Neurocomputing 409 (2020) 306–315

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Toward effective mobile encrypted traffic classification through deep learning

Giuseppe Aceto [a,b], Domenico Ciuonzo [a], Antonio Montieri [a], Antonio Pescapé [a,b,*]

[a] University of Napoli "Federico II", Italy
[b] Network Measurement and Monitoring (NM2) s.r.l., Italy

ARTICLE INFO

Article history:
Received 3 March 2020
Revised 20 April 2020
Accepted 7 May 2020
Available online 26 May 2020
Communicated by Lei Zou

Keywords:
Android apps
Deep learning
Encrypted traffic
iOS apps
Machine learning
Mobile apps
Privacy
Traffic classification

ABSTRACT

Traffic Classification (TC), consisting in how to infer applications generating network traffic, is currently the enabler for valuable profiling information, other than being the workhorse for service differentiation/blocking. Further, TC is fostered by the blooming of mobile (mostly encrypted) traffic volumes, fueled by the huge adoption of hand-held devices. While researchers and network operators still rely on machine learning to pursue accurate inference, we envision Deep Learning (DL) paradigm as the stepping stone toward the design of practical (and effective) mobile traffic classifiers based on automatically-extracted features, able to operate with encrypted traffic, and reflecting complex traffic patterns. In this context, the paper contribution is fourfold. First, it provides a taxonomy of the key network traffic analysis subjects where DL is foreseen as attractive. Secondly, it delves into the non-trivial adoption of DL to mobile TC, surfacing potential gains. Thirdly, to capitalize such gains, it proposes and validates a general framework for DL-based encrypted TC. Two concrete instances originating from our framework are then experimentally evaluated on three mobile datasets of human users' activity. Lastly, our framework is leveraged to point to future research perspectives.

1. Introduction

In last years network operators have experienced tremendous growth of network traffic, mostly generated by mobile devices [1]. To face this unique challenge, sophisticated network monitoring systems, incorporating intelligence through *machine learning (ML)*, are employed by several network players [2]. Yet, their success resorts to the design of handcrafted features, thanks to domain experts. Such process is impractical when facing the fast-paced mobile traffic evolution, because it can be neither automated nor crowdsourced to non-experts (due to the high specialization required). After a large number of ML-based approaches [3–6], recently *deep learning (DL)* [7,8], a cutting-edge subset of ML techniques, has emerged as the disruptive breakthrough toward the automatic design of accurate inference systems able to capture complex dependencies among data, thus limiting human expert intervention.

A pillar for network monitoring services is represented by *traffic classification (TC)* [9], namely how to infer the application generating the traffic. Indeed, TC represents a key prerequisite for security and QoS enforcement, and additional appeal is arising for *mobile TC* [10–13] due to its potential for valuable profiling information (e.g. to advertisers and security agencies), while also implying privacy downsides (e.g. recognition of health or dating apps, or in bring-your-own-device scenarios). Concurrently, the broad adoption of encrypted protocols (TLS) and dynamic ports blocks the road to accurate TC, defeating traditional deep packet inspection and port-based techniques [9,14]. This paves the way to DL techniques, here envisioned as the stepping stone toward the fulfillment of high performance in the challenging *encrypted traffic* [11,15] contexts, allowing to train classifiers directly from input data by automatically distilling structured and complex feature representations [7,16]. Still, DL adoption in network TC is thorny, and currently less understood [13]. More important, other than the encrypted-traffic issue, mobile TC is marked by a high number of apps, possibly generating similar traffic patterns and with complex fingerprints. The latter is due to scarce number of training samples per app and device/OS/version diversity. Hence, such challenging and dynamic scenario justifies DL higher complexity and training requirements.

* Corresponding author.
E-mail addresses: giuseppe.aceto@unina.it (G. Aceto), domenico.ciuonzo@unina.it (D. Ciuonzo), antonio.montieri@unina.it (A. Montieri), pescape@unina.it (A. Pescapé).

https://doi.org/10.1016/j.neucom.2020.05.036
0925-2312/

1.1. Summary of the contributions and paper organization

In view of the discussed considerations, the contributions of this work are manifold:

- We give an overview of the *key network traffic analysis subjects* where DL is foreseen as attractive, since their common intent is to capitalize network-level raw data automatically to extract valuable info.
- We categorize the *state-of-the-art in DL-based TC* toward its effective application in *mobile and encrypted context*, providing also a systematic taxonomy, of the most-related literature.
- To pinpoint and overcome the limitations of literature, we propose a *general framework for DL-based mobile and encrypted TC*, based on a rigorous definition of its milestones: (i) the choice of the traffic object, (ii) the definition of the input(s), (iii) the simultaneous TC tasks required, and (iv) the corresponding DL architecture. Thanks to the above framework, clear guidelines are provided to designers for the judicious choice of relevant segmentation criteria and unbiased (while effective) input(s) in DL-based TC [13,17]. More importantly, our proposal overcomes the design limitations of current works (limited to either single-modality or single-task learning, e.g. [17–20]), by envisioning the joint use of multi-modal and multi-task techniques via the "connectionist" approach granted by DL.
- We validate *two actual implementations of the proposed framework* on three recent human-generated mobile traffic datasets. One instance coincides with the best DL-based baseline on mobile encrypted TC [13], while the other is a *novel architecture*, drawn from our proposal, we devise herein to exploit multiple inputs. We show that the latter instance surpasses the former, accurately predicts the app generating the traffic, and beats the state-of-the-art in ML-based mobile TC [11].
- Finally, our framework allows us to *surface future perspectives* toward an effective mobile and encrypted TC by means of advanced DL techniques.

The rest of the paper is organized as follows: Section 2 presents a review of the recent success achieved by DL in network traffic analysis; Section 3 provides a categorization of literature background on TC through DL; the proposed general framework for DL-based mobile and encrypted TC is described in Section 4, with Section 5 reporting the experimental validation of its two proposed implementations; finally, Section 6 suggests insights and possible future directions.

To foster manuscript readability, Table 1 summarizes the acronyms used in the main text. Conversely, we report those used only in tables within the corresponding captions.

Table 1
Summary of the acronyms used in the manuscript.

Acronym	Definition
CNN	Convolutional Neural Network
CR	Classified Ratio
DL	Deep Learning
ECE	Expected Calibration Error
FB	FaceBook
FBM	FaceBook Messenger
KPI	Key Performance Indicator
ML	Machine Learning
MM	Multi Modal
MT	Multi Task
RTPE	Run Time Per-Epoch
SM	Single Modal
ST	Single Task
TC	Traffic Classification
TI	Traffic Identification

2. Deep learning in network traffic analysis

Telecom operators and ISPs have a long history of traffic-data analysis operations, possess a huge availability of network-level data, and have thus enjoyed decades-long research and applications on the topic. The huge success of DL in several fields is recently igniting global interests in exploiting it also in networking, where its adoption can leverage this solid know-how and help facing new challenges of *mobile* network-level data analysis.

To this end, in this section we review the recent success achieved by DL in network traffic analysis, discussing the key *subjects* which have found beneficial impact (and can benefit further) from its adoption, as summarized in Table 2. For each subject, we highlight the related privacy (*P*), security (*S*), and network management (*M*) *concerns* (possibly even partially affected by the considered subject), along with the *inference task* associated (i.e. *time-series prediction* or *binary/multiclass classification* task). Moreover, we list a few exemplifying *papers* showing the successful adoption of DL, along with the *DL family* proposed as the design solution. We exclude therein *traffic identification* and *classification*, whose detailed analysis is provided in later Table 3.

We remark that this taxonomy is not strictly tight, since some degree of overlapping could be possible between certain works on related subjects. A description of these subjects is given hereinafter. For example, studies tackling malware classification usually also perform malware detection, as a preliminary step of their analysis. Moreover, malware and (normal) traffic classification have been also investigated together, as in W. Wang et al. [29] and H. Huang et al. [28] (see Table 3), both as separate problems or in a multi-task fashion, respectively.

