Classification of Communication and Collaboration Apps via Advanced Deep-Learning Approaches

Idio Guarino, Giuseppe Aceto, Domenico Ciuonzo, Antonio Montieri, Valerio Persico, Antonio Pescapé
University of Napoli "Federico II" (Italy)
{idio.guarino, giuseppe.aceto, domenico.ciuonzo, antonio.montieri, valerio.persico, pescape}@unina.it

***Abstract*—The lockdowns and lifestyle changes during the COVID-19 pandemic have caused a measurable impact on Internet traffic in terms of volumes and application mix, with a sudden increase of usage of communication and collaboration apps. In this work, we focus on five such apps, whose traffic we collect, reliably label at fine granularity (per-activity), and analyze from the viewpoint of traffic classification. To this aim, we employ state-of-art deep learning approaches to assess to which degree the apps, their different use cases (activities), and the pairs app-activity can be told apart from each other. We investigate the early behavior of the biflows composing the traffic and the effect of tuning the dimension of the input, via a sensitivity analysis. The experimental analysis highlights the figures of the different architectures, in terms of both traffic-classification performance and complexity w.r.t. different classification tasks, and the related trade-off. The outcome of this analysis is informative for a number of network management tasks, including monitoring, planning, resource provisioning, and (security) policy enforcement.**

***Index Terms*—communication apps; collaboration apps; COVID-19; deep learning; encrypted traffic; multimodal techniques; traffic classification.**

I. Introduction

The outbreak of the COVID-19 pandemic has induced governments worldwide to impose lockdown periods, that forced millions of citizens to stay at home, and also study/work from there if possible. As a consequence, Internet traffic from residential users has witnessed a significant growth ($+15-20\%$ in terms of volume) [1] as people engaged in remote work, education, commerce, and entertainment activities. The sudden change in the timing and mix of online presence has had a measurable impact on network performance in terms of increased variability of delay and loss rate [2]. This shift in both volume and nature of the Internet traffic poses a challenge to efficient network resource management, that in turn calls for enhanced network monitoring capabilities. More specifically, when observing network traffic the possibility to infer the application or the type of application that generated it (the process of *Traffic Classification*, TC in the following) becomes paramount to most management and planning actions. While TC has been a hard problem and an active field of research for decades, its application is further challenged by specific characteristics of communication and collaboration apps: consistent use of encryption; common usage of application-level protocols as transport sublayers (namely, TLS and HTTP); different functioning modes (*activities*) for a single application; execution from mobile devices (platforms characterized by frequent and automated software updates). These challenges are now pushing toward the adoption of advanced *Deep Learning* (DL) approaches, able to cope with frequently changing input nature, and offering promising performance when dealing with complex and hard-to-model problems [3]. Thus, on the one hand, a better understanding is required of the traffic of applications that have seen a surge in utilization after the COVID-19 pandemic. On the other hand, an assessment of modern TC approaches is needed, applied to this specific scenario.

To respond to these needs, with this work we target five communication and collaboration apps (`GotoMeeting`, `Skype`, `Teams`, `Webex`, and `Zoom`), that have seen dramatic increase of usage in correspondence to lockdowns. We analyze their traffic and assess the performance of the state-of-art of TC approaches for classifying the specific app, the kind of activity (*Webinar*, *Video-call*, *Video-conference*) or the combination $app \times activity$ (finest grain). More in details, the contributions of this work are summarized in the following. *(i)* We collect a dataset of mobile-app traffic traces, that is human-generated, recent, and reliably labeled with both the *application* that generated the traffic and the *activity* that was performed by the user. *(ii)* For each activity, we analyze the traffic in terms of (payload-carrying) packet direction, payload length, payload content, TCP window size, and inter-arrival time, for the initial part of the biflow (*early behavior* analysis). *(iii)* We apply state-of-art DL architectures (an 1D-CNN, a hybrid 2D-CNN + LSTM, and the multimodal MIMETIC [4]) to the dataset, to classify the traffic at app, $activity$, and $app \times activity$ granularity. *(iv)* We compare and discuss the results of the different approaches, deriving conclusions on the nature of the traffic of communication and collaboration apps, on the performance of the considered approaches, and future avenues of research.

II. Related Work

A. Impact of COVID-19 on the Nature of Internet Traffic

Following the global spread of the COVID-19 pandemic, several works have analyzed its impact on the Internet, focusing on the nature of the traffic and on network performance.

This work was carried out by Dr. Idio Guarino under the grant "O. Carlini" funded by GARR, the Italian national network of University and Research.

