

# Towards efficient real-time traffic classifier: A confidence measure with ensemble Deep Learning

Ola Salman<sup>\*</sup>, Imad H. Elhadj, Ali Chehab, Ayman Kayssi

Electrical and Computer Engineering Department, American University of Beirut, Beirut, Lebanon

## ARTICLE INFO

### Keywords:

Internet traffic  
Real-time classification  
Internet of Things  
Traffic classification  
Traffic management

## ABSTRACT

Traffic classification is a key network function for managing both Quality of Service (QoS) and security. While some traffic classification applications (e.g. QoS based path allocation) can tolerate delays, other applications (e.g. attack detection) are time critical. In this context, early traffic classification has been proposed based on the first few packets of flows. However, the choice of the number of packets to inspect is method dependent and based on empirical assessment without considering the information carried by these packets (features). In this paper, we aim at identifying the sufficient number of packets,  $N$ , that guarantees high classification accuracy while optimizing the response time, based on both empirical classification results and information theory. We propose a confidence measure based on the variations in the model training accuracy and the average mutual information among the packets' features and the label vector. This measure is then used to define the value of  $N$ , which optimizes the trade-off between the time overhead and the classification accuracy. In addition, we propose an ensemble Deep Learning (DL)-based classifier model to enhance the classification accuracy by training successive DL models based on the traffic stream. The proposed ensemble method output is based on the average of the individual classifiers predictions. The experimental results show that when using the proposed confidence measure, we can achieve good classification accuracy at early phase of the flow. In addition, using the proposed ensemble method presents enhancement in the early classification accuracy. Consequently, combining the ensemble method with the confidence measure criteria allows for striking a good balance between high accuracy and fast response time.

## 1. Introduction

Internet traffic is in continuous growth. With the emergence of the Internet of Things (IoT), it is expected to have more than 29 billions devices connected to the Internet with more than 4.8 zettabytes of data generated per year by 2022 [1]. IoT systems include new types of connected devices, deployed applications, and generated traffic. Having different QoS and security requirements, traffic classification is key for network management. Different methods have been proposed to classify traffic based on (1) the port number, and (2) the packet content. However, today's applications rely on dynamic ports and data encryption. In addition, traffic tunneling and anonymization have emerged to protect users' privacy and to hide the traffic context information. Machine Learning (ML) based traffic classification methods have been proposed to classify traffic using statistical features. Different features and methods have been employed. Most of them consider statistical measures: minimum, maximum, standard deviation, mean, etc., of the flow packet sizes and inter-arrival times, in addition to information about Transmission Control Protocol (TCP) flags, port numbers, and

Internet Protocol (IP) addresses, etc. However, one limitation of these methods is that the features are hand-crafted and some of them are data and network dependent. Recently, Deep Learning (DL) has emerged as a new ML method that has the representation learning capability and it can be applied on raw data. Consequently, recent traffic classification methods have considered raw traffic data with DL-based classifiers [2–8]. However, the limitation of these methods is that the packet fields are dynamic and some values are connection and network dependent. Alternatively, in [9,10], we proposed a quasi-raw data representation based on basic flow packets characteristics. This method consists of extracting the size, inter-arrival time, and direction from the first  $N = n \times n$  packets of a flow.

However, deciding on the number of packets, that should be considered per flow, is not a straightforward task. Previous work consider a subset of packets or a window of time per flow for early traffic classification based on empirical classification, without optimizing the trade-off between speed of classification and accuracy. In this paper, we aim to define a stopping criteria to guarantee good accuracy and low

<sup>\*</sup> Corresponding author.

E-mail address: [oms15@mail.aub.edu](mailto:oms15@mail.aub.edu) (O. Salman).

response time. In this context, a confidence measure is proposed based on the training accuracy and the mutual information between the data and the class vector. In addition, we propose an ensemble DL-based classifier, where consecutive sets of DL-based classifiers are trained with different input dimensions. Consequently, our proposed real-time classifier is based on the combination between the proposed stopping criteria (based on confidence measure) and the ensemble classification. The implementation results show that the ensemble method enhances the accuracy of the individual classifiers and yields better classification performance at early stages of the flows. Combining the proposed stopping criteria with ensemble classification results in choosing  $n$ , which strikes a good balance between real time traffic classification, high accuracy, and less training overhead.

The rest of this paper is organized as follows. Section 2 reviews the related work of early traffic and time series classification. In Section 3, we present our proposed hierarchical classification framework including, definition of classes, data collection, and data representation. In Section 4.1, we present our proposed confidence measure. Section 4.2 presents our DL-based ensemble method. Section 5 presents the experimental results. Finally, we conclude in Section 6.

## 2. Related work

In this section, we review the related work to the two main ideas of this paper, which are: (1) the hierarchical and (2) the real-time traffic classification.

### 2.1. Hierarchical traffic classification

From the hierarchical classification perspective, in [11], Yu et al. present the first hierarchical traffic classification framework. This framework consists of three levels: at the first level, traffic is classified into P2P and non-P2P; at the second level, the P2P traffic is classified into 3 classes based on the traffic type (messenger, file sharing, and Tele-Vision (TV)), at the third level, the P2P is classified into 11 classes and non-P2P into 5 classes based on the application. Similarly, in [12], Grimaudo et al. propose a three-levels classification framework: at the first level, the traffic is classified into known and unknown, at the second level it is classified based on the protocol, and at the third level, the classification is based on the application. In [13,14], a multilateral and hierarchical traffic classification framework is presented. At the multilateral level, the classification consists of 4 classes application, protocol, function, and service. Then, for each class, a hierarchical classification is performed based on different criteria. In [15], Dong et al. propose a hierarchical approach for video traffic classification. Considering only the Hypertext Transfer Protocol Secure (HTTPS) traffic, in [16], a two-level hierarchical classification framework is presented. Recent works [17,18] have considered The Onion Router (TOR) traffic for hierarchical classification. More recently, a multi-task classifier is proposed in [19] based on DL to perform a hierarchical classification with the same network while minimizing noticeably the training time compared to the case where a classifier is trained for each classification task (level). However, in all the previous work, there is no well and uniform definition of the different levels of classification. In addition, for some works, different features are used for the different levels of classification. Moreover, there is no work reaching the device level in the classification granularity. In our work, we define 4 granularity levels for traffic classification going from 4 broad traffic types towards a very granular level identifying the device type. Moreover, we present a systematic general definition for the different classification levels and traffic was collected according to this definition.

### 2.2. Real-time traffic & time-series classification

Being critical for real-time traffic classification, early classification has been considered in previous works. Most of the previous works consider the first  $N$  packets of a flow, with no well-defined rule on how  $N$  is chosen. In [20–22], Bernaille et al. considered packet sizes of the first few packets of a flow to classify Internet traffic. The empirical results showed that the first 4 to 6 packets are sufficient to achieve high accuracy results (up to 90%). Similarly, in [23], Huang et al. extracted statistical features from the first 20 packets of a round, that is the ensemble of packets sent from one side in a talk between two parties, and a talk being the ensemble of packets sent in both directions during a time interval. In [24], Lu et al. showed that removing 72% of the packets per flow does not affect greatly the classification accuracy. An analysis of the effect of the considered number of packets on the accuracy was performed in [25]. The authors showed that adversary attacks, manifested by the injection of fake packets, can have a negative impact on the early traffic classification. Yang et al. showed in [26] that 6 packets are sufficient to achieve up to 95% accuracy for protocol-based traffic classification. In [27], Zhao et al. propose a real-time hierarchical classification based on sub-set of packets and the Error-Correcting Output Codes (ECOC) was used to enhance the classification performance based on the successive classification results of the sub-set of packets (or sub-flows). In [28], it was shown that the first 5 or 6 packets carry sufficient information for accurate classification. In [29], Peng et al. showed that 5 to 7 packets are sufficient for early traffic classification with high accuracy. In a more recent work, the authors analyzed the effect of packet sampling on early traffic classification [30]. The results showed that systematic sampling is the most efficient technique to reach a high classification accuracy. In [31], Nguyen et al. considered sub-flows to provide timely and continuous classification without being limited to the first few packets of the flow.