Network Prediction. It refers to forecasting network traffic or performance indicators given historical measurements or related data. Specifically for mobile networks, given the high variability of both traffic and network conditions, and the stringent QoS requirements of new applications, this constitutes a challenging subject. Hence, the design of algorithmic solutions with increased traffic prediction abilities directly reflects on improved network management.

Anomaly Detection and Attack Classification. The aim is to reveal anomalies in the traffic due to attacks (*anomaly detection*) based on patterns drawn from normal network behavior, and, possibly, to infer also the specific attack experienced (*attack classification*). Accordingly, both these subjects are directly linked to the security aspect, whereas attack classification allows a finer network management, for example attack-specific network countermeasures.

Malware Detection and Classification. The aim of *malware detection* is to identify whether the observed network traffic is generated by either legitimate applications or malware, whereas *malware classification* also tries to infer the malware type. Hence, these subjects both pertain to the security aspect. Besides, privacy aspects are involved when malware provokes *data exfiltration*, while the network management aspect is partially (resp. fully) affected by advances in malware detection (resp. classification).

Website Fingerprinting. The aim is to classify which website (and, at a finer level, which webpage) has been visited by a user via its traffic inspection, among a set of websites that an eavesdropper is monitoring. Since these sites may be targeted for censorship, this subject has a direct impact on the network privacy aspect.

Traffic Identification and Classification. Traffic identification (TI) consists in identifying a specific application (or protocol) among the network traffic, modeled as a binary classification task (i.e. application vs. other). Differently, *traffic classification (TC)* discriminates several applications (or protocols) among the network traffic

Table 2
Taxonomy of network traffic analysis subjects leveraging DL [21–31]. **Concerns**: Privacy (**P**), Security (**S**), Management (**M**). **DL Family:** AutoEncoder (AE), Convolutional Neural Network (CNN), Deep Belief Network (DBN), Deep Neural Network (DNN), Long Short-Term Memory (LSTM). "+" symbol indicates hybrid architectures.

	Concerns					
Subject	**P**	**S**	**M**	**Inference Task**	**DL Family**	**Paper**
Network Prediction	○	○	●	Time-series prediction	DBN	L. Nie *et al.*, 2017, *Proc. IEEE WCNC*, "Network Traffic Prediction based on Deep Belief Network in Wireless Mesh Backbone Networks." [21]
					CNN+LSTM	C. Zhang *et al.*, 2018, *Proc. ACM Mobihoc*, "Long-Term Mobile Traffic Forecasting Using Deep Spatio-Temporal Neural Networks." [22]
Anomaly Detection	○	●	◐	Binary classification	DNN	T. A. Tang, *et al.*, 2016, *Proc. IEEE WINCOM*, "Deep Learning Approach for Network Intrusion Detection in Software Defined Networking." [23]
					CNN	G. Marín, *et al.*, 2018, *ACM SIGCOMM Posters and Demos*, "Rawpower: Deep Learning based Anomaly Detection from Raw Network Traffic Measurements." [24]
Attack Classification	○	●	●	Multi-class classification	LSTM	A. Diro, *et al.*, 2018, *IEEE Commun. Mag.*, "Leveraging LSTM Networks for Attack Detection in Fog-to-Things Communications." [25]
					AE	N. Shone, *et al.*, 2018, *IEEE Trans. ETCI*, "A Deep Learning Approach to Network Intrusion Detection." [26]
Malware Detection	◐	●	◐	Binary classification	DNN	Y. C. Chen, *et al.*, 2017, *Proc. IEEE PIMRC*, "Deep Learning for Malicious Flow Detection." [27]
					CNN	H. Huang *et al.*, 2018, *IAOE iJET*, "Automatic Multi-Task Learning System for Abnormal Network Traffic Detection." [28]
Malware Classification	◐	●	●	Multi-class classification	CNN	W. Wang *et al.*, 2017, *Proc. IEEE ICOIN*, "Malware Traffic Classification Using Convolutional Neural Network for Representation Learning." [29]
Website Fingerprinting	●	○	○	Multi-class classification	AE CNN LSTM	V. Rimmer *et al.*, 2018, *Proc. NDSS*, "Automated Website Fingerprinting through Deep Learning." [30]
					AE CNN	P. Sirinam *et al.*, 2018, *Proc. ACM SIGSAC*, "Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning." [31]
Traffic Identification	●	◐	◐	Binary classification	Detailed analysis of DL-based TI/TC is reported in Tab. 3	
Traffic Classification	●	◐	●	Multi-class classification		

and constitutes a multi-class generalization of TI. Besides monitoring goals, TI and TC outcomes are capitalized in enforcing specific policing rules to the targeted application (or class of applications) traffic, such as prioritization, throttling, or blocking. This leads to a finer network management. Also, TI and TC are both tightly-coupled to the privacy aspect, for instance recognition of context-sensitive apps in mobile scenarios. Lastly, both also have security applications, such as detection of unexpected or unauthorized network services that, although not malicious in nature, either expose a wider attack surface or violate policies. A deeper analysis of TI and TC is the object of the next section.

3. Deep learning in traffic classification

In this section we provide an intuitive categorization, via a systematic taxonomy, of literature on DL-based TI and TC. We point out that a number of works have faced mobile TC in the last five years, under encrypted-traffic assumption, mostly using ML and based on bot-generated traffic [10,11].

On the other hand, the appeal of DL to TC is confirmed by several recent works providing initial design attempts of DL-based traffic classifiers, either not-mobile or not-encrypted. All these works use human-generated traffic datasets to evaluate their proposals. Also, from our thorough search, TC in the mobile and encrypted scenario by means of DL appears unexplored, save from our own preliminary analyses [13,40,39]. Indeed, in mobile and encrypted context, DL-based TC is challenged by a high number of apps generating similar traffic patterns, hard-to-learn app fingerprints (due to device/OS/version diversity, encryption, and scarce number of samples) and bot-generated traffic less representative of human behaviour.

Accordingly, in Table 3 we summarize and categorize each work performing TC via DL based on whether (*a*) it tackles TI, TC, or both, (*b*) it focuses on the mobile scenario, and (*c*) it tackles encrypted TC. For each study, we surface from a design viewpoint: (*i*) the traffic segmentation criterion employed (i.e. the *traffic object*), (*ii*) the input type used to feed the classifier, (*iii*) the specific DL classifier adopted, and (*iv*) whether the DL architecture is fed with multiple

Table 3
Recap of previous works adopting DL for TI/TC [18,19,17,29,32,28,33–36,20,37,38,13,39]. All the works use a single input type, and validate their approach on human-generated traffic. *Encrypted Traffic (ET). Multi-Modal (MM). Multi-Task (MT). Traffic object (TO):* biflow (BF), flow (F), HTTP session (H), packet (P); ☆ symbol indicates various applicable traffic objects. *Input Data:* Raw data of PCAP trace (PCAP), X^{th} layer of ISO/OSI model (LX). *DL Classifier:* AutoEncoder (AE), Auxiliary Classifier Generative Adversarial Network (AC-GAN), Bidirectional Gated Recurrent Unit (bi-GRU), Convolutional Neural Network (CNN), Deep Belief Network (DBN), Deep Neural Network (DNN), Long Short-Term Memory (LSTM), Stacked AutoEncoder (SAE), Variational AutoEncoder (VAE); "+" symbol indicates hybrid architectures.

