, *Feature Maps*) to highlight significant decision-contributing features or *Markovian Distillation* to interpret predictions based on Markov Chains' transition probabilities. Lastly, others propose *explainability-by-design* architecture and compare input data with prototypes specific to each class.

Positioning. In this work, *we address the inherent lack of interpretability of DL models for joint app and activity classification (viz. Joint-TC).* We offer *interpretability* through XAI techniques starting from the methodology proposed by Nascita et al. [17]. Our approach involves conducting a comprehensive analysis of DL models using Deep SHAP to provide interpretable results. In detail, we adapted such methodology to interpret Mimetic-All [20], which classifies both app and activity and leverages an additional modality based on Context Inputs extracted from "contextual" traffic.

III. Methodology

This section delves into the methodology employed to explain multimodal DL architectures in a Joint-TC task. We recall that Joint-TC involves jointly classifying both the app and user activity performed. To this end, in Sec. III-A, we introduce Mimetic-All [20], a multimodal architecture tailored for this task. Then, we outline the technique to interpret the predictions of multimodal architectures in Sec. III-B.

A. SOTA Multimodal Architectures for TC: Mimetic-All

Mimetic-All address Joint-TC task at the biflow-level[1] based on its first packets (viz. *early* TC). It consists of *three per-modality* branches, named bPAY, bSEQ, and bContext, each fed with a specific input representing a different "view" on network traffic: (i) bPAY takes as input the first N_b bytes of the transport-layer payload (PAY); (ii) bSEQ takes as input informative fields extracted from the sequence of the first N_p packets (SEQ); (iii) bContext is fed with contextual inputs (viz. Context) that are obtained by aggregating information from the biflows concurrent with the target biflow (B_r) until the arrival of its N_p-th packet (viz. contextual biflows). Contextual biflows are selected according to defined criteria to ensure causality and that only relevant contextual information is considered.

B. Interpreting Multimodal Architectures: SHAP

In this section, we explain the methodology for interpreting a DL model $f(\cdot)$ for a probabilistic TC task. We used a simpler explanation model $g(\cdot)$—that closely approximates $f(\cdot)$—to evaluate the soft output for the generic i^{th} class, denoted as $p_i(\cdot)$, and identify the inputs with the most significant impact on the confidence probability value.

[1] A bidirectional flow (biflow) encompasses all the packets sharing the same 5-tuple (i.e., IP$_{src}$,IP$_{dst}$,port$_{src}$,port$_{dst}$, proto$_{L4}$) with interchangeable source and destination [22].

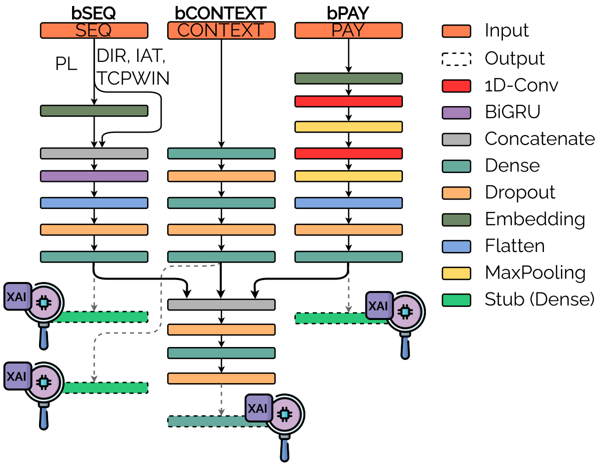


Fig. 1. MIMETIC-ALL classifier. Colors indicate the layer type.