Having a streaming nature, traffic can be treated as time-series data. In this context, different solutions have been proposed for early time series classification, where some are based on the classification confidence. Xing et al. in [32], proposed an Early Classification on Time Series (ECTS) method, which consists of delaying the decision until the classification output became stable. A similar approach was proposed in [33], where the authors relied on an ensemble of classifiers. In this case, the early classification was performed when the ensemble decision became stable. RELCLASS is another method proposed by Parrish et al. [34], which consists of stopping when the decision did not differ from the one obtained when full time series is considered. REACT (Reliable EARly ClassificaTion) is proposed in [35]; it is based on a reliable classification method for multivariate time series using categorical and numerical attributes. TEASER, standing for Early and Accurate Time Series Classification, is proposed in [36] leveraging the fact that in real-life time series data can be generated at arbitrary times. Other methods are based on a confidence measure of the early classification [37,38], reflecting the reliability of the obtained classification at each timestamp. Thus, when a certain confidence level is attained, the corresponding output is taken as an early classification decision.

Since DL emerged as a new ML method, having the representation learning power, it has been applied in different domains. Especially, DL has been applied for time series classification, as time series are high dimensional data. In this context, new DL methods were developed like attention and reinforcement learning techniques for early time series classification [39–42]. Another family of methods is based on shapelets detection [43–45], which are local features that have high differentiation power [46]. The main limitation of these methods is that they do not consider the trade-off between early classification and accuracy. To overcome this limitation, other work tried to compromise between early classification and accuracy by modeling the situation as an optimization problem of a certain cost function. Dachraoui et al. in [47] combine clustering and cross-validation in a cost function that has to be optimized for early classification decision. Based on this

cost function, the decision is taken whenever the cost is unlikely to decrease. Similarly, Tavenard & Malinowski in [48] modeled early classification as an optimization problem, but they employed binary classification instead of clustering. In [49], Mori et al. proposed a two-step method to optimize the trade-off between early classification and accuracy. The first step consists of training different classifiers, and in a second step, a stopping rule is applied to optimize the cost function reflecting speed and the accuracy obtained in the first step based on a genetic algorithm. In a recent work [50], Mori et al. considered a multi objective optimization technique to optimize the trade-off between speed and accuracy.

As a summary, previous works consider early traffic classification based on empirical results without defining any rule to optimize the trade-off between response time and accuracy. Moreover, other existing work that consider the trade-off between earliness and accuracy were not designed for the traffic classification application neither for Deep Learning based classification. Thus, they cannot be applied directly for traffic classification. Instead, in this work, we propose a confidence measure that helps in selecting the required number of packets ( $N = n \times n$ ), which optimizes the traffic classification accuracy while minimizing the training overhead, and the testing response time. The experimental results show that relying on the proposed measure we can achieve good classification accuracy early on. However, relying on the empirical accuracy, results in considering higher number of packets for classification and consequently this results in higher latency (response time). Another contribution of this work is the proposed Deep Learning based ensemble method that builds up the classification confidence in a real-time manner.

In fact, our proposed confidence measure is based on the training accuracy of a DL-based classifier and the mutual information between the packets features and the class vector. Moreover, to enhance the classification accuracy, we propose an ensemble classification model, consisting of successive classifiers trained on different input dimensions (by increasing sequentially  $n$  by  $\delta$ ). This ensemble classifier takes advantage of the streaming nature of data to incrementally build the model confidence and to enhance the classification accuracy. The contributions of this paper can be summarized as follows:

- Proposing a new confidence measure based on classifier training accuracy and the mutual information between features and class vector.
- Defining a stopping criteria to minimize the number of packets required for accurate traffic classification based on the proposed confidence measure.
- Proposing an ensemble-based classification method by aggregating successive classification results for incremental values of  $n$  to benefit from the streaming nature of traffic and to enhance the classification accuracy of individual classifiers.
- Combining the stopping criteria with the ensemble classification to achieve high early classification accuracy.

### 3. Hierarchical classification

In this section, we briefly describe our proposed hierarchical data classification model with the definition of classes, data collection, feature extraction and data representation, and classifier model. A detailed explanation of the proposed data representation and classification model can be found in [9,51]. Table 1 lists all the notations used in this paper.

#### 3.1. Classes definition

To classify the traffic, we define a hierarchical classification model consisting of four granularity levels (See Fig. 1). At the first level, we define four categories reflecting different QoS and security requirements including *bulk data transfer*, *interactive*, *streaming*, and *transaction*.

**Table 1**  
Summary of notations.

Notation	Definition
$F_i$	$i$ th Flow
$N$	Number of packets
$n$	Dimension of the image representation, with $N = n \times n$
$ F_i $	The number of packets in a flow
$M_i = \frac{ F_i }{n \times n}$	The number of sub-flows of $F_i$
$S_i$	Size of the $i$ th packet of a flow
$T_i$	Inter-arrival time of the $i$ th packet of a flow
$D_i$	Direction of the $i$ th packet of a flow
$Acc(n)$	The training accuracy
$MI(n)$	The average mutual information between the features vector
$C(n)$	Model confidence measure
$\delta_{Acc}(n)$	Changing rate in $Acc(n)$
$\delta_{MI}(n)$	Changing rate in $MI(n)$
$\delta_C(n)$	Change rate in $C(n)$

The *bulk data transfer* consists of large size traffic generated basically from files exchanged between client and server in an asynchronous manner. For this type of traffic, high bandwidth and reliability are essential requirements. The *interactive* traffic takes place between two entities in a synchronous manner. Here, the low latency and jitter requirements are of high importance. The *streaming* traffic consists of real-time traffic requested by a client from a server. Here, time is important but this type of traffic can tolerate some latency, in contrast to the *interactive* one. The *transaction* traffic has a small size and it includes sensing, actuation, or payment data communication. This traffic needs highly secure and reliable communication. At the second level, more granular classes are defined. At this level, user actions for each level-1 class are defined. For bulk data transfer, 4 actions are considered: file download, file upload, system update, and app download. For *interactive*, 4 actions are defined: voice call, video call, gaming, and texting. For *streaming*, 2 types of actions are considered: voice and video calls. Finally, for *transaction*, 3 actions are defined: sensing, actuation, and payment. At Level-3, classes presenting the used applications are defined. For each level-2 class, a set of applications is considered. Finally, at Level-4, the classification aims at identifying the device type generating the level-3 class traffic. For most traffic, mobile phones and laptops are considered as generating devices. However, for transaction traffic, a set of IoT devices is used to generate sensing and actuation traffic [51].

#### 3.2. Data collection

For collecting traffic, the setup was implemented in a home environment. The traffic was collected using phones, laptops and IoT devices (D-Link Water Sensor, D-Link Camera, D-Link Siren, D-Link Plug, and Samsung home kit). A laptop was configured as a router connected to the Internet through the home gateway. To collect traffic, Wireshark was launched on the WLAN interface for the case of Wi-Fi enabled devices and on the Ethernet interface for the hub connected devices. One hour for each type of devices and applications was collected and saved in PCAP files. The summary of the collected flows is included in Table 2. In addition, to show the generalization of our method, we applied the proposed confidence measure on the VPN data. The description of this dataset is included in Table 3. This dataset consists of six classes: chat, file transfer, mail, streaming, torrent, and Voice over IP (VoIP).