TI/TC	Mobile	ET	TO	Input Data	Features	Classifier	MM	MT	Research
TI&TC	○	○	BF	TCP payload [1000 B]	○	SAE	○	○	Z. Wang, 2015, *Briefing Black Hat USA*, "The Applications of Deep Learning on Traffic Identification." [18]
TI&TC	○	●	BF	6 fields [20 packets]	○	LSTM+2D-CNN	○	○	M. Lopez-Martin *et al.*, 2017, *IEEE Access*, "Network Traffic Classifier with Convolutional and Recurrent Neural Networks for Internet of Things." [19]
TC	○	●	P	L2 payload [1500 B]	○	SAE, 1D-CNN	○	○	M. Lotfollahi *et al.*, 2017, *Preprint arXiv*, "Deep Packet: A Novel Approach for Encrypted Traffic Classification Using Deep Learning." [17]
TC	○	●	F/BF	PCAP [784 B] L4 payload [784 B]	○	2D-CNN	○	○	W. Wang *et al.*, 2017, *Proc. IEEE ICOIN*, "Malware Traffic Classification Using Convolutional Neural Network for Representation Learning." [29]
TI&TC	○	●	F/BF	PCAP [784 B] L4 payload [784 B]	○	1D-CNN	○	○	W. Wang *et al.*, 2017, *Proc. IEEE ISI'17*, "End-to-End Encrypted Traffic Classification with One-Dimensional Convolution Neural Networks." [32]
TI&TC	○	●	BF	PCAP [1024 B]	○	2D-CNN	○	●	H. Huang *et al.*, 2018, *IAOE iJET*, "Automatic Multi-Task Learning System for Abnormal Network Traffic Detection." [28]
TC	●	○	H	HTTP fields [28×36 B]	○	VAE	○	○	D. Li *et al.*, 2017, *Proc. 13th IEEE CIS*, "Traffic Identification of Mobile Apps based on Variational Autoencoder Network." [33]
TC	○	●	BF	ML&DL-selected features	●	DBN	○	○	H. Shi *et al.*, 2018, *Computer Networks*, "An Efficient Feature Generation Approach based on Deep Learning and Feature Selection Techniques for Traffic Classification." [34]
TI	○	●	BF	Flow-based statistics	●	AC-GAN	○	○	L. Vu *et al.*, 2018, *Proc. 8th ACM SoICT*, "A Deep Learning based Method for Handling Imbalanced Problem in Network Traffic Classification." [35]
TC	○	●	BF	Flow-based statistics	●	SAE	○	○	C. Zhang *et al.*, 2018, *Wiley Trans. on ETT*, "Deep Learning–based Network Application Classification for SDN." [36]
TC	○	●	BF	Flow-based statistics	●	DNN	○	●	H. Sun *et al.*, 2019, *IEEE Access*, "Common Knowledge Based and One-Shot Learning Enabled Multi-Task Traffic Classification." [20]
TC	○	●	BF	IP packet lengths	○	bi-GRU	○	○	C. Liu *et al.*, 2019, *Proc. IEEE INFOCOM'19*, "FS-Net: A Flow Sequence Network For Encrypted Traffic Classification." [37]
TI&TC	○	●	BF	PCAP [900 B]†	○	1D-CNN, LSTM, SAE	○	○	Y. Zeng *et al.*, 2019, *IEEE Access*, "*Deep-Full-Range*: A Deep Learning Based Network Encrypted Traffic Classification and Intrusion Detection Framework." [38]
TC	●	●	BF	PCAP [784 B] L4 payload [784 B] 6 fields [20 packets]	○	SAE, LSTM, 1D-CNN, 2D-CNN, LSTM+2D-CNN	○	○	G. Aceto *et al.*, 2019, *IEEE Trans. Netw. Service Manag.*, "Mobile Encrypted Traffic Classification Using Deep Learning: Experimental Evaluation, Lessons Learned, and Challenges." [13]
TC	●	●	BF	L4 payload [784 B] 4 fields [20 packets]	○	1D-CNN, GRU, LSTM+2D-CNN	●	○	G. Aceto *et al.*, 2019, *Elsevier ComNet*, 'MIMETIC: Mobile Encrypted Traffic Classification using Multimodal Deep Learning." [39]
TI&TC	●	●	★	*L4 payload [N Bytes]* *K fields [N_p packets]*	○	*DNN, AE, SAE, 1D-CNN, 2D-CNN, LSTM, (bi-)GRU*	●	●	*DL framework proposed in this paper*

† TCP/UDP headers and MAC addresses are removed.

types of input (i.e. multi-modal) and handles different TC tasks (i.e. multi-task). Furthermore, the flag *features* integrates the viewpoint (*ii*), stressing the use of handcrafted features as input data for DL architectures.

The above categorization prompts some caveats and *warning flags* in the adoption of the approaches reported in Table 3 to the mobile and encrypted context. Each of these is discussed hereinafter with regards to each separate aspect.

Regarding the *traffic objects*, we observe that the flows and biflows are the most-common choices under the encrypted-traffic assumption, whereas the HTTP sessions cannot be used in presence of encrypted traffic, due to the need to access the cleartext of transport layer payload to define such packet aggregation. Similarly, though DL-based TC can in principle be performed on a per-packet basis [17], the common labeling among packets of the same communication and the unavailability of cleartext payload in each encrypted packet discourage the use of this traffic object.

Regarding *inputs*, although raw payload is widely used as a relevant input type for DL architectures, the size and layer chosen vary from work to work and layer choices lower than transport level are likely to introduce bias in TC performance [40]. The same reasoning applies to byte-converted raw traces including also PCAP metadata [29,32] and inputs comprising source/destination port fields [19]. Equally important, the counter-productive application of DL to manually-extracted traffic features, as opposed to input data, nullifies a key asset of DL paradigm, that is, no need of human-expert intervention for designing informative features.

Referring to *DL architectures*, almost all the works, with the exception of [28,20], have proposed design solutions able to solve a single TC problem, in contrast with *multi-task* ones. Similarly, all previous DL traffic classifiers, except that developed in our previous work [39], have been designed based on a single input type. Furthermore, some research has used arbitrarily-shaped 2-D convolutional layers as the relevant block to handle a naturally 1-D input (i.e. a traffic packet series). Lastly, only three works (marked with + in the *classifier* column) started exploiting the composition possibilities offered by hybrid architectures allowed by the connectionist philosophy underlying *deep* learning.

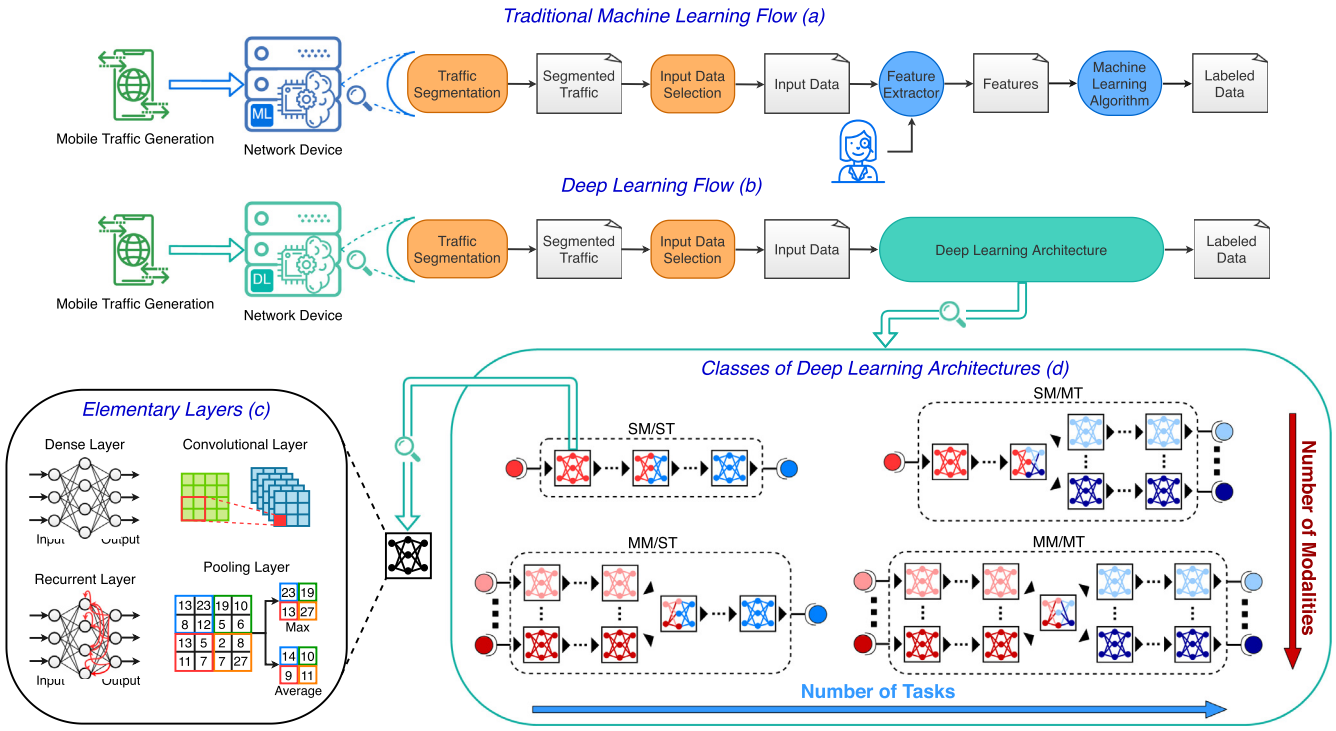


Fig. 1. Traditional ML flow (a) vs. DL flow (b). Bottom boxes depict the most used elementary layers (c) composing DL architectures and the four different classes of DL architectures (d), based on single/multiple input modalities (SM/MM) and single/multiple classification tasks (ST/MT).