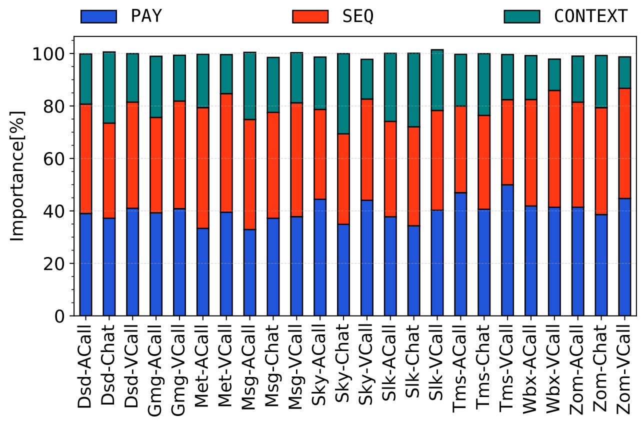


Fig. 2. Median importance of each input (i.e., PAY, SEQ, and Context) for correctly classified samples according to the corresponding $\langle app, activity \rangle$.

Our approach involves using *local* methods to explain $f(\boldsymbol{x})$ in the neighborhood of a specific instance $\boldsymbol{x}$. It is worth noting that, in the case of TC, this results in a per-flow explanation and uses simplified inputs $\boldsymbol{x}'$ that are mapped to the original inputs $\boldsymbol{x}$ through the mapping $\boldsymbol{x} = \boldsymbol{h}_{\boldsymbol{x}}(\boldsymbol{x}')$.

We leverage Additive Feature Attribution (AFA) as $g(\cdot)$ for determining the contribution (viz. importance) $\phi_m \in \mathbb{R}$ of each input towards the output of $f(\boldsymbol{x})$. To compute AFA solutions, we use *SHapley Additive exPlanation* (SHAP), an approximation method that estimates Shapley values using conditional expectation. We use DEEP SHAP, a rapid approximation algorithm, to explain the soft-output for the predicted class, denoted as $\hat{p}(\boldsymbol{x})$. Accordingly, positive (resp. negative) values increase (resp. decrease) the confidence in the i^{th} class compared to its average value. We aggregate local explanations to achieve *global* explanations. Our method involves computing range-normalized SHAP values, i.e., $\widetilde{\phi}_m \triangleq \phi_m \,/\, \sum_{m=1}^{M} \phi_m$, due to soft output variability. This enables us to derive measures of importance independent of specific confidence levels over test samples.

In the following, we use DEEP SHAP to analyze the importance of different inputs in the MIMETIC-ALL architecture. The inputs include PAY, SEQ, and Context, and $\hat{p}(\boldsymbol{x})$ refers to the specific app and its activity.