#### 3.3. Feature extraction and data representation

The flow is defined as being the ensemble of packets having the same source IP, source port number, destination IP, and destination port number in either direction. We choose to extract 3 features per packet, packet size, inter-arrival time, and direction of packet flow. The extracted features can be represented as an RGB image (or matrix) as



Table 3

VPN dataset.	
Class	Number of flows
Chat	890
File transfer	513
Mail	364
Streaming	143
Torrent	268
Voice over IP (VoIP)	703



$$\begin{pmatrix} [s, t, d]_1 & \cdots & [s, t, d]_n \\ \vdots & \ddots & \vdots \\ [s, t, d]_{nx(n-1)} & \cdots & [s, t, d]_{nxn} \end{pmatrix}$$

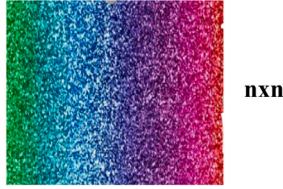


Fig. 2. Data representation.

the timestamp of the  $i$ th packet and  $D_i$  the direction of the  $i$ th packet. These values are normalized between 0 and 1; the packet size is checked if it is greater than the Maximum Transmission Unit ( $MTU = 1500$ ). If yes, it is set to the  $MTU$ . Then, all packet sizes are divided by the  $MTU$  value. The inter-arrival time is checked if it is greater than 1 s, and if so, it is set to 1. Finally, the direction is set to 0 if the packet is in the same direction of the first packet of the flow and to 1 otherwise. After dividing the flows into 80% training, and 20% testing, the flows are divided into  $M$  sub-flows of  $(n \times n)$  packets each. The obtained features vectors are of length  $(n \times n \times 3)$ .

### 3.4. Classifier model

In a previous work [9], a comparison has been conducted between different Deep Learning architectures (including Convolution Neural Network (CNN), and Recurrent Neural network (RNN)) with the considered features: packets size, inter-arrival time, and direction. The results show that CNN gives better results than RNN and LSTM. Actually, the image representation allows to capture the correlation between different features (packet inter-arrival time, direction, and size), contrarily to other networks (e.g. RNN), which are specialized to 1D input.

CNN is a specialized DL architecture that has shown promising results when applied to different types of data including images, videos, audio signals, etc. A CNN architecture consists mainly of four types of layers: convolution layer, pooling layer, dropout layer, and fully connected layer. As the name indicates, the convolution layer consists of a convolution operation between filters (or “sliding windows”) and the input data. These filters, having defined dimensions, have values corresponding to their function. There are different types of convolution filters that have different functions including: identity, edge detection, box blur, sharpen, Gaussian blur, etc. Another layer is the pooling layer, consisting also of applying filters to the input. The aim of this layer is to reduce the input dimensions. In addition, pooling can use

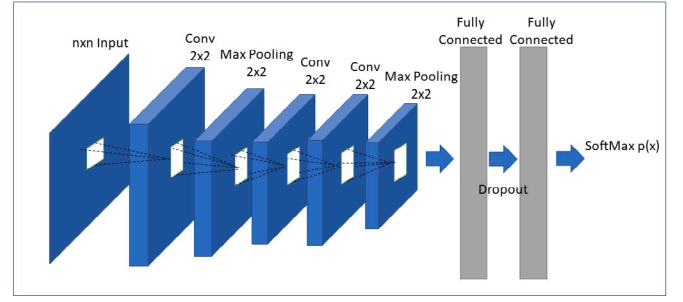


Fig. 3. The used CNN architecture.

one of different functions: maximum, average, sum, etc. To avoid overfitting, the dropout layer aims at dropping part of the input data with a certain probability. The classification architecture consists of three convolution layers, two pooling layers, two fully connected layers, and one dropout layer, as shown in Fig. 3. Moreover,  $(2 \times 2)$  convolutional filters and  $(2 \times 2)$  sub-sampling (pooling) layers are applied at stride 1 at the corresponding layers.

## 4. Proposed real-time classifier

In this section, we detailed our proposed confidence measure and DL-based ensemble classification method. In fact, at a first stage, we rely on a stopping criteria (based on the confidence measure) to define the number of packets  $N$  to be considered by the real-time classifier. Then, the ensemble classification method is used to enhance the accuracy at the chosen  $N$ , by benefiting from the streaming nature of traffic.

### 4.1. Confidence measure

Since the choice of  $n$  affects the classification accuracy, increasing  $n$  does not always guarantee better accuracy, rather it increases the training overhead. Moreover, waiting for large number of packets is not acceptable in real-time applications like intrusion detection. In this case, finding the optimal number of packets that guarantees good testing accuracy while minimizing the required number of packets is key. This applies to time series data where, for example, being able to recognize the voice from the first samples of an audio signal is required for real-time applications. While the problem can be compared to a traditional one of features selection or reduction, yet, the corresponding techniques are not effective in the case of streaming data due to the inability to have the full set of features ahead of time. Thus, the importance of features should be analyzed as the data comes in to reach a level where the considered data features are enough to guarantee good classification accuracy.

In this section, we present our proposed confidence measure based on the mutual information between the features, the class labels vector and the training accuracy. Then, this measure is used to choose an optimal value of  $n$ , meeting the compromise between the response time and the test accuracy. As mentioned in Section 3.3, the flows features are transformed into three channels: R, G, B representing the size, timestamp, and direction. As shown in Fig. 4, a flow can be represented by three matrices  $X_R$ ,  $X_G$ , and  $X_B$ , where each matrix component,  $r_i$ ,  $g_i$ , and  $b_i$  represents a pixel of the RGB image. Consequently, each component of these matrices can be represented by a random variable  $R_i$ ,  $G_i$ , and  $B_i$  for R, G, and B channels. The mutual information is the average sum of the individual mutual information between each color component and the label vector, as shown in the following equations:

$$MI_R = \frac{1}{n \times n} \times \sum_{i=0}^{i=n \times n} MI(R_i, Y);$$

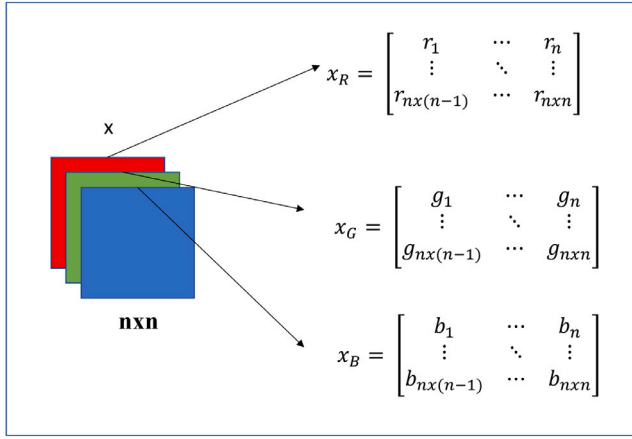


Fig. 4. Image representation.