The above analysis clearly underlines the *scattered nature* of the existing approaches pursuing DL-based TC, as well as their implicit (or partially-justified) design choices. This underlies the lack of a systematic design path, defining the key pillars for the conception and implementation of a practical DL-based mobile TC architecture, and motivates the need for a general framework explicitly capitalizing these aspects by molding them into rigorously-defined milestones. In this regard, we highlight the potentialities of the framework proposed herein in the last row of Table 3. We particularly emphasize its generalization ability according to different viewpoints, that is, with no constraints regarding specific design choices (e.g. traffic object, input data, and traffic classifier selection) and the capability of solving possibly *multiple TC tasks* fed with *multiple inputs*. We will provide the details of its design milestones in the next section.

4. A general framework for deep learning-based mobile encrypted traffic classification

In the following, we introduce and dissect our DL framework for mobile and encrypted TC. Fig. 1 illustrates the proposed framework in terms of its *workflow*, highlighting the key differences with respect to a traditional *ML workflow* (cf. Fig. 1(a)). Specifically, the mobile traffic flowing over a *network device* is captured and segmented into defined packet aggregates of traffic (*traffic segmentation*). Then, from each traffic object, raw input data is selected (*input data selection*) and used to feed a *DL architecture*, in charge of labeling the segmented traffic. As aforementioned, a DL-based classification system obviates to the need of an expert-handcrafted *feature extractor* module, by automatically learning the complex and informative features for an accurate TC.

To design a DL system for TC, the following milestone design choices should be made:

- *Traffic object*: the traffic aggregate atom which induces the segmentation criterion.
- *Type(s) of input data:* the number and sets of input selected from each traffic object to feed the DL architecture.
- *Classification tasks:* the number and the type of TC problems that a single DL-based traffic classifier is in charge to solve simultaneously.
- *DL architecture:* the peculiar DL architecture (e.g. the composition instance of elementary learning layers), coping with input and output constraints originating from the design choices concerning the *type(s) of input data* and *classification tasks*.

We now discuss each design element of our DL-based mobile TC framework separately.

Traffic Object. A key choice regards how raw traffic is segmented into multiple discrete units. Considering mobile and encrypted traffic, we here suggest the use of either *flows* or *biflows* [9], with the latter achieving better performance in most related works [29,32]. In detail, a flow is a stream of packets sharing the same 5-tuple (i.e. source IP and port, destination IP and port, and transport-level protocol), thus taking into account their directions. Differently, in a biflow the source and destination (IP, port) pairs can be swapped. In both cases the termination is based on a user-defined timeout. Other appealing choices are given by the *TCP connection* and the *service burst*. The former differs from the biflow only in the initiation and termination heuristics [9]. The latter has been recently adopted in mobile TC [11,10], and is defined by aggregating packets with an inter-packet time smaller than a given "burst" threshold and then grouping those that belong to biflows with the same transport protocol and destination (IP, port) pair. Still, the service bursts have not seen their direct application to security and policy enforcement so far, as opposed to the ubiquitous (bi) flows.

Types of Input Data. The recommended types of input data [40] of a generic TC object ingested by DL architectures may be roughly grouped within two categories: (i) the first N bytes of the payload [29,18,32] at transport level or higher; (ii) K selected informative data fields of the first N_p packets [19,37]. In the first case, the payload data being fed to the DL architecture is represented in binary

format, arranged in a byte-wise fashion and normalized within $[0, 1]$. In the second case, the type of input data is represented by selected protocol fields, not pertaining to the explicit inspection of encrypted payload (e.g. the packet size) of the first N_p packets. In both cases, instances longer than the considered fixed-length (N or N_p) data inputs are truncated to the designed length of bytes (N) or packets (N_p), whereas shorter instances are zero-padded. Based on the discussion of previous section, both recommended types correspond to unbiased input data and imply DL-based mobile traffic classifiers suited for *early TC* [41], namely using only the first segments of traffic aggregate to take a decision.

Classification Tasks. Given the polyvalent traffic nature, multi-task learning [28,20] is becoming attractive as a viable means to design a single TC system able to label traffic according to different classification *views*, for example to discriminate simultaneously among the sets of applications and user activities. In this respect, we propose the DL paradigm as the perfect suit for the design of multi-task classifiers, since DL architectures can be effectively conceived and trained in a multi-output form, namely to minimize a multi-objective loss function.

DL Architecture. Our framework defines four classes of DL architectures, as shown in Fig. 1(d), based on two orthogonal aspects:

- Whether they are fed with a single type (*single-modal, SM*) or multiple types (*multi-modal, MM*) of input (*modalities*), to capitalize complementary viewpoints of the same traffic object (e.g. using the first N bytes of transport-level payload together with the informative data fields of the first N_p packets).
- Whether they are in charge of providing inference for one (*single-task, ST*) or multiple (*multi-task, MT*) TC problems (e.g. inferring both the traffic-type and the specific application generating a (bi)flow).

These classes of DL architectures are obtained by composition of *elementary layers* [7,16], whose common choices are *dense*, *convolutional*, *pooling*, and *recurrent layers* (Fig. 1(c)):

- *Dense layers* are the simplest atoms of feed-forward DL architectures, consisting of an affine matrix operation (i.e. a linear transformation) on inputs, followed by an entry-wise activation function. It is worth noticing that the encoding layer of an Auto-Encoder [13,17,18,33,36,38], when used for TC, and Deep Belief/Neural Networks [8,20,34] belong to this category.
- *Convolutional layers* are the basic building blocks of Convolutional Neural Networks (CNNs), made of a set of translation-invariant filters with a limited extent (i.e. the "receptive field") which are convolved with the input, with the aim of extracting the features of a certain input region. The most common architectures in TC adhere to a 1-D [13,17,32,38,39] or a 2-D [13,19,28,29,39] layout, depending on the specific input nature (or reshaping).
- *Pooling layers* are other key components of CNNs and typically follow a convolutional layer. They perform the down-sampling of the intermediate representations from convolutional layers, with the aim of complexity reduction and overfitting mitigation. Max-pooling [19,29,32] and average-pooling [17] are the most commonly employed in TC architectures.
- *Recurrent layers* present loopy connections and have in Long Short-Term Memory [13,19,38] and Gated Recurrent Unit [37,39] their most popular variants. These are in charge of *recalling* values over time, via a state vector, and accept as input a vector sequence. Differently, they output either the final state or its entire time-evolution. Note that Long Short-Term Memory and Gated Recurrent Unit layers can be also conceived in an improved "bidirectional" form, i.e. their internal representation is split into forward and backward directions.

5. Experimental validation

In this section, we test two actual implementations of the proposed DL framework for mobile and encrypted TC based on three recent human-generated mobile traffic datasets. First, we describe the aforementioned datasets and the key performance indicators (KPIs) adopted for evaluation of TC effectiveness (Section 5.1). Secondly, we show and discuss the experimental results obtained (Section 5.2).