Feldmann et al. [1] inspect network flow data from multiple vantage points and analyze the effect of the lockdown on traffic shifts: an increase in European Internet traffic is found, mainly associated to VPN and videoconferencing applications ($+200\%$ in volume). In addition, Lutu et al. [5] analyze the changes in mobility (-50%) and their impact on the cellular network traffic, finding notable variations of voice and data traffic volumes, and packet loss. Additional degradations of network services are highlighted by Böttger et al. [6], who investigate Internet traffic growth (as seen from Facebook edge network) at the beginning of the pandemic, and observe a correlation between the phase of traffic growth and the spread of COVID-19. Moreover, the study by Candela et al. [2] assesses the impact on Internet latency due to the changes in users' behavior during COVID-19 restrictions. Favale et al. [7] analyze the effect of lockdown measures on an Italian campus network, showing how the increased use of collaboration platforms, VPNs, and remote desktop services has pushed to unprecedented peaks of $1.5\,\mathrm{Gbit/s}$, with few cases of poor experienced performance. Finally, Affinito et al. [8] consider websites and domains used during the enforcement of the social distancing measures, showing that `Youtube`, `Netflix`, `Facebook`, `Whatsapp`, `Skype`, and `Zoom` result among the most used applications.

In line with these studies, the object of our work is the analysis of the network traffic generated by the most popular communication and collaboration mobile apps, whose pandemic-driven surge deeply shapes the nature of Internet traffic and potentially relates to performance issues.

B. DL-based Traffic Classification

Recently, several works have faced TC via DL approaches. Wang [9] has first used a Stacked AutoEncoder (SAE) for unencrypted traffic identification, achieving superior performance w.r.t. standard neural networks ($\geq 90\%$ precision and recall). **Encrypted TC** is targeted by Wang et al. [10], who propose a method based on 1D Convolutional Neural Network (1D-CNN)—outperforming the 2D variant—to tackle different TC tasks related to encrypted and VPN-tunneled applications and traffic classes. Similar tasks are tackled by Lotfollahi et al. [11] proposing *Deep Packet* (based on 1D-CNN and SAE), able to outperform Machine Learning (ML)-based classifiers for encrypted TC at packet granularity. Recurrent Neural Networks have been considered by Lopez-Martin et al. [12], proposing different hybrid DL architectures that combine Long Short-Term Memory (LSTM) and 2D-convolutional layers.

Focusing on the classification of **mobile-app traffic**, Rezaei et al. [13] leverage a CNN fed with the header and the payload of the first six packets of a biflow. Similarly, Liu et al. [14] devise *FS-Net*, an encoder-decoder architecture based on Bidirectional Gated Recurrent Units (BiGRU) taking as input IP-packet sizes of flow sequences. In this context, in our previous work [3], we define a systematic framework to dissect the encrypted mobile TC using DL, and compare a number of the aforementioned techniques for a comprehensive evaluation. Common usage of biased inputs (e.g., local-network metadata [10], or source and destination ports [12]) inflating TC performance is also discussed and discouraged.

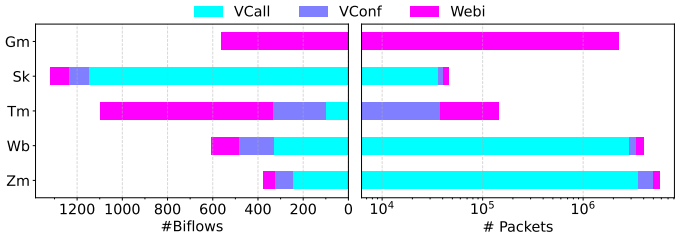


Figure 1: Communication and collaboration apps considered. Performed activities, number of biflows (*left*-bar) and number of packets (*right*-bar) are reported for each app. Note that the log scale is used to report the packets' number.

Multimodal DL solutions have been recently proposed to face mobile-app TC. We propose MIMETIC [4], a general framework for capitalizing the heterogeneous views associated with a traffic object, along with a novel training procedure based on pre-training and fine-tuning. Experimental results show that MIMETIC classifier outperforms single-modal, ML-based, as well as late-combination of traffic classifiers both in terms of TC performance and training complexity. Following along the same research direction, Wang et al. [15] propose *App-Net*, consisting of two modalities: a (bidirectional) LSTM and a 1D-CNN. Experimental results show that App-Net outperforms ML-based and single-modal DL-based traffic classifiers, while performing almost on par w.r.t. MIMETIC.