$$MI_G = \frac{1}{n \times n} \times \sum_{i=0}^{i=n \times n} MI(G_i, Y);$$

$$MI_B = \frac{1}{n \times n} \times \sum_{i=0}^{i=n \times n} MI(B_i, Y);$$

$$MI = \frac{MI_R + MI_G + MI_B}{3}$$

To find the best value of  $n$  meeting the compromise between the accuracy and the classification performance, we define a new confidence measure, that is based on the training accuracy and the mutual information between the data and the label vector. The rate of change of this confidence measure is expressed in terms of the rate of change in the accuracy and the mutual information, as follows:

$$\delta_{Acc}(n) = \frac{Acc(n) - Acc(n-1)}{Acc(n-1)};$$

$$\delta_{MI}(n) = \frac{MI(n) - MI(n-1)}{MI(n-1)};$$

$$\delta_C(n) = \frac{\delta_{Acc}(n) + \delta_{MI}(n)}{2};$$

where  $Acc(n)$  is the training accuracy and  $MI(n)$  is the average mutual information between the features vector of the  $(n \times n)$  packets features and the class labels vector:

$$MI(n) = \frac{\sum_{i=0}^{i=n \times n} MI(x_i, Y)}{n \times n} \quad (2)$$

Consequently, the confidence  $C(n)$  can be expressed in terms of  $\delta_C(n)$  as follows:

$$C(n) = C(n-1) \times (1 + \delta_C(n)) \quad (3)$$

It should be noted here that a necessary condition should be met to have  $0 < C(n) < 1$ . This condition is  $\delta_C(n) < \frac{1-C(n-1)}{C(n-1)}$ . In our case, we check if this condition is met, and if not,  $\delta_C(n)$  is set to  $\frac{1-C(n-1)}{C(n-1)}$ . In fact, this bound is found based on the fact that  $0 < C(n) < 1$ , consequently,  $-C(n-1) < C(n) - C(n-1) < 1 - C(n-1)$ . Thus, having  $0 < C(n-1) < 1$ , we can divide all the inequality's sides by it. As a result, we get  $-1 < \delta_C(n) < \frac{1-C(n-1)}{C(n-1)}$ .  $MI(n)$  should be calculated in the training phase and not in the testing one. After defining the optimal value of  $n$ , based on the proposed confidence measure, at the testing phase, we can directly use the optimal  $n$  to classify the traffic. Moreover, the proposed ensemble method allows the model to give early classification results at earlier time (i.e. for  $n < n_{optimal}$ ).

Algorithm 1 describes the optimization for the choice of  $n$ . First, we configure the bounds of  $n$ :  $n_{min}$  and  $n_{max}$ .  $n_{max}$  is chosen based on the

#### Algorithm 1 Selection of $n$ based on confidence measure

```

1: procedure OPTIMIZE  $n(Acc, MI)$ 
2:    $n = n_{min}$ 
3:    $C(n) \leftarrow (Acc(n) + MI(n))/2$ 
4:   while  $n < n_{max}$  do
5:      $\delta_{Acc}(n) \leftarrow \frac{Acc(n) - Acc(n-1)}{Acc(n) + Acc(n-1)}$ 
6:      $\delta_{MI}(n) \leftarrow \frac{MI(n) - MI(n-1)}{MI(n) + MI(n-1)}$ 
7:      $C(n) \leftarrow C(n-1) \times (1 + \frac{\delta_{Acc}(n) + \delta_{MI}(n)}{2})$ 
8:      $n = n + \delta$ 
9:    $max_C = C(n)$ 
10:   $n_{optimal} = 0$ 
11:  for  $n \in range(n_{min}, n_{max})$  do
12:    if  $C(n) > max_C$  then
13:       $n_{optimal} = n$ 
14:       $max_C = C(n)$ 
15:  return  $n_{optimal}$ 

```

application and the training data size. To get the training accuracy, we train  $n_{max} - n_{min}$  models by successively increasing  $n$  by  $\delta$ . The training accuracy is computed based on the validation data. Then, after computing the training accuracy, the mutual information at each value of  $n$  is computed between each packet feature and the label vector. The average sum is computed and the confidence measure is calculated as stated above. Finally, the optimal value of  $n$  ( $n_{optimal}$ ) is the one yielding the maximum confidence.

#### 4.2. Ensemble classifier model

In this section, we present our ensemble classifier model, which aims at enhancing the classification accuracy at a certain value of  $n$  by considering the preceding classifiers results. This model considers a streaming data type, in our case, the successive packets of a flow. At each value of  $n$ , a model is trained with input having dimensions of  $(n \times n \times 3)$ . The classification results of the preceding classifiers are aggregated and the average is used to get the ensemble decision.

As shown in Fig. 5, our model consists of successive classifiers of different dimensions. For each value of  $n$ , a classifier  $C_i$ , where  $i = n_{min}, \dots, n_{max}$ , is trained on data with dimension  $(n \times n)$ . At each  $n$ , the classifications at previous values of  $n$  are considered. The successive classifiers form an ensemble model, where the classifiers predictions are averaged to get the final decision. In the algorithm below, we describe how these classifiers predictions are aggregated and how the accuracy of the ensemble classifier is computed. It should be noted, as shown in Algorithm 2, that the probabilistic output of the classifiers is considered, and not only the class label. Consequently, the aggregated results include also the classification confidence, as follows:

$$Y_i = index_{max} \left( \frac{1}{n_{max} - n_{min}} \left[ \sum_{i=n_{min}}^{i=n_{max}} P_{i1}, \dots, \sum_{i=n_{min}}^{i=n_{max}} P_{im} \right] \right) \quad (4)$$

where  $Y_i$  is the output label of the flow  $F_i$ ,  $index_{max}$  is a function that returns the index of the maximum element of a vector, and  $P_{ij}$  is the output probability of the classifier  $i$  for class  $j$ , with  $j = 1, \dots, m$ .

#### 4.3. Accuracy vs. Speed trade-off

Based on the proposed confidence measure and ensemble method, our proposed real-time classifier could benefit from the streaming nature of traffic to strike a good balance between high accuracy and low response time. The confidence measure helps in choosing the optimal  $n$  based on which the system can guarantee good accuracy at an early

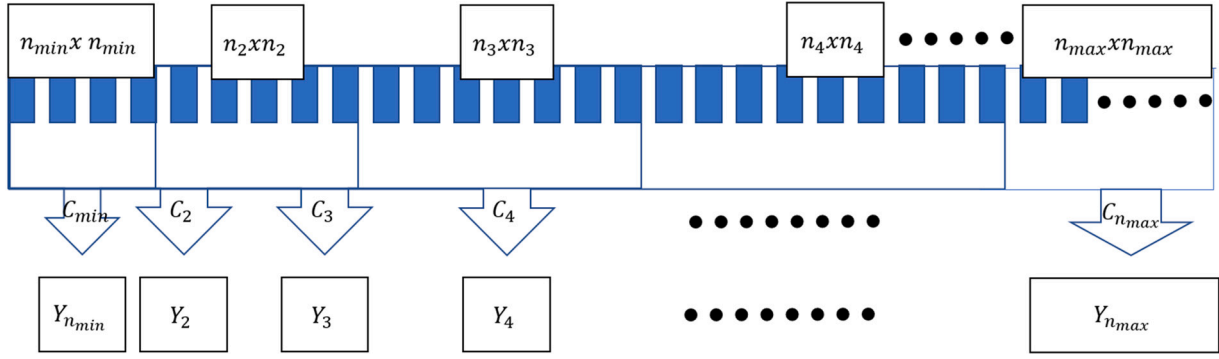


Fig. 5. Ensemble classifier model.

**Algorithm 2** Ensemble classification

---

```

1: procedure OPTIMIZE ACCURACY( $n$ )
2:   for  $item \in X_{test}$  do
3:     while  $i < n$  do
4:        $load\ model_i$ 
5:        $P_{item}(i) = model_i.predict(item)$ 
6:        $i = i + 1$ 
7:        $P_{item} \leftarrow \frac{\sum_{i=0}^{i=n} P_{item}(i)}{n}$ 

```

---

flow phase. Moreover, the system could benefit from the classification at previous values of  $n$  to enhance the accuracy at the optimal  $n$ , applying the ensemble classification concept. As such, the proposed real-time classifier consists of the combination of ensemble classification with the stopping criteria based on the proposed confidence measure.