5.1. Description of datasets and KPIs

We validate our framework based on three mobile encrypted datasets (cf. Table 4) suitable for ST classification, either recommended or produced by a global mobile solution provider and collected by *human users* using both Android and iOS apps, as opposed to works based on bot-generated mobile traffic [10,11]. The traces capture traffic generated by users running a single app at a time on a given device/OS, allowing to label traces with the associated known ground truth. The TC object chosen is the *biflow*, due to its suitability for mobile and encrypted traffic and fruitful adoption in most DL-based TC works (cf. Table 3).

The first (*binary*) dataset, named *FB/FBM*, was collected in the ARCLAB laboratory of University of Napoli Federico II. In detail, the capture sessions pertain to either Facebook (FB) or Facebook Messenger (FBM) with the aim of *billing differentiation* between similar apps. To explore app diversity, users were requested to perform different activities (e.g. posting contents, commenting, liking, sending messages, making (video-) calls, etc.). As the apps required user login the *sign-in*, *first login*, and *already logged-in* scenarios have been explored as well. More than 280 users have been involved in the dataset collection on a volunteering base, with each user performing 12 capture sessions of ≈ 5 minutes. Background traffic was removed in the post-capture stage, leveraging the network system-calls (e.g. connect, bind, getsockname, etc.), traced on the mobile devices (by means of the *strace*[1] utility via the Android Debug Bridge[2]) to identify the biflows associated with the user-controlled app and discard the rest. In detail, we relate each socket descriptor to the name of the Android package originating the call. Given this capture setup, the traces result anonymized. Indeed, no identification information is associated to the (local) IP address and purposely created user accounts have been used for all the apps. Of the $31k$ biflows collected, $17.5k$ and $13.5k$ instances were generated by FB and FBM, respectively, corresponding to a 44/56 percent share. We refer to [42] for detailed information and to http://traffic.comics.unina.it/mirage/ for downloading an open super set of the FB/FBM dataset.

The second and third (*multi-class*) datasets, named *Android* and *iOS*, are generated from different apps on Android and iOS devices, respectively, and are explored with a *service prioritization* goal. The traces were collected by the provider and shared, already anonymized and cleaned from background traffic, under a non-disclosure agreement. The detailed report of biflow statistics for each class can be found in [12], where the Android and iOS datasets were employed for ML-based (handcrafted) mobile TC.

The performance evaluation resorts to a stratified 10-fold cross-validation: for each KPI, we report the mean (μ) and standard deviation (σ), as a $\mu \pm 3\sigma$ confidence interval. The main KPIs considered are the *accuracy*, being the fraction of correctly classified samples, and the well-known *F-measure*, defined on a per-class basis as the harmonic mean of precision (i.e. the fraction of per-class predictions that are correct) and recall (i.e. the class-

[1] https://strace.io/
[2] https://developer.android.com/studio/command-line/adb.html

Table 4
Details of the datasets employed in experimental evaluation. Average duration of each trace is ≈ 5 minutes.

Dataset	Type (#Apps)	#Traces	#Biflows	%ET	OS Version	Collection	Source	Aim
FB/FBM	Binary(2)	> 1100	31.0k	91%	Android 6.0.1	May '17 - Mar. '18	Self-generated@UniNa	Billing differentiation
Android	Multi-class (49)	607	55.5k	47%	4.2.2 - 6.0.1	Apr. '15 - Jan. '17	Mobile solutions provider	Service prioritization
iOS	Multi-class (45)	419	37.2k	60%	7.0 - 10.0	Sept. '14 - Jan. '17	Mobile solutions provider	Service prioritization

conditional accuracy). Specifically, we employ the arithmetic mean of per-class F-measures, that is the *macro F-measure*.

Moreover, we investigate the use of a *reject option*, which allows the traffic classifiers to assign labels only to the biflows which can be labeled reliably, namely those whose highest class-prediction probability exceeds a threshold γ. Differently, the decisions on the other biflows are *censored*. In this respect, we take into account both the generic KPI and *classified ratio (CR)*, namely the percentage of reliably labeled biflows, vs. the censoring threshold γ. Hence, each classifier can improve its KPI with γ at the price of a reduced CR.

Indeed, tuning γ enables a fine-grained control of the classifiers and further (useful) flexibility to mobile TC [11]. Specifically, given the high number of flows commonly generated by mobile apps, there is an excellent chance of identifying them only considering the more characteristic flows, namely those corresponding to a classification confidence above γ.

Furthermore, to analyze the computational complexity of considered instances of our framework, we report also the time needed for their training, in terms of *Run-Time Per-Epoch (RTPE)*[3]. This KPI is of interest due to frequent re-training requirements of mobile TC, due to aging of training data as a result of app and OS updates [11].

Lastly, we perform a *calibration analysis*, that allows to check whether the class-probability estimates are representative of the true-class (posterior) probabilities. Indeed, a miscalibrated classifier produces confidences (i.e. class-prediction probabilities) that could not represent the true probabilities, leading to either excessively optimistic or pessimistic decisions. Specifically, we leverage *reliability diagrams* that show the accuracy as a function of confidence and are obtained by partitioning the predictions into M equally-spaced bins and calculating the accuracy of each bin. If the classifier is perfectly calibrated, then the diagram should plot the identity function (e.g. operating with 70 percent confidence leads to 70 percent accuracy) and any deviation from a perfect diagonal represents a miscalibration. In addition to reliability diagrams, for conciseness we report also the *Expected Calibration Error (ECE)*. The latter KPI is defined as the weighted (based on the number of samples) mean, evaluated over all the bins, of the difference between accuracy and confidence [43].

5.2. Experimental results

Herein, we investigate the performance of three different mobile (encrypted) traffic classifiers. The first is the ML-based state-of-the-art Random Forest (*Base-ML*), taking as input 40 handcrafted input features, namely the best-ranked statistics (i.e. min, max, mean, standard deviation, variance, mean absolute deviation, skewness, kurtosis, and percentiles) on the basis of the Gini impurity score, calculated on the sequences of upstream, downstream, and bidirectional IP packet sizes [11].

On the other hand, the latter two are *different DL-based TC implementations* of our framework, trained for 90 epochs—as also suggested in related works [13,19,32]—with adaptive moment estimation optimizer (with a batch size of 50) and randomly-initialized parameters. Also, to avoid overfitting, both have been equipped with an early-stopping procedure set with a 1 percent threshold and evaluated on the training accuracy.

The *first* implemented instance is the best-performing SM-DL approach (i.e. taking only one input type) devised for the mobile TC task, namely an optimized 1D-CNN fed with the first $N = 784$ bytes of L4 payload, being the current DL baseline (*Base-DL*) [13]. The *second* implemented instance is a *Proposed* (drawn from our framework) MM-DL hybrid architecture using both the recommended unbiased input sets, namely the first $N = 576$ bytes of L4 payload (first modality) and four informative fields[4] of the first $N_p = 12$ packets (second modality). For the first modality, we adopt two "light" 1D-convolutional layers (16 and 32 filters and rectifier activations), each followed by a 1D max-pooling layer, and one dense layer (256 nodes). For the second modality, we use a Gated Recurrent Unit (64 nodes) and one dense layer (256 nodes). Lastly, the intermediate outputs of the two branches are stacked and fed to a shared dense layer (128 nodes).

We highlight that hyperparameter optimization for both these architectures has been performed either via trial-and-error procedures for some parameters (e.g. the architecture depth), while a grid search [13,39] has been numerically evaluated for some others (e.g. the input size).

Fig. 2a and c report, in dotted and solid lines, respectively, both the *F-measure* and *CR* vs. the censoring threshold γ on the three datasets. The results show that the Proposed MM-DL classifier gains either in terms of F-measure or CR over both Base-DL and Base-ML for all the datasets considered. For example, in an uncensored case (CR = 100, percent) it gains up to +7.12 and +9.28 percent F-measure (on the iOS dataset) over Base-DL and Base-ML, respectively. Conversely, with a 90 percent target F-measure, it gains +5 and +10 percent CR over Base-DL and Base-ML, respectively. It is worth noting that, for the hardest classification task, that is discriminating between the very similar FB and FBM apps, the Proposed classifier also guarantees a (less-evident) improvement of +1.10 percent over the best Base-ML baseline in the uncensored setting, and a significant gain in terms of CR otherwise (e.g. +20 percent with $\gamma = 0.7$).