Capitalizing on the latest advancements on TC solutions via DL, we investigate the performance of a state-of-art multimodal architecture and compare its performance against some other recent but simpler proposals.

III. EXPERIMENTAL SETUP

A. Dataset Collection and Ground Truth Generation

The dataset was collected by students and researchers within April–June 2021 leveraging the MIRAGE architecture [16] (conveniently optimized to capture traffic of communication and collaboration apps) in the ARCLAB laboratory at the University of Napoli "Federico II".[1] Experimenters used three mobile devices: a Google Nexus 6 (Android 10) and two Samsung Galaxy A5 (Android 6.0.1). In each capture session—whose duration spanned from 15 to 80 minutes based on the activity—the experimenters performed a specific activity, so as to obtain a traffic dataset that reflects the common usage of considered apps.[2] Each session resulted in a PCAP traffic trace and additional system log-files with ground-truth information. Based on the latter, each biflow[3] was reliably

[1]We highlight that the captures were carried out by adhering to the distancing/mask-wearing rules prescribed by regional/national decrees in force at the moment of the collection.

[2]Each traffic capture session has been performed with the up-to-date version of the app. Also, to limit background traffic, network access has been disabled for all the apps but the one under test.

[3]A bidirectional flow (biflow) encompasses 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 [3].

labeled with the corresponding *Android package-name* by considering established network-connections (via `netstat`). This information was further enriched with a label referring to the specific activity performed by the user operating the device (see Sec. III-B).

B. Apps' and Activities' Selection Rationale

Communication and collaboration apps—used for business meeting, classes, and social interaction—have experienced a huge utilization increment when "stay-at-home" orders were issued worldwide. Based on both popularity and utilization boost, herein we focus on *five* of them: `GotoMeeting` (*Gm*), `Skype` (*Sk*), `Teams` (*Tm*), `Webex` (*Wb*), and `Zoom` (*Zm*). Indeed, `Zoom` has obtained the steepest increment with its traffic scaling to orders of magnitude, followed by `Webex`, `GotoMeeting`, `Teams`, `BlueJeans` (whose traffic we are currently collecting), and `Skype` [17]. Also, during 15th–21st March 2020, `Zoom` was downloaded $14\times$, $20\times$, and $55\times$ more than the weekly average during Q4 2019 in the US, UK, and Italy, respectively [18]. Similarly, `Teams` also experienced significant growth in Italy (resp. France) with $30\times$ (resp. $16\times$) more downloads. The considered apps have been extensively exploited for remote (and blended) teaching in Italian[4] and European[5] institutions and universities.

Specifically, according to the observed app usage, the experimentation covered the following activities (all related to live events): *Webinar (Webi)*—involves many attendees and one presenter transmitting his/her own audio together with slides and/or his/her own video (e.g., seminar or online lesson); *Video-call (VCall)*—involves just two participants transmitting both audio and video; *Video-conference (VConf)*—involves more than two participants broadcasting audio/video.

Figure 1 summarizes the mobile apps used in this study, highlighting also the activities carried out with each, and the amount of traffic collected in terms of biflows and packets.

C. Traffic Classification Methodology

In the following, we consider state-of-the-art classifiers selected among the best single-modal and multimodal alternatives—based on extensive performance evaluation carried out in our previous works [3, 4]—in terms of both DL architecture and *unbiased* input data. According to the results of previous work, we leverage the biflow as the relevant traffic object of our TC tasks.

Going into details, we consider an 1D-CNN fed with the first N_b bytes of transport-layer payload (`PAY`) of each biflow, and having an architecture analogous to that proposed in [10] in terms of both elementary layers and related hyperparameters. Moreover, we evaluate a hybrid composition of *2D-CNN + LSTM* (named HYBRID hereinafter), having as input: (i) the number of bytes in transport-layer payload (`PL`), (ii) the TCP window size (`TCPWIN`, set to zero for UDP packets), (iii) the inter-arrival time (`IAT`), and (iv) the packet direction (`DIR`) $\in \{-1, 1\}$ of the first N_p packets of each biflow, as proposed in [12]. Finally, we also consider the multimodal MIMETIC classifier we proposed [4]. MIMETIC consists of two modalities fed each with one of the input types used for the other classifiers, and is trained via a two-phase procedure consisting of pre-training of individual modalities and fine-tuning of the whole architecture.