## 5. Experimental results

In this section, we include the results of the proposed real-time classification method, considering our collected dataset, as summarized in Table 2. In addition, we consider a publicly available dataset (VPN-nonVPN2016 dataset) [52] for validating our proposed method. For each dataset, the flows are extracted from the PCAP files using the dpkt Python library [53]. To implement the classifiers, TFlearn [54] is used, which is a high level DL python library based on Tensorflow [55]. In all classification tests, cross validation is applied with 4 folds. To optimize the loss function of the CNN architecture, the ADAM optimizer is used and Rectifier Linear Unit (ReLU) is applied for activation at different layers. The dropout probability is chosen to be 0.5. The cross entropy is chosen to be the output function. The learning rate is 0.001, the weights are initialized by the truncated normal distribution, and the biases are initialized to 0.

To evaluate the effectiveness of the proposed real-time classification method, we start by calculating the confidence measure using our data. The results are included in 5.1. Then, we compare the average sum to other ensemble techniques. The comparison results, included in 5.2, show that the average sum is the ensemble technique presenting the best accuracy results. Adopting the average sum, in 5.3, we include the enhancements in terms of the number of packets and flow time when applying the proposed real-time classification method.

### 5.1. Confidence measure results

At this phase, we start by getting the training accuracy of the models at different values of  $n$ . To this end, we train  $n_{max} - n_{min}$  models by successively increasing  $n$  by  $\delta$ . In fact,  $n_{max}$  and  $n_{min}$  are chosen based on the application requirements and the dataset. For example, for classifying traffic flows, 4 packets is commonly used as the minimum

number for flow based classification and 784 packets is the average maximum flow length.

For our dataset, we choose  $n_{max} = 28$  and  $n_{min} = 2$  with  $\delta = 1$ . The training accuracy is obtained by testing the trained model on the validation data, being 20% of the training one. In addition, the mutual information at each value of  $n$  is computed between each packet feature and the label vector. Then, the confidence measure is computed based on the Eqs. (1), (2), and (3).

Examples of the results, including the training accuracy, the mutual information, and the computed confidence measure, are shown in Fig. 6: (a) for Level-1, (b) for Level-2, (c) for Level-3, (d) for Level-4, and (e) for VPN. These figures clearly indicate that one can choose the value of  $n$  corresponding to the maximum confidence value to achieve good classification performance at an early stage. It can be noticed here that the results reach high testing accuracy, which could be explained by the fact that the classifier is well trained and the features are well chosen to differentiate between the target classes. However, to avoid any overfitting, regularization and early stopping techniques could be applied. However, in our case, the accuracy results have relative notions, and thus applying any of these techniques will not modify our confidence measure results or the choice of  $n$ .

### 5.2. Ensemble classifier results

For assessing the proposed ensemble classifier, we compare different ensemble techniques to compute the final decision including: average sum, weighted sum, and majority voting. In addition, we compare the performance of the ensemble method compared to the individual classifiers performance. For the average method, we compute the final decision based on Algorithm 2. For the weighted sum, the probabilistic output of each classifier is multiplied by its confidence measure before computing the average over the sum of weights (confidence measures of the successive classifiers). However, for the majority voting, the final decision is computed based on the maximum number of votes for the output class.

In Table 4, we show the average results for the different levels and for the VPN dataset. It is clear from the results that the average sum gives the best results, except for the VPN dataset where the majority voting method gives slightly higher accuracy. Moreover, it can be seen that the ensemble-based accuracy is greater than the accuracy obtained by the individual classifiers. At level-1, the average ensemble accuracy is 89.77% and for the individual classifiers, it is 87.34%. At level-2, on average, the ensemble methods presents an accuracy of 84.13%, and the individual classifiers present an accuracy of 81.78%. At level-3, 71.60% accuracy is achieved by the average ensemble methods, while 68.51% by the individual classifiers. At level-4, 84.87% is attained by the average ensemble classifier accuracy and 83.36% by the individual classifiers. For the VPN dataset, the ensemble method presents also a 1% enhancement compared to the individual classifiers.

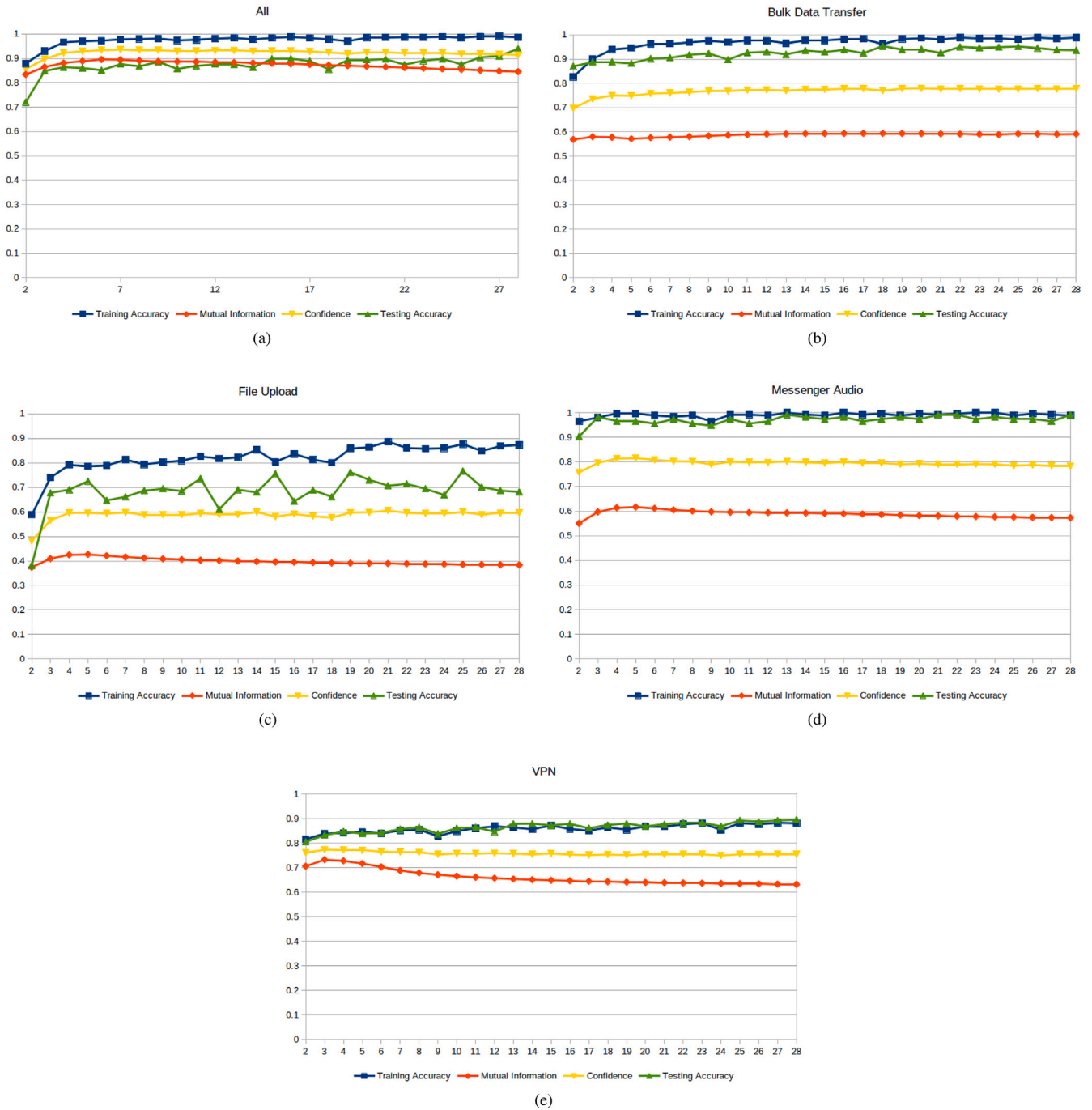


Fig. 6. Confidence measure results in function of  $n$ .