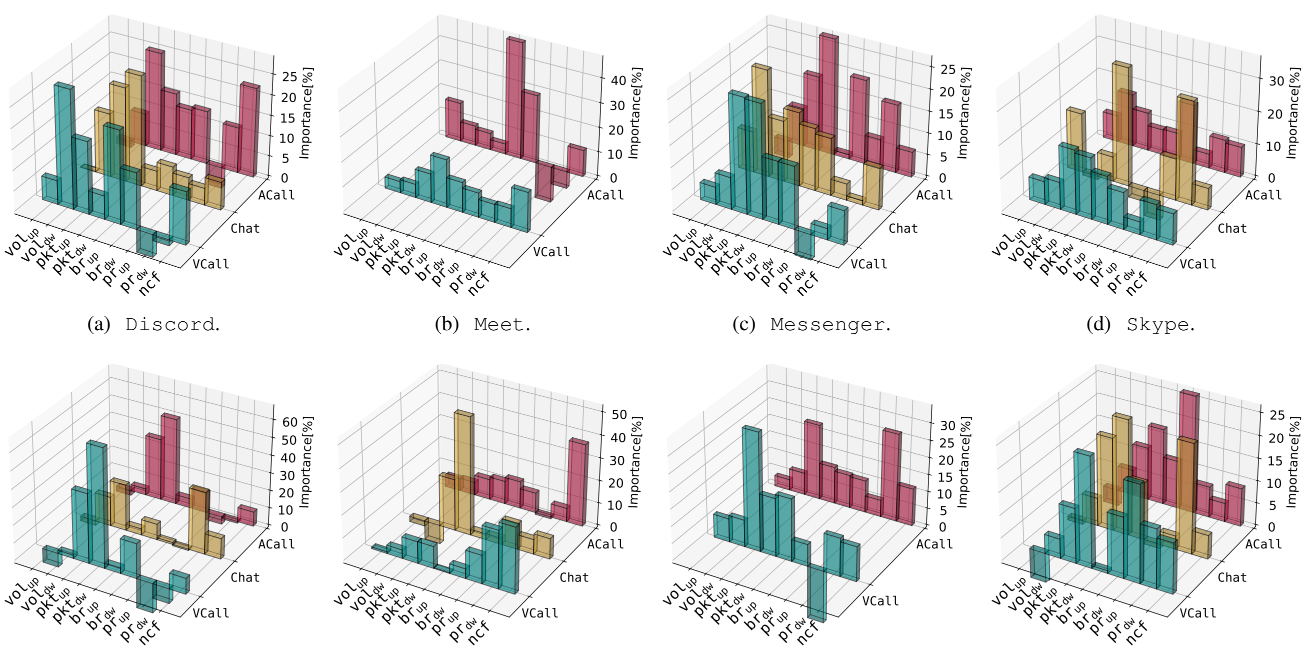
IV. EXPERIMENTAL SETUP

The MIRAGE-COVID-CCMA-2022 dataset[2], collected between April and December 2021 at the ARCLAB laboratory University of Napoli “Federico II”, is leveraged for experiments since it provides ground truth at both the app and user activity levels. It encompasses 9 CC apps (in brackets the abbreviations for each them): Discord (Dsd), GotoMeeting (Gmg), Meet (Met), Messenger (Msg), Skype (Sky), Slack (Slk), Teams (Tms), Webex (Wbx), and Zoom (Zom). For each app, traffic related to different *activities* is collected according to the app usage. These activities included (i) *Chat* (Chat)—entails two participants exchanging textual messages and/or multimedia content (e.g., images or GIFs); (ii) *Audio-call* (ACall)—involving only two participants, this activity consists of transmitting audio exclusively; (iii) *Video–call* (VCall)—involves multiple attendees who can transmit both video and audio (e.g., video calls or webinars).

For experiments, we borrow the setup employed in [20]. Specifically, for bPAY branch, we use the first $N_b = 576$ payload-bytes as input. For bSEQ branch, we use the sequences of the transport-layer payload length (PL), TCP window size (TCPWIN), inter-arrival time (IAT), and packet direction (DIR) [3] for the first $N_p = 20$ packets as SEQ input. As input of bCONTEXT branch, we aggregated data from contextual biflows of BF_r to compute 9 metrics: *(a)* number of contextual biflows (ncf), *(b)* volume of transmitted bytes/packets (vol$_*$/pkt$_*$), and *(c)* bit/packet-rate (br$_*$/pr$_*$), in both upstream ($*$=up) and downstream ($*$=dw).

From an architectural viewpoint (see Fig. 1), bPAY consists of two 1D-Conv layers, each followed by a max-pooling layer, and a final Dense layer. In contrast, bSEQ includes a BiGRU layer followed by a Dense layer. Lastly, bCONTEXT is a Multi-Layer Perceptron network with three Dense layers. [4] The features extracted by the single-modality branches are joined via a concatenation layer and fed to a (Dense) *shared representation layer* before performing the classification through a softmax. MIMETIC-ALL is trained via a two-phase procedure: (i) an independent *pre-training* of each modality branch followed by (ii) a *fine-tuning* of the whole architecture. The pre-training of the p^{th} modality branch is achieved by means of a *softmax stub*. This setup maintains consistent class sample ratios across each fold. For further information on dataset collection, hyper-parameters, and training strategy, please refer to [20].

[2] http://traffic.comics.unina.it/mirage/mirage-covid-2022



(a) Discord. (b) Meet. (c) Messenger. (d) Skype.

(e) Slack. (f) Teams. (g) Webex. (h) Zoom.

Fig. 3. Median importance of Context Inputs for correctly classified samples. Results refer to Discord (a), Meet (b), Messenger (c), Skype (d), Slack (e), Teams (f), Webex (g), and Zoom (h), based on the activity type.