To foster a fair comparison, all the classifiers are trained (via a 10-fold cross-validation procedure) for a total of 90 epochs (for MIMETIC, we consider 25 epochs for pre-training of each modality and 40 epochs for fine-tuning) for minimizing categorical cross-entropy loss, and exploit the Adam optimizer (with a batch size of 50) and the early-stopping technique to prevent overfitting.

In this study, we take advantage of the ground-truth information associated to each biflow (reporting both the app generating the traffic and the activity performed) to *instruct and evaluate different supervised strategies* corresponding to *three* TC tasks: *(i)* classifying the app (App-TC), *(ii)* classifying the activity (Act-TC), and *(iii)* classifying both the app and the activity (Joint-TC). Accordingly, we *train* the above models with: *(a)* app-related ground truth only (APP), *(b)* activity-related ground truth only (ACT), and *(c)* the joint app-and-activity ground truth (APP×ACT). Note that APP and ACT produce classifiers able to address only the specific task they are trained for (i.e. classifying either apps or activities), whereas when training the DL architectures based on APP×ACT, *all* three TC tasks above-described can be addressed.

IV. EXPERIMENTAL ANALYSIS

A. Biflow-level Characterization of Early Behavior

Based on the input fed to considered DL architectures, we first focus on the possible structure within the information exchanged across the biflows in the initial part of the communication. To this end, we report the sequence of `PL`/`DIR`/`IAT`/`TCPWIN` of the first 36 app-level packets[6] and the first 2048 payload bytes. Specifically, for a given app, we report the average value on all biflows for each packet/byte index. For brevity, the analysis depicted in Fig. 2 focuses on `Skype` and `Webex`.[7] For both the apps, the above information is broken down into the considered activities (*VCall*, *VConf* and *Webi*) and also reported in summary form (*All*).

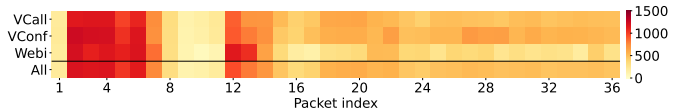
Referring to *All*, it is apparent that there is an appreciable difference between the behavior of the first and the second half of the sequence (≈ 16 packets) in terms of `PL`/`DIR`/`IAT` (the behavior is slightly less evident for `TCPWIN`), which highlights the very initial part of the biflows. Indeed, during the second half of the 36 packets, the trend is less structured and also very similar among different activities. Conversely, looking at the first 16 packets, the corresponding trend *depends on the specific app*. For instance, for `Skype` there is a different `PL`/`DIR`/`IAT`/`TCPWIN` signature for *VCall*, whereas *Webi* and

[4]Fondazione CRUI – COVID-19 | Strumenti per la didattica digitale.

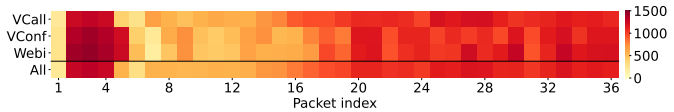
[5]European University Institute – Software available at the EUI.

[6]Packets with no payload are discarded since they reflect transport-layer signaling neither depending on the nature of the app nor the performed activity.

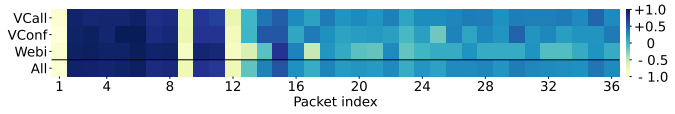
[7]A similar behavior as `Webex` was observed for `GotoMeeting`, `Teams`, and `Zoom`.



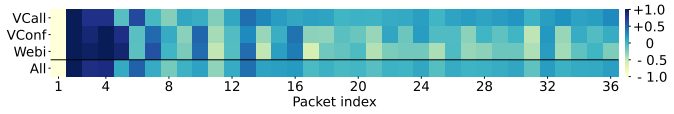
(a) Average PL [B] of the first 36 packets for Webex.



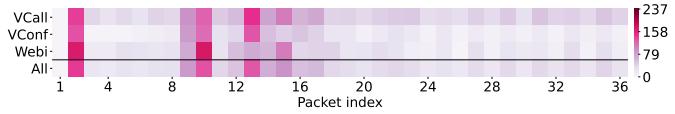
(b) Average PL [B] of the first 36 packets for Skype.



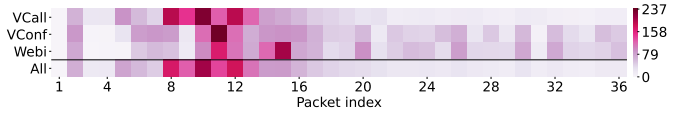
(c) Average DIR of the first 36 packets for Webex.