It should be noted here that the results include the average accuracy for  $n = 2, \dots, 28$ , for our data-set with a step of 1 for the ensemble and individual classifiers. Thus, 26 individual classifiers are trained on our dataset.

### 5.3. Real-time classification results

In this section, we present the results obtained by applying the proposed real-time classification method based on the optimal choice of  $n$  (obtained by applying the proposed confidence measure), and the ensemble classifier. The results show the efficiency of the proposed confidence measure along with the ensemble classifier in the case of time series data for striking a good balance between earliness and accuracy.

In Table 5, we include the results of  $n$  and the corresponding accuracy results when relying on the mutual information, confidence or the training accuracy and the corresponding maximum accuracy attained in the testing phase. In this table, the column  $N$  includes the number of packets at which the mutual information, confidence, training accuracy, or testing accuracy is maximum. The column “Individual Accuracy” includes the accuracy at the corresponding  $N$  and the “Ensemble Accuracy” is the obtained accuracy when applying the ensemble method. It is clear from the results that:

- Relying on the maximum mutual information to find the optimal  $n$  yields a low average in the number of packets (36 at level-1, 127 at level-2, 35 at level-3, 164 at level-4, and 9 for the VPN data)

**Table 4**  
Ensemble accuracy results.

Data	Average	Weighted average	Majority voting	Individual
Level-1	89.77%	86.81%	89.76%	87.34%
Level-2	84.13%	82.43%	83.95%	81.78%
Level-3	71.60%	70.12%	70.64%	68.51%
Level-4	84.87%	84.67%	84.65%	83.36%
VPN	76.15%	76.15%	76.9%	75.27%

**Table 5**  
Best choice of  $n$ .

Mutual information			Training			Confidence			Testing		
$N$	Individual accuracy (%)	Ensemble accuracy (%)	$N$	Individual accuracy (%)	Ensemble accuracy (%)	$N$	Individual accuracy (%)	Ensemble accuracy (%)	$N$	Individual accuracy (%)	Ensemble accuracy (%)
Level-1											
36	87.94	90.09	729	91.55	91.55	49	88.11	90.26	729	91.55	91.55
Level-2											
127	84.70	84.80	402	81.59	84.93	149	86.19	85.46	351	89.46	84.83
Level-3											
35	66.95	69.72	540	70.84	74.13	286	71.16	72.26	391	79.46	74.41
Level-4											
164	82.34	83.01	229	84.93	85.52	196	84.43	84.54	251	91.31	86.83
VPN data											
9	69.87	72.67	625	79.84	82.62	36	74.16	75.08	729	82.34	82.71

and a good accuracy of (87.94 at level-1, 84.70 at level-2, 66.95 at level-3, 82.34% at level-4, and 69.87% for the VPN data).

- However, relying on the maximum training accuracy to find the optimal  $n$  yields a high number of packets (729 at level-1, 402 at level-2, 540 at level-3, 229 at level-4, and 625 for the VPN data) and a better accuracy of (91.55% at level-1, 81.59% at level-2, 70.84% at level-3, 84.93% at level-4, and 79.84% for the VPN data).
- Having the confidence as a combination of these two measures, relying on the maximum confidence for finding the optimal  $n$ , the number of packets is higher than that of the mutual information (49 at level-1, 149 at level-2, 286 at level-3, 196 at level-4, and 36 for the VPN data) but it shows a higher accuracy of (88.11% at level-1, 86.19% at level-2, 71.16% at level-3, 84.43% at level-4, and 74.16% for the VPN data).

It can be noticed also that applying the ensemble classification, the accuracy presented in the column “Ensemble Accuracy” shows an enhancement to the accuracy obtained with individual classifiers presented in the column “Individual Accuracy”. Also note that ( $N = n \times n$ ) represents the number of packets for the case of network traffic.

To evaluate the enhancement and/or degradation in terms of the number of packets and classification accuracy, we computed the *difference* between the number of packets (samples)  $N$  and the accuracy, when choosing  $n$  corresponding to the maximum mutual information, training accuracy, or our proposed confidence measure, and the ones obtained for the optimal case during testing phase. The results are included in Table 6. The  $N$  column includes the *difference* between the optimal  $N$  when the mutual information, training accuracy, or confidence are maximum and the one obtained when the testing accuracy is maximum. The “Individual Accuracy” column includes the *difference* between the corresponding accuracy and the maximum testing accuracy. The “Ensemble Accuracy” includes the *difference* in terms of accuracy, when the ensemble method is applied while considering the individual maximum testing accuracy. The “Testing Ensemble Accuracy” includes the *difference* in accuracy when the ensemble method is applied to get the maximum testing accuracy.

It can be noticed that on average, our proposed confidence measure, if used to choose the optimal  $n$ , exhibits a better enhancement in the number of packets than the training accuracy and a lower degradation in the accuracy than the mutual information method (as shown in Table 6):

- The confidence measure method presents an enhancement of 680 in the number of packets with a degradation of 3.44% in the classification accuracy at Level-1 and a degradation of 1.29% in the accuracy with the ensemble method.
- At Level-2, it shows an enhancement of 202 in the number of packets with a degradation of 3.28% in the individual classification accuracy and an enhancement of 0.63% with the ensemble method.
- At Level-3, it achieves an enhancement of 105 in the number of packets with a degradation of 8.29% in the classification accuracy, which decreases to 2.15% with the ensemble method.
- At Level-4, it offers an enhancement of 54 in the number of packets with accuracy degradation of 6.88%, which decreases to 2.29% with the ensemble method.
- For the VPN data, it presents an enhancement of 693 in the number of packets with a degradation of 8% in the classification accuracy, which decreases to 7% with the ensemble method.

To evaluate the enhancement and/or degradation in terms of time, when choosing  $n$  relying on the mutual information, training accuracy, or our proposed confidence measure, we computed the *difference* between the flow time for the obtained  $n$  and the one obtained for the best  $n$  during the testing phase, as shown in Table 7. It can be noticed that on average, our proposed confidence measure, if used to choose the optimal  $n$ , provides a better enhancement in the required flow time than the training accuracy. The confidence measure method presents an enhancement of 5.94 s in the flow time at Level-1, 2.78 s at Level-2, 3.09 s at Level-3, and 0.96 s at Level-4.

## 6. Conclusion

In this paper, we proposed a confidence measure that helps in selecting a value of  $n$ , for the required number of packets, which

**Table 6**Effect of choosing optimal  $n$  based on the mutual information measure, the training accuracy, and the proposed confidence measure.