For completeness, in Figs. 2d and 2f we report also the training-phase RTPE of the two DL architectures. Interestingly, the proposed MM-DL classifier requires a RTPE lower than Base-DL, with a 3.5× speed in the hardest classification (i.e. Android) setup and a less severe trend with the size of the TC task L (i.e. moving from the FB/FBM dataset to the iOS and Android datasets), corresponding to +41 percent complexity burden as opposed to +64 percent for Base-DL. This is the outcome of shorter inputs and computationally-lighter layers. For instance, the proposed approach requires ≈ 56 minutes training in the hardest TC task, thus being a good candidate for frequent re-training in practical mobile contexts.

To deepen this investigation, Table 5 reports the number of parameters to be trained (viz. learned) for both the Proposed MM-DL and the Base-DL architectures. It can be noticed that the RTPE is strongly dependent on this number of parameters, with the Proposed classifier having ≈ 6.2× and ≈ 3.6× fewer trainable

[3] The training phase of DL classifiers is performed on multiple epochs in a cyclic fashion.

[4] IP packet size, direction, inter-arrival time, and TCP window size (set to zero for UDP packets).

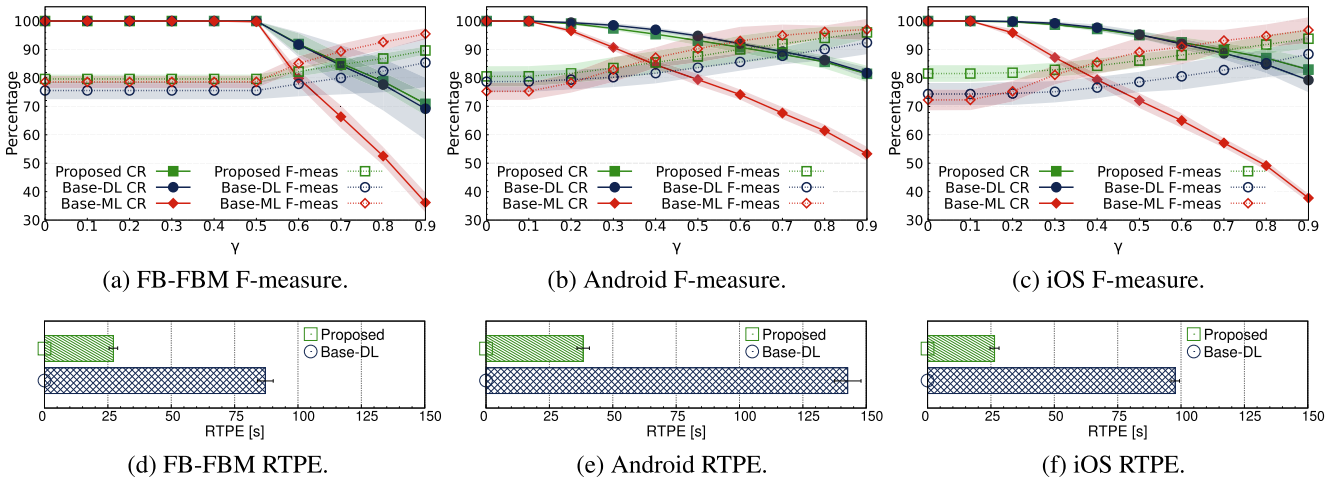


Fig. 2. Top row: F-measure and ratio of classified samples (CR) [%] vs. censoring threshold γ of MM/ST (Proposed) approach vs. best SM/ST (Base-DL) and ML-based (Base-ML) baseline classifiers. Bottom row: run-time per epoch (RTPE) of Proposed and Base-DL classifiers. Note that RTPE is not defined for Base-ML.

Table 5
Number of trainable parameters (in millions) of the Proposed MM-DL hybrid architecture and Base-DL baseline.

	FB/FBM	Android	iOS
Proposed	0.9346	1.6202	1.6176
Base-DL [13]	5.8223	5.8705	5.8664

parameters when trained on the FB/FBM dataset and the multi-class datasets, respectively. This outcome results from shorter inputs and composition of "lighter" elementary layers leveraged in the Proposed approach with respect to the Base-DL baseline.

In view of the results obtained, we can affirm that a DL architecture having a more complex structure and inputs does not guarantee a more accurate classification and may instead incur into overfitting issues. On the other hand, the Proposed MM-DL solution, being able to exploit multi-modality with the right amount of complexity, is able to obtain both a higher F-measure and a lower RTPE, despite using a similar training procedure (e.g. a comparable number of epochs, same optimizers and initialization).

To deepen the performance analysis of the mobile traffic classifiers taken into account, Fig. 3 reports the *reliability diagrams* and the *Expected Calibration Error* (ECE) of the Proposed MM-DL architecture, along with the Base-DL and Base-ML baselines for the Android (top row) and iOS (bottom row) datasets. In this case, we do not report performance for the FB/FBM dataset, as similar trends have been observed also in this case. Additionally, since it is a binary dataset, the dynamic of reliability diagrams is reduced since the class prediction probability is always higher than 0.5.

It can be seen that the Proposed classifier results to be better calibrated with respect to the two considered baselines, resulting in an ECE being *less than half* of that of Base-DL and Base-ML. This applies to both datasets. Furthermore, by looking at the behavior of each classifier on the two multi-class datasets, we can observe (*i*) an invariance of the miscalibration pattern and (*ii*) a different ECE trend.

Specifically, referring to point (*i*), both the DL-based classifiers interestingly exhibit almost always (except for the last bin) a miscalibration that tends to be over-confident (optimistic) in its predictions (i.e. in each bin the confidence is higher than the accuracy). This effect can be attributed to a slight overfitting phenomenon and is one of the distinctive characteristics of DL architectures [43] (although MM architectures reduce it). Differently, the Base-ML classifier shows an accuracy always higher than the related confidence, which can be attributed to a slight bias due to its "ensemble" nature (i.e. a Random Forest whose decision is taken based on the average of multiple parallel decision trees).

On the other hand, referring to point (*ii*), the Proposed MM-DL performs better on the Android dataset, whereas the two baselines are more effective on the iOS one. However, a relative performance inversion is observed between the two baselines when passing from the Android to the iOS dataset, namely Base-ML performs better than Base-DL on Android, whereas Base-DL outperforms Base-ML on iOS. This confirms that the difference in software and hardware ecosystems of these mobile operating systems impact also on the traffic proprieties and consequently on the app discrimination ability of mobile traffic classifiers, being not generalizable between the two cases.

6. Discussion and future perspectives

In this work we envisioned a DL application to the field of network traffic analysis, focusing on the identification and classification of mobile and encrypted traffic. The result of our study is a DL-based TC framework able to capitalize heterogeneous input data from mobile traffic and solve multiple TC tasks at the same time.

By means of our framework we highlight several shortcomings with previous DL-based attempts to TC, namely: (*i*) traffic segmentation is often implicit or overlooked; (*ii*) surprisingly, some studies preliminarily extract features from data, instead of leveraging DL for that; (*iii*) input data selection in some studies causes *biased inputs* being fed to DL algorithm, jeopardizing the validity of results; (*iv*) the choice of DL architecture is seldom well-matched with the nature of input data, encouraging MM approaches instead. We validated our framework on mobile datasets from human users, and results have confirmed the strong appeal of such paradigm, which outperformed current ML-based state-of-the-art mobile TC approaches and, also, the current DL-based baseline in encrypted TC, attaining up to +9.28 percent F-measure.