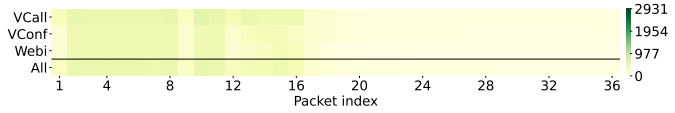
(d) Average DIR of the first 36 packets for Skype.



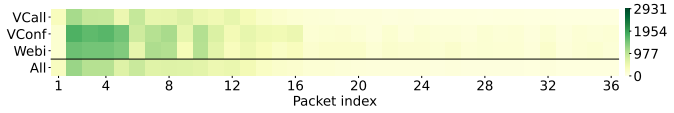
(e) Average IAT [ms] of the first 36 packets for Webex.



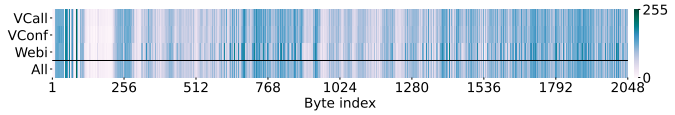
(f) Average IAT [ms] of the first 36 packets for Skype.



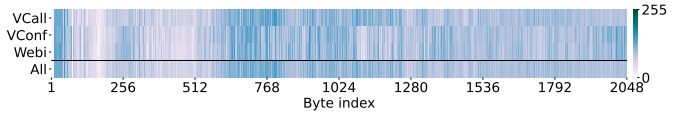
(g) Average TCPWIN of the first 36 packets for Webex.



(h) Average TCPWIN of the first 36 packets for Skype.



(i) Average PAY of the first 2048 Bytes for Webex.



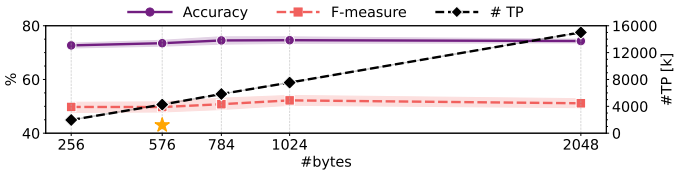
(j) Average PAY of the first 2048 Bytes for Skype.

Figure 2: Properties of biflows' time-series with respect to PL, DIR, IAT, TCPWIN, and PAY for Webex (a, c, e, g, i) and Skype (b, d, f, h, j) based on the *activity*-type and in summary (*All*) form. The downstream and upstream DIR is mapped on $+1$ and -1, respectively.

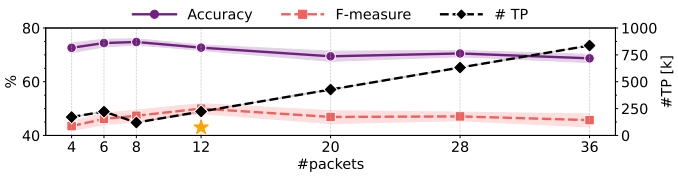
VConf appear more indistinguishable. Such signature seems correlated for PL, DIR, and TCPWIN and related to the first 5 packets, whereas for IAT the peculiarity of *VCall* can be observed in the index interval 8–16. On the contrary, for Webex, there is a quite-similar pattern among the different activities for all fields associated to the 36 packets. Finally, referring to PAY, a similar observation applies to the identifiability of *VCall* activity within Skype, whereas for Webex a slightly distinctive behavior of *Webi* is apparent.

B. Sensitivity Analysis

Herein we perform a sensitivity analysis aimed at tuning the dimension of the two types of input (cf. Sec. III-C) in terms of number of bytes N_b and number of packets N_p employed. Figure 3 depicts the Accuracy and F-measure[8] attained by the 1D-CNN and HYBRID architectures, respectively when varying $N_b \in [256, 2048]$ B and $N_p \in [4, 36]$ packets.[9] Also, to highlight the input size-complexity trade-off, we also report



(a) 1D-CNN.



(b) HYBRID.

Figure 3: Accuracy [%], F-measure [%], and number of trainable parameters [k] of 1D-CNN (a) and HYBRID (b) when varying the input dimensions N_b and N_p, respectively. Results refer to the Joint-TC task. The best trade-off value is highlighted via a $\star$ marker.

the number of trainable parameters vs. the size of considered input data. Results refer to the (hardest) Joint-TC task.