V. Experimental Evaluation

In the following, we discuss the interpretability of Mimetic-All predictions when dealing with the Joint-TC task. Specifically, we first analyze the relative contribution of each modality (viz. branch) for the final classification relying on Deep SHAP[5] , then thoroughly investigate the inputs corresponding to each of them. In more detail, to achieve *per-modality interpretation*, we examine the *stub output* associated with the p^{th} modality. This approach isolates the considered modality from the influence of other modalities and from the combined effect of intermediate fusion achieved by the shared representation layers. To clarify further, our focus lies on test samples correctly classified [17]. This approach allows us to prioritize the correct functioning of a DL-based traffic classifier and then interpret its occasionally counter-intuitive (yet correct) decisions.

Accordingly, Fig. 2 reports the median importance of each modality (i.e., bPAY, bSEQ, and bContext) of Mimetic-All w.r.t. to each $\langle app, activity \rangle$. Overall, bPAY and bSEQ branches have more importance (30-50%) w.r.t. the bContext one (12-30%). Notably, for 5 apps (i.e., Discord, GotoMeeting, Slack, Webex, and Zoom), both bPAY and bSEQ contribute similarly regardless of activity. However, for Skype, while bPAY and bSEQ have similar importance for Chat, this does not hold for ACall and VCall, where the former modality is more important. Furthermore, we observe that for Teams (resp. Meet and Messenger), bPAY is more (resp. less) important than bSEQ regardless of activity. Finally, focusing on bContext, we observe that in almost all cases, its importance is $\geq 20\%$. However, when dealing with VCall for Webex and Zoom, its importance falls in the range $[12, 15]\%$.

Takeaway: *Among* Mimetic-All *branches, b*PAY *and b*SEQ *are more important than b*Context *for Joint-TC, with importance values in* $[30, 50]\%$ *and* $[12, 30]\%$*, respectively.*

A. How do Context Inputs affect Joint-TC?

Fig. 3 depicts the median importance of Context, providing a per-activity breakdown for different apps.

Overall, we observe that in all apps and activities, the packets exchanged in both directions ($\mathtt{pkt_{up}}$ and $\mathtt{pkt_{dw}}$) always contribute positively to the accurate prediction. Considerably, these features have a higher importance value than others, especially for activities related to Discord, Messenger, Slack, Webex, and Zoom. For almost all apps, we observe that ncf assumes significant relative importance, particularly when classifying ACall and VCall.

In contrast, traffic volumes generally have less relative importance and may be detrimental to the final outcome. For instance, $\mathtt{vol_{up}}$ has negative importance on ACall (resp. Chat) for Discord and Messenger (resp. Teams). Notably, regardless of the activity, $\mathtt{vol_{up}}$ always has a negative effect on activities of Slack and Zoom. However, the opposite holds for Meet and Messenger.

[3] TCPWIN is set to zero for UDP packets while DIR $\in \{-1, 1\}$.

[4] Since LeakyReLU is not fully supported by Deep SHAP Python library, we have replaced it with a ReLu.

[5] We used a *background set* of 500 randomly selected training samples used by Deep SHAP to determine a *reference value* for the explanations.