Further analyses on the inner structure of DL networks can be conducted along the lines of *explainable AI* [44], a recent field of study that has yet to see application to TC. Complementary to this, the proposed DL-based framework suggests a number of research directions, the most prominent described in the following. First, further performance gain is foreseen via exploitation of massive unsupervised data for improved learning, along with the use of pre-trained architectures and sophisticated DL layers. Additionally, although some efforts have been made from a system viewpoint,

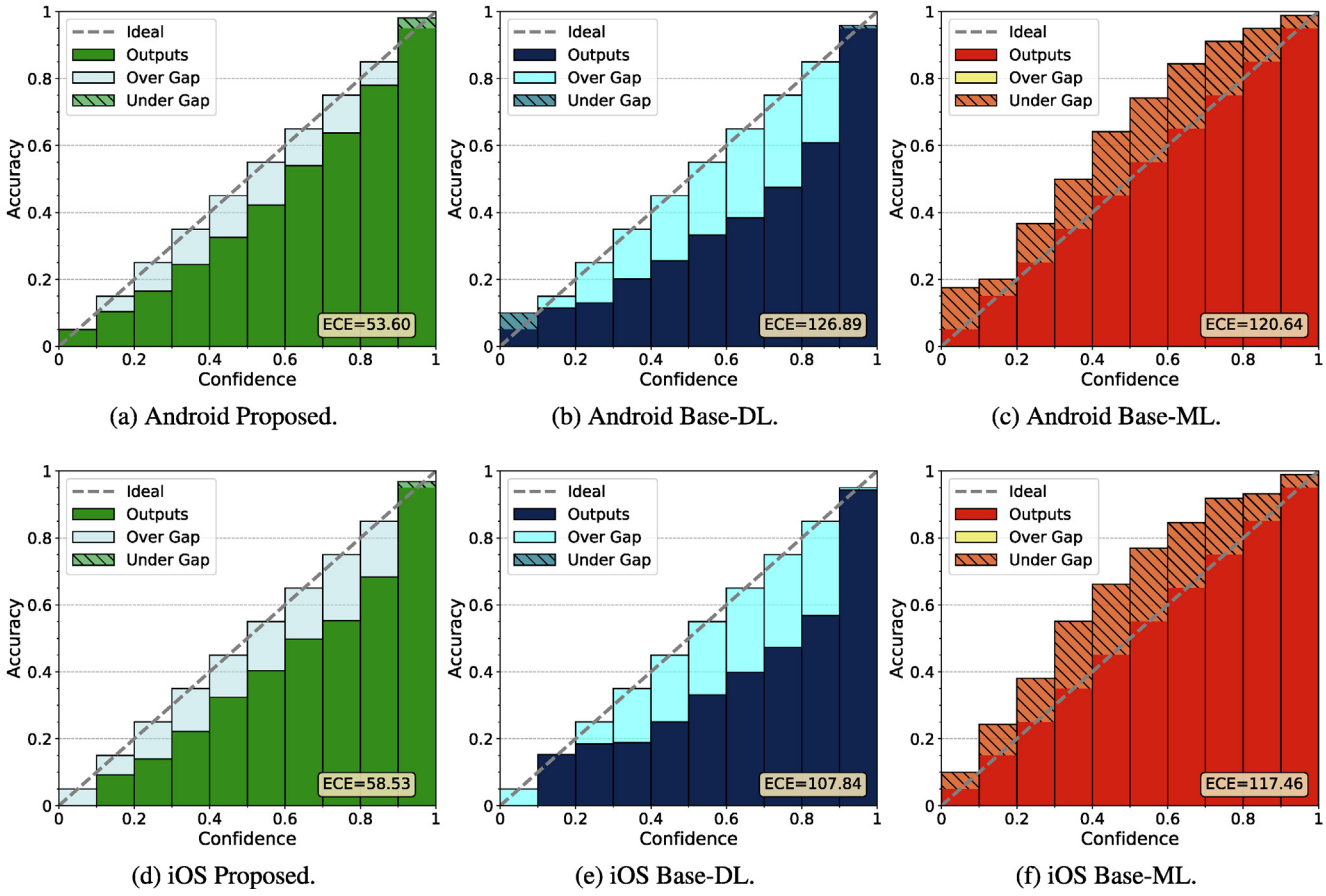


Fig. 3. Reliability diagrams of Proposed (a & d), Base-DL (b & e), and Base-ML (c & f) mobile traffic classifiers trained on the Android (top row) and iOS (bottom row) datasets. Bin width is $M = 10$. Under and over gap represent an under-confident (pessimistic) and over-confident (optimistic) miscalibration pattern, respectively.

design and real-world implementations of accurate MM/MT-DL architectures are still unexplored.

Such real-world implementations should be able to operate under an *open-world* assumption, that is they should be able to handle (during the operational phase) unknown classes not present in the training set (viz. *open-set TC*).

Analogously, the design of DL architectures able to cope with more challenging—but promising—TC objects (e.g. the service burst [11]) is of clear interest. Finally, the increased training complexity of DL-based architectures paves the way to a justified and sensible adoption of the Big Data paradigm to mobile TI and TC [45].

CRediT authorship contribution statement

Giuseppe Aceto: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing - original draft, Writing - review & editing. **Domenico Ciuonzo:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing - original draft, Writing - review & editing. **Antonio Montieri:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - original draft, Writing - review & editing, Visualization. **Antonio Pescapé:** Conceptualization, Methodology, Supervision, Project administration.

Declaration of Competing Interest

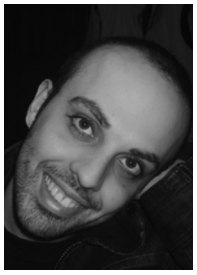
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

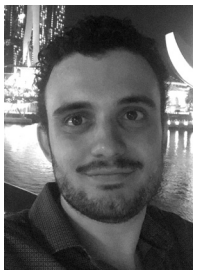
[1] N. Heuveldop, et al., Ericsson mobility report, Ericsson AB, Technol. Emerg. Business, Stockholm, Sweden, Tech. Rep. EAB-17.
[2] M. Cooney, Cisco: How AI and machine learning are going to change your network.,https://tinyurl.com/network-world-18-cisco-ai, [Online; accessed day], 2018.
[3] A. Callado, C. Kamienski, G. Szabó, B.P. Gero, J. Kelner, S. Fernandes, D. Sadok, A survey on internet traffic identification, IEEE Commun. Surv. Tutorials 11 (3).
[4] T.T. Nguyen, G. Armitage, A survey of techniques for internet traffic classification using machine learning, Commun. Surveys Tuts. 10 (4) (2008) 56–76, ISSN 1553-877X.
[5] V. Carela-Español, P. Barlet-Ros, M. Solé-Simó, A. Dainotti, W. de Donato, A. Pescapè, K-Dimensional Trees for Continuous Traffic Classification, in: 2nd International Workshop on Traffic Monitoring and Analysis (TMA), 141–154, 2010.
[6] N. Zeng, H. Zhang, W. Liu, J. Liang, F.E. Alsaadi, A switching delayed PSO optimized extreme learning machine for short-term load forecasting, Neurocomputing 240 (2017) 175–182.
[7] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT Press, 2016.
[8] N. Zeng, Z. Wang, H. Zhang, W. Liu, F.E. Alsaadi, Deep belief networks for quantitative analysis of a gold immunochromatographic strip, Cognitive Comput. 8 (4) (2016) 684–692.
[9] A. Dainotti, A. Pescapè, K.C. Claffy, Issues and future directions in Traffic Classification, IEEE Network 26 (1) (2012) 35–40.
[10] T. Stöber, M. Frank, J. Schmitt, I. Martinovic, Who do you sync you are? smartphone fingerprinting via application behaviour, in: ACM 6th conference on Security and privacy in wireless and mobile networks (WISEC), 7–12, 2013.
[11] V.F. Taylor, R. Spolaor, M. Conti, I. Martinovic, Robust smartphone app identification via encrypted network traffic analysis, IEEE Trans. Inf. Forensics Secur. 13 (1) (2018) 63–78.
[12] G. Aceto, D. Ciuonzo, A. Montieri, A. Pescapè, Multi-classification approaches for classifying mobile app traffic, J. Network Comput. Appl. 103 (2018) 131–145.
[13] G. Aceto, D. Ciuonzo, A. Montieri, A. Pescapé, Mobile encrypted traffic classification using deep learning: experimental evaluation, lessons learned, and challenges, IEEE Trans. Network Service Manage. 16 (2) (2019) 445–458.