Mutual information enhancement				Training accuracy enhancement				Confidence enhancement			
$N$	Individual accuracy (%)	Ensemble accuracy (%)	Testing Ensemble accuracy (%)	$N$	Individual accuracy (%)	Ensemble accuracy (%)	Testing Ensemble accuracy (%)	$N$	Individual accuracy (%)	Ensemble accuracy (%)	Testing ensemble accuracy (%)
Level-1											
-693	-3.61	-1.46	-1.46	0	0.00	0.00	0.00	-680	-3.44	-1.29	-1.29
Level-2											
-224	-4.76	-4.67	-0.03	51	-7.87	-4.53	0.10	-202	-3.28	-4.00	0.63
Level-3											
-356	-12.50	-9.74	-4.69	149	-8.62	-5.33	-0.28	-105	-8.29	-7.20	-2.15
Level-4											
-8	-8.98	-8.31	-3.82	-21	-6.38	-5.79	-1.31	-54	-6.88	-6.77	-2.29
VPN data											
-720	-12	-9	-10	-104	-2	0.2	-0.08	-693	-8	-7	-7

**Table 7**

Enhancement in terms of time (in seconds)

Data	Time (in seconds)				Enhancement (in seconds)		
	Mutual information	Training accuracy	Confidence	Testing accuracy	Mutual information	Training accuracy	Confidence
Level-1	0.64	6.76	0.82	6.76	-6.12	0.00	-5.94
Level-2	1.72	5.44	1.88	4.66	-2.94	0.79	-2.78
Level-3	0.76	7.46	3.13	6.22	-5.46	1.24	-3.09
Level-4	2.11	2.86	2.59	3.55	-1.44	-0.69	-0.96
VPN	0.41	6.48	1.13	7.21	-6.79	-0.72	-6.07

optimizes the traffic classification accuracy while minimizing the training overhead, and the testing response time. To do so, we relied on the mutual information between the packets features, the class vector and the training accuracy. In addition, we proposed an ensemble classification method that considers the average of the classification results given by the set of successive classifiers of different dimensions (i.e. successive values of  $n$ ). The experimentation results showed that the proposed ensemble method provides an enhancement in terms of accuracy compared to the individual classifiers. Combining the confidence measure stopping criteria and the ensemble classification, we were able to optimize the trade-off between the earliness and the classification accuracy.

This work paves the way for a new direction in the Deep Learning based traffic classification area, where the classification can be performed in a real-time manner. Several aspects still need to be further researched such as the real-time data capture and processing and its impact on the classification. Beyond networking, the proposed method can be applied for various time-series based classification problems, including voice recognition and video classification, where time is a critical performance factor.

#### CRedit authorship contribution statement

**Ola Salman:** Conceptualization, Methodology, Software, Validation, Writing – original draft. **Imad H. Elhadj:** Supervision, Writing – review & editing. **Ali Chehab:** Writing – review & editing. **Ayman Kayssi:** Writing – review & editing.

#### 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.

#### Acknowledgments

This paper is funded by the Maroun Semaan Faculty of Engineering and Architecture at the American University of Beirut, the National Council for Scientific Research (NCSR), Lebanon, and TELUS corp, Canada.

#### References

- [1] Cisco annual internet report - cisco annual internet report (2018–2023) white paper - cisco, 2020, <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>. (Accessed 3 December 2020).
- [2] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, M. Zhu, HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection, *IEEE Access* 6 (2018) 1792–1806.
- [3] Z. Li, Z. Qin, K. Huang, X. Yang, S. Ye, Intrusion detection using convolutional neural networks for representation learning, in: *International Conference on Neural Information Processing*, Springer, 2017, pp. 858–866.
- [4] M. Lotfollahi, M.J. Siavoshani, R.S.H. Zade, M. Saberian, Deep packet: A novel approach for encrypted traffic classification using deep learning, *Soft Comput.* (2017) 1–14.
- [5] H. Huang, H. Deng, J. Chen, L. Han, W. Wang, Automatic multi-task learning system for abnormal network traffic detection., *Int. J. Emerg. Technol. Learn.* 13 (4) (2018).
- [6] Z. Wang, The applications of deep learning on traffic identification, in: *BlackHat USA*, vol. 24, 2015.
- [7] W. Wang, M. Zhu, J. Wang, X. Zeng, Z. Yang, End-to-end encrypted traffic classification with one-dimensional convolution neural networks, in: *2017 IEEE International Conference on Intelligence and Security Informatics, ISI, 2017*, pp. 43–48, <http://dx.doi.org/10.1109/ISI.2017.8004872>.
- [8] W. Wang, M. Zhu, X. Zeng, X. Ye, Y. Sheng, Malware traffic classification using convolutional neural network for representation learning, in: *2017 International Conference on Information Networking, ICOIN, IEEE, 2017*, pp. 712–717.
- [9] O. Salman, I.H. Elhadj, A. Chehab, A. Kayssi, A multi-level internet traffic classifier using deep learning, in: *2018 9th International Conference on the Network of the Future, NOF, IEEE, 2018*, pp. 68–75.