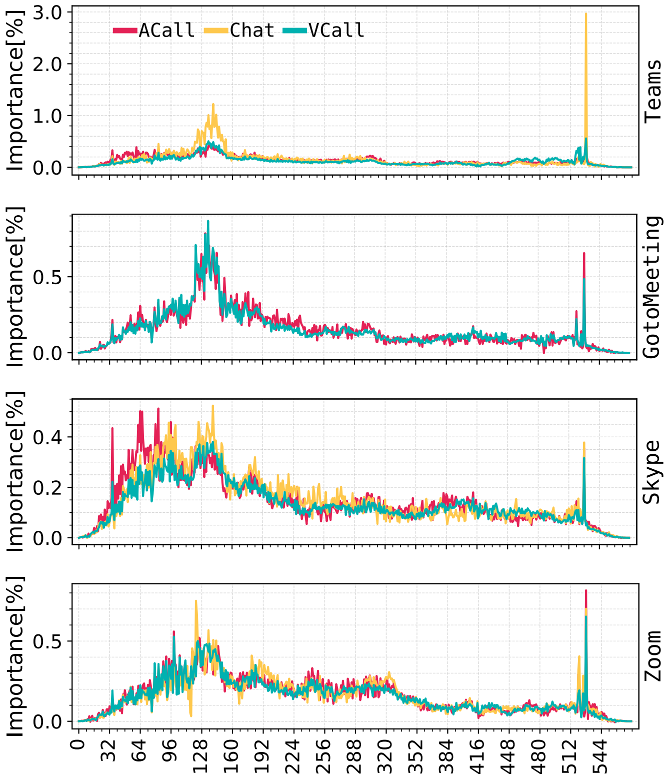


Fig. 4. Median importance of PAY inputs (identified by their index) for correctly classified samples. Results refer to Teams, GotoMeeting, Skype, and Zoom, based on the activity type.

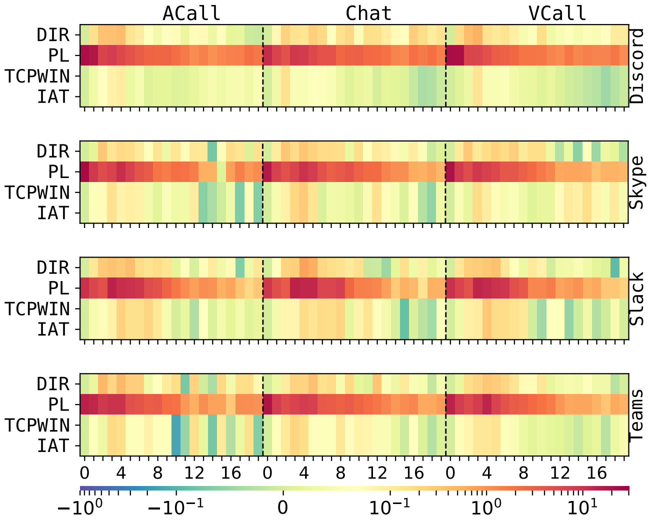


Fig. 5. Median importance (in Symlog-scale) of SEQ inputs (identified by their packet index) for correctly classified samples of Discord, Skype, Slack, and Teams, based on the activity type.

Finally, focusing on bit- and packet-rates, the importance varies greatly depending on the app, activity, and direction. Specifically, while bit rates always positively impact the final decision, packet rates generally have a negative effect, especially when dealing with VCall (resp. ACall) on Discord, Slack, and Webex (resp. Meet). $\texttt{pkt}_{\texttt{dw}}$ has overall non-negligible importance in the case of Chat, especially for Skype, Slack and Zoom.

Takeaway: *About Context, the number of exchanged packets and contextual biflows mostly affect the accurate prediction. However, exchanged traffic volumes may lead to incorrect decisions. The importance of upstream and downstream bit- and packet-rate varies depending on the app and activity.*

B. How do payload bytes affect Joint-TC?

Fig. 4 shows the importance of PAY inputs—consisting of the first $N_b = 576$B of the biflow—on bPAY by reporting the activity breakdown for different apps, including Teams, GotoMeeting, Skype, and Zoom [6]. Specifically, we report the per-activity breakdown of the median importance value of each byte composing PAY input for a given app.

As shown, while all bytes positively affect the final outcome, their importance varies by app and activity, which is highlighted by different regions. Bytes from 32^{nd} to 320^{th} generally are the most important, indicating that they are crucial to correctly classifying the app and activity. A peak around 528^{th} byte is experienced for all apps, sometimes obtaining higher importance w.r.t. other bytes (e.g., for Zoom and Teams). On the other hand, the first 32 and the last 48 bytes correspond to the lower importance, suggesting that they are less relevant to the classification task under consideration.

Finally, comparing the behavior as the activity changes, we notice a significant difference in the importance of certain bytes for Skype and Teams. For Skype, the most notable difference is seen between the 32^{nd} and 96^{th} bytes, where these bytes are more important for ACall compared to Chat and VCall. A similar observation applies to Teams, where we also observe that for Chat the importance of bytes between 120^{th} and 152^{nd} and especially 528^{th} is significantly higher compared to ACall and VCall.