[14] A. Dainotti, F. Gargiulo, L.I. Kuncheva, A. Pescapé, C. Sansone, Identification of Traffic Flows Hiding behind TCP Port 80, in: 2010 IEEE International Conference on Communications, 2010, pp. 1–6, ISSN 1550-3607.
[15] B. Saltaformaggio, H. Choi, K. Johnson, Y. Kwon, Q. Zhang, X. Zhang, D. Xu, J. Qian, Eavesdropping on fine-grained user activities within smartphone apps over encrypted network traffic, in: 10th USENIX Workshop on Offensive Technologies (WOOT 16), 2016.
[16] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26.
[17] M. Lotfollahi, R. Shirali, M.J. Siavoshani, M. Saberian, Deep packet: a novel approach for encrypted traffic classification using deep learning, Soft. Comput. (2017) 1–14.
[18] Z. Wang, The applications of deep learning on traffic identification, Black Hat USA, Las Vegas (2015).
[19] M. Lopez-Martin et al., Network traffic classifier with convolutional and recurrent neural networks for internet of things, IEEE Access 5 (2017) 18042–18050.
[20] H. Sun, Y. Xiao, J. Wang, J. Wang, Q. Qi, J. Liao, X. Liu, Common knowledge based and one-shot learning enabled multi-task traffic classification, IEEE Access 7 (2019) 39485–39495.
[21] L. Nie, D. Jiang, S. Yu, H. Song, Network traffic prediction based on deep belief network in wireless mesh backbone networks, in: IEEE Wireless Communications and Networking Conference (WCNC), 1–5, 2017.
[22] C. Zhang, P. Patras, Long-term mobile traffic forecasting using deep spatio-temporal neural networks, in: 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc), 231–240, 2018.
[23] T.A. Tang, L. Mhamdi, D. McLernon, S.A.R. Zaidi, M. Ghogho, Deep learning approach for network intrusion detection in software defined networking, in: IEEE International Conference on Wireless Networks and Mobile Communications (WINCOM), 258–263, 2016.
[24] G. Marín, P. Casas, G. Capdehourat, RawPower: deep learning based anomaly detection from raw network traffic measurements, in: ACM SIGCOMM Conference on Posters and Demos, 2018, pp. 75–77.
[25] A. Diro, N. Chilamkurti, Leveraging LSTM networks for attack detection in fog-to-things communications, IEEE Commun. Mag. 56 (9) (2018) 124–130.
[26] N. Shone, T.N. Ngoc, V.D. Phai, Q. Shi, A deep learning approach to network intrusion detection, IEEE Trans. Emerging Top. Computat. Intell. 2 (1) (2018) 41–50.
[27] Y.-C. Chen, Y.-J. Li, A. Tseng, T. Lin, Deep Learning for malicious flow detection, in: IEEE 28th International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 1–7, 2017.
[28] H. Huang et al., Automatic multi-task learning system for abnormal network traffic detection, Int. J. Emerging Technol. Learn. 13 (04) (2018) 4–20.
[29] W. Wang, M. Zhu, X. Zeng, X. Ye, Y. Sheng, Malware Traffic Classification using convolutional neural network for representation learning, in: IEEE International Conference on Information Networking (ICOIN), 712–717, 2017.
[30] V. Rimmer, D. Preuveneers, M. Juarez, T. Van Goethem, W. Joosen, Automated Website Fingerprinting through Deep Learning, in: Network and Distributed Systems Security Symposium (NDSS), 2018.
[31] P. Sirinam, M. Imani, M. Juarez, M. Wright, Deep fingerprinting: undermining website fingerprinting defenses with deep learning, in: ACM Conference on Computer and Communications Security (SIGSAC), 1928–1943, 2018.
[32] W. Wang, et al., End-to-end encrypted Traffic Classification with one-dimensional convolution neural networks, in: IEEE Int. Conf. on Intelligence and Security Informatics (ISI), 43–48, 2017.
[33] D. Li, Y. Zhu, W. Lin, Traffic Identification of Mobile Apps Based on Variational Autoencoder Network, in: 13th IEEE International Conference on Computational Intelligence and Security (CIS), 2017, pp. 287–291.
[34] H. Shi, H. Li, D. Zhang, C. Cheng, X. Cao, An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification, Comput. Netw. 132 (2018) 81–98.
[35] L. Vu, C.T. Bui, Q.U. Nguyen, A deep learning based method for handling imbalanced problem in network traffic classification, in: ACM 8th International Symposium on Information and Communication Technology (SoICT), 2017, pp. 333–339.
[36] C. Zhang, X. Wang, F. Li, Q. He, M. Huang, Deep learning–based network application classification for SDN, Wiley Trans. Emerging Telecommun. Technol. 29 (5) (2018) e3302.
[37] C. Liu, L. He, G. Xiong, Z. Cao, Z. Li, FS-Net: A Flow Sequence Network For Encrypted Traffic Classification, in: IEEE Conference on Computer Communications (INFOCOM), 1171–1179, 2019.
[38] Y. Zeng, H. Gu, W. Wei, Y. Guo, *Deep – Full – Range*: a deep learning based network encrypted traffic classification and intrusion detection framework, IEEE Access 7 (2019) 45182–45190.
[39] G. Aceto, D. Ciuonzo, A. Montieri, A. Pescapè, MIMETIC: mobile encrypted traffic classification using multimodal deep learning, Comput. Netw. 165 (2019) 106944.
[40] G. Aceto, D. Ciuonzo, A. Montieri, A. Pescapè, Mobile Encrypted Traffic Classification Using Deep Learning, in: IEEE/ACM Network Traffic Measurement and Analysis Conference (TMA), 569–574, 2018b.
[41] L. Bernaille, R. Teixeira, K. Salamatian, Early application identification, in: ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT), 6, 2006.
[42] G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, A. Pescapé, MIRAGE: Mobile-app Traffic Capture and Ground-truth Creation, in: 2019 4th International Conference on Computing, Communications and Security (ICCCS), IEEE, 1–8, 2019b.
[43] C. Guo, G. Pleiss, Y. Sun, K.Q. Weinberger, On Calibration of Modern Neural Networks, in: 34th International Conference on Machine Learning (ICML), 1321–1330, 2017.
[44] H. Hagras, Toward Human-Understandable, Explainable AI, IEEE Computer 51 (9) (2018) 28–36.
[45] G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, A. Pescapé, Know your big data trade-offs when classifying encrypted mobile traffic with deep learning, in: 2019 Network Traffic Measurement and Analysis Conference (TMA), IEEE, 121–128, 2019c.

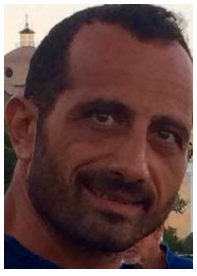
Giuseppe Aceto is an Assistant Professor at University of Napoli Federico II. He has a PhD in telecommunication engineering from the same University. His work falls in monitoring of network performance and security (focusing on censorship) both in traditional and SDN network environments. He is also working on bioinformatics and ICTs applied to health. He is the recipient of a best paper award at IEEE ISCC 2010, and 2018 Best Journal Paper Award by IEEE CSIM.



Domenico Ciuonzo is an Assistant Professor at University of Napoli Federico II. He holds a Ph.D. in Electronic Engineering from the University of Campania "L. Vanvitelli" and, from 2011, he has held several visiting researcher appointments. Since 2014, he has been in the editorial board of different IEEE Elsevier and IET journals. His research concerns data fusion, wireless sensor networks, machine learning and network analytics. He is an IEEE Senior Member.



Antonio Montieri is a Postdoctoral Researcher at the Department of Electrical Engineering and Information Technology of the University of Napoli Federico II since 2017. He has received a Ph.D. in Information Technology and Electrical Engineering from the same University in 2020. His work concerns network measurements, (encrypted and mobile) traffic classification and modeling, monitoring of cloud network performance. Antonio has co-authored 22 papers in international journals and conference proceedings.



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