Looking at Fig. 3a, we can notice that even if the best Accuracy and F-measure are attained with $N_b = 1024$, when considering $N_b > 576$ B, the performance is almost

[8]Accuracy is the share of correctly-classified samples, while F-measure is the harmonic mean of precision (the proportion of classifier decisions for a given class which are actually correct) and recall (the per-class accuracy).

[9]We did not perform the same analysis also for MIMETIC due to the combinatorial complexity resulting from considering all the possible combinations of N_b and N_p, and the time needed to train/test each resulting classifier.

insensitive to the higher input dimensionality: when passing from $N_b = 576$ to $N_b = 1024$, the Accuracy (resp. F-measure) raises by only $+1\%$ (resp. $+2\%$). On the other hand, the same variation of N_b causes a steep increment ($4\times$) in the number of trainable parameters, thus obtaining a much more complex architecture with almost the same TC performance.

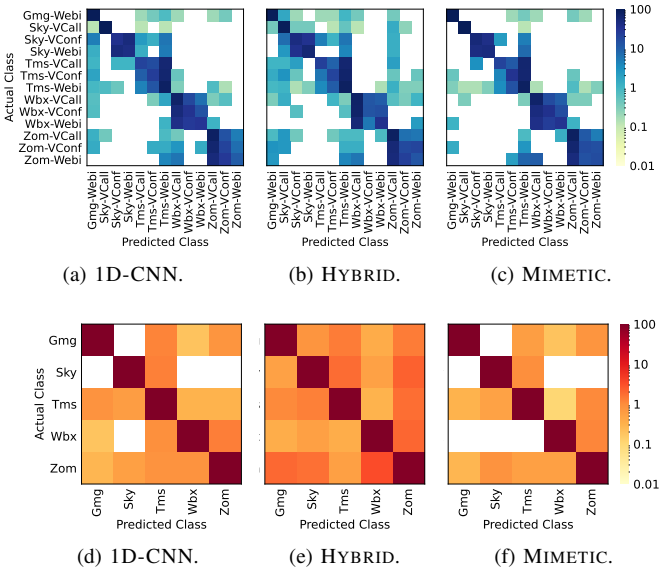
Regarding Fig. 3b, a more evident trend can be observed. Indeed, by using $N_p = 8$ or $N_p = 12$, HYBRID attains the best performance in terms of Accuracy or F-measure, respectively, keeping also a limited (viz. manageable) number of trainable parameters. Regarding the latter, a notable exception is given by $N_p \in [4, 6]$ corresponding to a number of trainable parameters comparable with $N_p = 12$. The reason is that, with such small inputs, we need to resort to a different padding—implying additional complexity—to implement the HYBRID architecture. In light of the above considerations, in the next analyses—considering the F-measure as the target performance measure, and to limit the complexity of obtained classifiers—we employ $N_b = 576$ B and $N_p = 12$ packets, if not explicitly stated otherwise.

C. App and Activity Classification

Table I reports the performance of the three considered architectures (i.e. 1D-CNN, HYBRID, and MIMETIC) in terms of both Accuracy and F-measure. In detail, according to the definition of the TC tasks we aim to tackle, the table reports the performance for Joint-TC, App-TC, and Act-TC (in different columns). Also, in line with the discussion in Sec. III-C, we evaluate these attained performance figures when adopting different training strategies: APP×ACT, APP, and ACT (in "Training Strategy" column). Overall, it is evident that Joint-TC is the hardest task to tackle by all the architectures, with F-measure exposing lower values (in the range 50%–53%). On the other side, for App-TC, all the classifiers achieve remarkably higher F-measure values (95%–98%). Finally, Act-TC performance sits in the middle of the other two (63%–68% F-measure). While Joint-TC is expected to be a harder task in nature (13 classes), Act-TC results in such a lower performance figure in spite of a simpler problem (3 classes). This result witnesses that different activities are hardly distinguishable, confirming the outcome of the early-behavior characterization in Sec. IV-A.

Considering all TC tasks, MIMETIC *always returns better performance than the two competing approaches in terms of both Accuracy and F-measure*. Specifically, for Joint-TC, MIMETIC achieves $+3\%$ F-measure w.r.t. both 1D-CNN and HYBRID. Similarly, when focusing on App-TC, MIMETIC shows $+1\%$ and $+3\%$ F-measure w.r.t. 1D-CNN and HYBRID, respectively, while on Act-TC, MIMETIC performs even better than both 1D-CNN ($+2\%$ F-measure) and HYBRID ($+5\%$ F-measure).