Takeaway: Bytes from 32^{nd} to 320^{th} of PAY input are the most crucial to address Joint-TC. A peak around 528^{th} byte is significant for all apps, especially Zoom and Teams. Since traffic of these apps is mostly composed of TLS biflows ($>60\%$ per-app), we deduce that the most important bytes include *Cipher Suite/Service Name* within the *Client Hello* and specific bytes within the *Server Certificate*. This underscores the potential inefficiency of this input type with the increasing adoption of an encrypted TLS header, such as TLS 1.3.

C. How do header fields affect Joint-TC?

Herein, we analyze the importance of header fields extracted from the first $N_p = 20$ packets of the biflow (viz. SEQ input) and feed the bSEQ.

Hence, Fig. 5 shows the median importance value for the 4 header fields across the first 20 packets of the SEQ input. The graph provides a breakdown of the activity of different apps, such as Discord, Skype, Slack, and Teams. [7]

[6] Other apps are omitted for brevity since Skype ≈ Messenger, GotoMeeting ≈ {Meet, Webex}, Zoom ≈ Slack, and Teams ≈ Discord.

[7] For brevity, we have omitted results for other apps, as they have similar results to those shown.

Overall, our analysis reveals that the first 10 packets are the most important for accurate prediction. Furthermore, PL not only consistently improves the prediction accuracy but also stands out as the most important field. This finding shows how the embedding layer can improve the information derived from the sequence of PL. Similarly, DIR also has a positive impact on the final decision in almost all cases, being the second most important feature.

Conversely, the importance of TCPWIN and IAT is generally lower, and in some cases, it may even adversely affect the accurate prediction. For instance, in the case of ACall for Skype and Teams, we observe that TCPWIN and IAT of some packets in the second half (i.e., from 11^{th} to 20^{th}) have negative values, which is not the case for other activities. This suggests that these fields could lead to incorrect model decisions. Similar findings are also noticeable for Chat and VCall on Slack and Discord.

Takeaway: *The first* 10 *packets of* SEQ *are the most important. Among the header fields,* PL *is the most crucial, followed by* DIR*. Conversely,* TCPWIN *and* IAT *are less important, and their impact can be negative on the final outcome.*

VI. Conclusions and Future Directions

In this work, we analyze the traffic of communication and collaboration apps (CC apps), experiencing rapid growth in the last few years. To effectively manage networks, DL finds extensive use in TC. Despite its promising outcomes, its opacity presents challenges in understanding the decision-making process, thereby hindering its broad adoption, echoing concerns raised by the European AI Act and AI HLEG.

In this paper, we use XAI techniques (i.e., Deep SHAP) to elucidate Mimetic-All, a multimodal architecture tailored for TC, in the Joint-TC task—viz. classify both app and user activities (i.e. *chat*, *video-*, and *audio-call*) jointly. Accordingly, we underscore the importance of Context Inputs, with median importance levels ranging from $\approx 12\%$ to $\approx 30\%$. More specifically, the number of exchanged packets and contextual biflows emerge as primary influencers for accurate predictions. Moreover, we assessed that bytes from the 32^{nd} to 320^{th} are critical for classification, with another significant peak observed near the 528^{th} byte. Additionally, we identified the first 10 packets, notably the payload length (PL), as crucial factors.

As *future directions*, we plan to (i) leverage SHAP explanations for feature selection, (ii) validate explanations using various feature attribution methods, and (iii) assess performance/importance through modality deletion. Although this application of XAI methodology requires *human-in-the-loop*, we aim to design automated dashboards to effectively exploit XAI explanations without the need for human involvement.

Acknowledgment

This work is partially supported by the “RESTART” Project and the “xInternet” Project within the PRIN 2022 program (D.D.104-02/02/2022), funded by MUR. This manuscript reflects only the authors’ views and opinions, and the Ministry cannot be considered responsible for them.