Interestingly, Tab. I also highlights that the training strategy adopted may impact the performance achieved by the models. Indeed, focusing on App-TC all the architectures achieve better performance when relying on APP training strategy. On the other hand, looking at Act-TC, APP×ACT training strategy



(a) 1D-CNN. (b) HYBRID. (c) MIMETIC.

(d) 1D-CNN. (e) HYBRID. (f) MIMETIC.

Figure 4: Confusion matrices of 1D-CNN (a, d), HYBRID (b, e), and MIMETIC (c, f) considering the best training training strategy in terms of F-measure. Results are reported on Joint-TC (a-c) and App-TC (d-f). Note that the log-scale is used to evidence small errors.

results in better performance for HYBRID and MIMETIC, whereas ACT is the best training strategy for 1D-CNN.

To detail the performance at a finer grain, Fig. 4 reports the *confusion matrices associated to the best training strategy in terms of F-measure* achieved by each architecture for both Joint-TC and App-TC tasks. Referring to the hardest Joint-TC task (Figs. 4a, 4b, and 4c), the performance figures reported in Tab I are confirmed by the fact that MIMETIC can also substantially reduce the misclassification patterns w.r.t. 1D-CNN and HYBRID, confining the errors within the activities of the same app. Similarly, for the App-TC task (Figs. 4d, 4e, and 4f), the errors of MIMETIC are less severe than those of 1D-CNN and HYBRID, confirming its specific suitability for such a task. The difficulty in discriminating among the activities is highlighted by the confusion matrices of the Act-TC task (not shown for brevity) which show analogous misclassifications for all the approaches, corroborating the traffic similarity of the considered activities regardless of the specific app employed.

Finally, if we further enlarge the set of aspects of interest in our investigation also taking into account the *complexity of the considered architectures*, the analysis provides other interesting pieces of evidence (last column of Tab. I). In fact, the complexity highly varies with considered architecture: 1D-CNN and MIMETIC are $\approx 20\times$ and $\approx 4\times$ more complex than HYBRID, respectively, in terms of trainable parameters (which are roughly proportional to both the training time and the memory occupation). *Hence,* MIMETIC *provides the best trade-off between TC performance and complexity.*

V. CONCLUSIONS AND FUTURE DIRECTIONS

The COVID-19 pandemic has caused a sudden—and possibly non-temporary—surge of the usage of communication and

Table I: Accuracy, F-measure, and number of Trainable Parameters (#TP) comparison of the three architectures (1D-CNN, HYBRID, and MIMETIC) when trained on different class-labels (i.e. related to APP×ACT, APP, and ACT) for different classification tasks. #TP depends on the classification task, but for each classifier the variations are smaller than shown precision. Results are in the format $avg.\ (\pm std.)$ obtained over 10-folds. The best result per metric (column) is highlighted in boldface.

Classifier	Training Strategy	Joint-TC		App-TC		Act-TC		#TP [k]
		Accuracy [%]	F-measure [%]	Accuracy [%]	F-measure [%]	Accuracy [%]	F-measure [%]	
1D-CNN	APP×ACT	73.50(±1.95)	49.74(±3.41)	97.17(±1.08)	96.98(±1.25)	74.22(±1.81)	64.62(±2.36)	4261
	APP	-	-	98.04(±0.94)	97.89(±1.16)	-	-	4253
	ACT	-	-	-	-	74.83(±2.21)	65.89(±2.72)	4251
HYBRID	APP×ACT	72.69(±1.86)	49.99(±3.23)	95.45(±1.41)	95.08(±1.46)	74.38(±1.47)	63.38(±2.47)	222
	APP	-	-	95.33(±1.96)	94.71(±2.25)	-	-	222
	ACT	-	-	-	-	73.50(±2.27)	62.50(±2.89)	**221**
MIMETIC	APP×ACT	**75.49(±1.84)**	**52.71(±3.75)**	98.07(±0.90)	97.86(±1.16)	**76.12(±1.69)**	**67.48(±2.12)**	942
	APP	-	-	**98.49(±0.60)**	**98.30(±0.73)**	-	-	937
	ACT	-	-	-	-	75.58(±2.82)	66.87(±3.38)	936

collaboration apps, which has impacted the nature of Internet traffic, calling for novel improved tools for network monitoring and management.

In this work, we focused on the TC of the most popular communication and collaboration apps via DL approaches. We considered three TC tasks (Joint-TC, App-TC, and Act-TC) and different training strategies based on the ground truth. MIMETIC (a state-of-art multimodal TC approach) has been compared against recent DL single-modal solutions (an 1D-CNN and a hybrid 2D-CNN + LSTM). The experimental results—based on a newly collected dataset covering five Android mobile apps (`GotoMeeting`, `Skype`, `Teams`, `Webex`, and `Zoom`) and three user activities (*Webi*, *VCall*, and *VConf*)—include the characterization of the early behavior of biflows generated by specific activities and apps, the tuning of the considered architectures with respect to the dimensionality of the input parameters, as well as the resulting complexity.

While all the considered architectures achieve good performance (95%–98% F-measure) when tackling App-TC, Joint-TC represents the hardest task (50%–53% F-measure). Despite the simpler TC problem (3 classes), the low performance of Act-TC (63%–68% F-measure) highlights that user activities are hardly distinguishable by the considered architectures via the adopted input configurations.

Comparing the considered architectures, MIMETIC is the best performing one for all TC tasks in terms of both Accuracy and F-measure. In addition, when considering the complexity aspects, MIMETIC proves to be even the best choice, exposing a complexity $\approx 4\times$ lower than 1D-CNN. Future directions of research will account for (i) use of advanced learning strategies encompassing multitask and hierarchical traffic classifiers; (ii) use of advanced DL layers (e.g., inception, residual, attention); (iii) further traffic analysis tasks as fine-grain modeling and prediction.

REFERENCES

[1] A. Feldmann, O. Gasser, F. Lichtblau, E. Pujol, I. Poese, C. Dietzel, D. Wagner, M. Wichtlhuber, J. Tapiador, N. Vallina-Rodriguez, O. Hohlfeld, and G. Smaragdakis, "The lockdown effect: Implications of the COVID-19 pandemic on Internet traffic," in *ACM Internet Measurement Conference (IMC)*, 2020, p. 1–18.

[2] M. Candela, V. Luconi, and A. Vecchio, "Impact of the COVID-19 pandemic on the Internet latency: A large-scale study," *Computer Networks*, vol. 182, p. 107495, 2020.

[3] 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.

[4] 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.

[5] A. Lutu, D. Perino, M. Bagnulo, E. Frias-Martinez, and J. Khangosstar, "A characterization of the COVID-19 pandemic impact on a mobile network operator traffic," in *ACM Internet Measurement Conference (IMC)*, 2020, p. 19–33.

[6] T. Böttger, G. Ibrahim, and B. Vallis, "How the Internet reacted to COVID-19: A perspective from Facebook's edge network," in *ACM Internet Measurement Conference (IMC)*, 2020, p. 34–41.

[7] T. Favale, F. Soro, M. Trevisan, I. Drago, and M. Mellia, "Campus traffic and e-Learning during COVID-19 pandemic," *Computer Networks*, vol. 176, p. 107290, 2020.

[8] A. Affinito, A. Botta, and G. Ventre, "The impact of COVID on network utilization: an analysis on domain popularity," in *IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2020, pp. 1–6.

[9] Z. Wang, "The Applications of Deep Learning on Traffic Identification." Black Hat USA, Las Vegas, 2015.

[10] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in *IEEE ISI'17*, 2017.

[11] M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade, and M. Saberian, "Deep packet: A novel approach for encrypted traffic classification using deep learning," *Soft Computing*, vol. 24, no. 3, pp. 1999–2012, 2020.

[12] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for Internet of Things," *IEEE Access*, vol. 5, pp. 18 042–18 050, 2017.

[13] S. Rezaei, B. Kroencke, and X. Liu, "Large-scale mobile app identification using deep learning," *IEEE Access*, vol. 8, pp. 348–362, 2020.

[14] C. Liu, L. He, G. Xiong, Z. Cao, and Z. Li, "FS-Net: A flow sequence network for encrypted traffic classification," in *IEEE Conference on Computer Communications (INFOCOM)*, 2019, pp. 1171–1179.

[15] X. Wang, S. Chen, and J. Su, "Automatic mobile app identification from encrypted traffic with hybrid neural networks," *IEEE Access*, vol. 8, pp. 182 065–182 077, 2020.

[16] G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapé, "MIRAGE: Mobile-app traffic capture and ground-truth creation," in *4th International Conference on Computing, Communications and Security (ICCCS)*, 2019, pp. 1–8.

[17] Sandvine, "The Global Internet Phenomena Report COVID-19 Spotlight." May 2020.

[18] Lexi Sydow - App Annie, "Video Conferencing Apps Surge from Coronavirus Impact." March 2020.