References

[1] Ericsson, “Ericsson mobility report,” https://www.ericsson.com/4ae12c/assets/local/reports-papers/mobility-report/documents/2023/ericsson-mobility-report-november-2023.pdf, 2023, accessed: 2024-05-06.

[2] Statista, “Number of mobile phone messaging app users worldwide from 2022 to 2030,” https://www.statista.com/statistics/1401734/mobile-messenger-user-worldwide/, 2024, accessed: 2024-05-06.

[3] Sandvine, “The Global Internet Phenomena Report 2023.” Jan 2023.

[4] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapè, “MIMETIC: mobile encrypted traffic classification using multimodal deep learning,” *Comp. Networks*, vol. 165, p. 106944, 2019.

[5] European Commission, “Ethics guidelines for trustworthy ai,” https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai, Jan 2024, online; accessed Mar. 2024.

[6] ——, “A european approach to artificial intelligence.” https://digital-strategy.ec.europa.eu/en/policies/european-approach-artificial-intelligence, Mar 2024, online; accessed Mar. 2024.

[7] K. Amarasinghe, K. Kenney, and M. Manic, “Toward Explainable Deep Neural Network Based Anomaly Detection,” in *IEEE HSI’18*, 2018, pp. 311–317.

[8] 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*, 2019, p. 29–36.

[9] A. Morichetta, P. Casas, and M. Mellia, “EXPLAIN-IT: Towards Explainable AI for Unsupervised Network Traffic Analysis,” in *ACM Big-DAMA’19*, 2019, p. 22–28.

[10] S. Rezaei, B. Kroencke, and X. Liu, “Large-Scale Mobile App Identification Using Deep Learning,” *IEEE Access*, vol. 8, pp. 348–362, 2020.

[11] C. Beliard, A. Finamore, and D. Rossi, “Opening the Deep Pandora Box: Explainable Traffic Classification,” in *IEEE INFOCOM WKSHPS*, 2020, pp. 1292–1293.

[12] 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.

[13] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, “DISTILLER: Encrypted traffic classification via multimodal multitask deep learning,” *Journ. of Netw. and Comp. App.*, vol. 183, p. 102985, 2021.

[14] G. Aceto, G. Bovenzi, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapè, “Characterization and Prediction of Mobile-App Traffic Using Markov Modeling,” *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 907–925, 2021.

[15] I. Akbari, M. A. Salahuddin, L. Ven, N. Limam, R. Boutaba, B. Mathieu, S. Moteau, and S. Tuffin, “A look behind the curtain: traffic classification in an increasingly encrypted web,” *ACM POMACS’21*, vol. 5, no. 1, pp. 1–26, 2021.

[16] A. Montieri, G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapè, “Packet-level prediction of mobile-app traffic using multitask Deep Learning,” *Comp. Networks*, vol. 200, p. 108529, 2021.

[17] A. Nascita, A. Montieri, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapé, “XAI meets mobile traffic classification: Understanding and improving multimodal deep learning architectures,” *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 4, pp. 4225–4246, 2021.

[18] A. M. Sadeghzadeh, S. Shiravi, and R. Jalili, “Adversarial Network Traffic: Towards Evaluating the Robustness of Deep-Learning-Based Network Traffic Classification,” *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 2, pp. 1962–1976, 2021.

[19] K. Fauvel, A. Finamore, L. Yang, F. Chen, and D. Rossi, “A Lightweight, Efficient and Explainable-by-Design Convolutional Neural Network for Internet Traffic Classification,” in *ACM SIGKDD’23*, 2023.

[20] I. Guarino, G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapè, “Contextual counters and multimodal deep learning for activity-level traffic classification of mobile communication apps during covid-19 pandemic,” *Comp. Networks*, vol. 219, p. 109452, 2022.

[21] I. Guarino, G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapè, “Explainable deep-learning approaches for packet-level traffic prediction of collaboration and communication mobile apps,” *IEEE Open Journ. of the Com. Soc.*, vol. 5, pp. 1299–1324, 2024.

[22] 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.