- [10] O. Salman, I.H. Elhajj, A. Kayssi, A. Chehab, Data representation for CNN based internet traffic classification: A comparative study, *Multimedia Tools Appl.* 80 (11) (2021) 16951–16977.
- [11] J.-H. Yu, H.-S. Lee, Y.-H. Im, M.-S. Kim, D.-H. Park, Real-time classification of internet application traffic using a hierarchical multi-class SVM, *KSII Trans. Internet Inf. Syst.* 4 (5) (2010) 859–876.
- [12] L. Grimaudo, M. Mellia, E. Baralis, Hierarchical learning for fine grained internet traffic classification, in: 2012 8th International Wireless Communications and Mobile Computing Conference, IWCMC, IEEE, 2012, pp. 463–468.
- [13] J.-h. Kim, S.-H. Yoon, M.-S. Kim, Study on traffic classification taxonomy for multilateral and hierarchical traffic classification, in: 2012 14th Asia-Pacific Network Operations and Management Symposium, APNOMS, IEEE, 2012, pp. 1–4.
- [14] S.-H. Yoon, K.-S. Shim, S.-K. Lee, M.-S. Kim, Framework for multi-level application traffic identification, in: 2015 17th Asia-Pacific Network Operations and Management Symposium, APNOMS, IEEE, 2015, pp. 424–427.
- [15] Y.-n. Dong, J.-j. Zhao, J. Jin, Novel feature selection and classification of internet video traffic based on a hierarchical scheme, *Comput. Netw.* 119 (2017) 102–111.
- [16] W.M. Shbair, T. Cholez, J. Francois, I. Chrisment, A multi-level framework to identify HTTPS services, in: NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium, 2016, pp. 240–248, <http://dx.doi.org/10.1109/NOMS.2016.7502818>.
- [17] A. Montieri, D. Ciunzo, G. Bovenzi, V. Persico, A. Pescapé, A dive into the dark web: Hierarchical traffic classification of anonymity tools, *IEEE Trans. Netw. Sci. Eng.* (2019).
- [18] J. Lingyu, L. Yang, W. Bailing, L. Hongxi, X. Guodong, A hierarchical classification approach for tor anonymous traffic, in: 2017 IEEE 9th International Conference on Communication Software and Networks, ICCSN, IEEE, 2017, pp. 239–243.
- [19] O. Barut, Y. Luo, T. Zhang, W. Li, P. Li, Multi-task hierarchical learning based network traffic analytics, in: ICC 2021 - IEEE International Conference on Communications, 2021, pp. 1–6, <http://dx.doi.org/10.1109/ICC42927.2021.9500546>.
- [20] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, K. Salamatian, Traffic classification on the fly, *ACM SIGCOMM Comput. Commun. Rev.* 36 (2) (2006) 23–26.
- [21] L. Bernaille, R. Teixeira, K. Salamatian, Early application identification, in: Proceedings of the 2006 ACM CoNEXT Conference, 2006, pp. 1–12.
- [22] L. Bernaille, R. Teixeira, Early recognition of encrypted applications, in: International Conference on Passive and Active Network Measurement, Springer, 2007, pp. 165–175.
- [23] N.-F. Huang, G.-Y. Jai, H.-C. Chao, Early identifying application traffic with application characteristics, in: 2008 IEEE International Conference on Communications, IEEE, 2008, pp. 5788–5792.
- [24] C.-N. Lu, C.-Y. Huang, Y.-D. Lin, Y.-C. Lai, Session level flow classification by packet size distribution and session grouping, *Comput. Netw.* 56 (1) (2012) 260–272.
- [25] B. Qu, Z. Zhang, L. Guo, D. Meng, On accuracy of early traffic classification, in: 2012 IEEE Seventh International Conference on Networking, Architecture, and Storage, IEEE, 2012, pp. 348–354.
- [26] B. Yang, G. Hou, L. Ruan, Y. Xue, J. Li, Smiler: Towards practical online traffic classification, in: 2011 ACM/IEEE Seventh Symposium on Architectures for Networking and Communications Systems, IEEE, 2011, pp. 178–188.
- [27] Y. Zhao, X. Xie, M. Jiang, Hierarchical real-time network traffic classification based on ECOC, *TELKOMNIKA Indones. J. Electr. Eng.* 12 (2) (2014) 1551–1560.
- [28] W. Linlin, L. Peng, M. Su, B. Yang, X. Zhou, On the impact of packet inter arrival time for early stage traffic identification, in: 2016 IEEE International Conference on Internet of Things, IThings and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE, 2016, pp. 510–515.
- [29] L. Peng, B. Yang, Y. Chen, Effective packet number for early stage internet traffic identification, *Neurocomputing* 156 (2015) 252–267.
- [30] L. Peng, H. Zhang, B. Yang, M. Su, Y. Chen, On the effectiveness of packet sampling for early stage traffic identification, in: 2016 IEEE International Conference on Internet of Things, IThings and IEEE Green Computing and Communications, GreenCom and IEEE Cyber, Physical and Social Computing, CPSCom. and IEEE Smart Data, SmartData, IEEE, 2016, pp. 468–473.
- [31] T.T. Nguyen, G. Armitage, P. Branch, S. Zander, Timely and continuous machine-learning-based classification for interactive IP traffic, *IEEE/ACM Trans. Netw.* 20 (6) (2012) 1880–1894.
- [32] Z. Xing, J. Pei, S.Y. Philip, Early classification on time series, *Knowl. Inf. Syst.* 31 (1) (2012) 105–127.
- [33] N. Hatami, C. Chira, Classifiers with a reject option for early time-series classification, in: 2013 IEEE Symposium on Computational Intelligence and Ensemble Learning, CIEL, IEEE, 2013, pp. 9–16.
- [34] N. Parrish, H.S. Anderson, M.R. Gupta, D.Y. Hsiao, Classifying with confidence from incomplete information, *J. Mach. Learn. Res.* 14 (1) (2013) 3561–3589.
- [35] Y.-F. Lin, H.-H. Chen, V.S. Tseng, J. Pei, Reliable early classification on multivariate time series with numerical and categorical attributes, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2015, pp. 199–211.
- [36] P. Schäfer, U. Leser, TEASER: Early and accurate time series classification, 2019, arXiv preprint arXiv:1908.03405.
- [37] J. Lv, X. Hu, L. Li, P. Li, An effective confidence-based early classification of time series, *IEEE Access* 7 (2019) 96113–96124.
- [38] G. He, W. Zhao, X. Xia, Confidence-based early classification of multivariate time series with multiple interpretable rules, *Pattern Anal. Appl.* (2019) 1–14.
- [39] W. Wang, C. Chen, W. Wang, P. Rai, L. Carin, Earliness-aware deep convolutional networks for early time series classification, 2016, arXiv preprint arXiv:1611.04578.
- [40] H.-S. Huang, C.-L. Liu, V.S. Tseng, Multivariate time series early classification using multi-domain deep neural network, in: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics, DSAA, IEEE, 2018, pp. 90–98.
- [41] E.-Y. Hsu, C.-L. Liu, V.S. Tseng, Multivariate time series early classification with interpretability using deep learning and attention mechanism, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2019, pp. 541–553.
- [42] C. Martinez, E. Ramasso, G. Perrin, M. Rombaut, Adaptive early classification of temporal sequences using deep reinforcement learning, *Knowl.-Based Syst.* 190 (2020) 105290.
- [43] Z. Xing, J. Pei, P.S. Yu, K. Wang, Extracting interpretable features for early classification on time series, in: Proceedings of the 2011 SIAM International Conference on Data Mining, SIAM, 2011, pp. 247–258.
- [44] G. He, Y. Duan, R. Peng, X. Jing, T. Qian, L. Wang, Early classification on multivariate time series, *Neurocomputing* 149 (2015) 777–787.
- [45] J. Grabocka, M. Wistuba, L. Schmidt-Thieme, Fast classification of univariate and multivariate time series through shapelet discovery, *Knowl. Inf. Syst.* 49 (2) (2016) 429–454.
- [46] L. Ye, E. Keogh, Time shapelets: A new primitive for data mining, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, pp. 947–956.
- [47] A. Dachraoui, Cost-Sensitive Early Classification of Time Series (Ph.D. dissertation), 2017.
- [48] R. Tavenard, S. Malinowski, Cost-aware early classification of time series, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2016, pp. 632–647.
- [49] U. Mori, A. Mendiburu, S. Dasgupta, J.A. Lozano, Early classification of time series by simultaneously optimizing the accuracy and earliness, *IEEE Trans. Neural Netw. Learn. Syst.* 29 (10) (2017) 4569–4578.
- [50] U. Mori, A. Mendiburu, I.M. Miranda, J.A. Lozano, Early classification of time series using multi-objective optimization techniques, *Inform. Sci.* 492 (2019) 204–218.
- [51] O. Salman, I.H. Elhajj, A. Chehab, A. Kayssi, A machine learning based framework for IoT device identification and abnormal traffic detection, *Trans. Emerg. Telecommun. Technol.* (2019) e3743.
- [52] G. Draper-Gil, A.H. Lashkari, M.S.I. Mamun, A.A. Ghorbani, Characterization of encrypted and vpn traffic using time-related, in: Proceedings of the 2nd International Conference on Information Systems Security and Privacy, ICISPP, 2016, pp. 407–414.
- [53] dpkt, 2019, <https://dpkt.readthedocs.io/en/latest/#>. (Accessed 2019).
- [54] TFLearn — TensorFlow Deep Learning Library, <http://tflearn.org/>.
- [55] TensorFlow, <https://www.tensorflow.org/>.



**Ola Salman** received her M.E. degree in Computer and Communications Engineering from the Lebanese University in 2013. In September 2014, she joined the Ph.D. accelerated track program in the Electrical and Computer Engineering (ECE) department at the American University of Beirut (AUB). Her research interests are in the area of Information Security and Networks, Software Defined Networks, Edge Computing, Artificial Intelligence, and Internet of things. In 2017, she received the CNRS-L/AUB doctoral scholarship award from the Lebanese National Council for Scientific Research (CNRS) in recognition of her research work.



**Imad H. Elhadj** received his Bachelor of Engineering in Computer and Communications Engineering, with distinction, from the American University of Beirut in 1997 and the M.S. and Ph.D. degrees in Electrical Engineering from Michigan State University in 1999 and 2002, respectively. He is currently an Associate Professor with the Department of ECE at AUB. Imad received Best Research Paper Award at the Third International Conference on Cognitive and Behavioral Psychology (CBP), Best Paper award at the IEEE Electro Information Technology Conference in June 2003, and at the International Conference on Information Society in the 21st Century in November 2000. Dr. Elhadj is recipient of the Teaching Excellence Award at the American University of Beirut, June 2011, the Kamal Salibi Academic Freedom Award, 2014, and the most Outstanding Graduate Student Award from the ECE Department at Michigan State University, April 2001.



**Ali M. Chehab** received his Bachelor degree in EE from AUB in 1987, the Master's degree in EE from Syracuse University in 1989, and the Ph.D. degree in ECE from the University of North Carolina at Charlotte, in 2002. From 1989 to 1998, he was a lecturer in the ECE Department at AUB. He rejoined the ECE Department at AUB as an Assistant Professor in 2002 and became a Full Professor in 2014. He received the AUB Teaching Excellence Award in 2007. He teaches courses in Programming, Electronics, Digital Systems Design, Computer Organization, Cryptography, and

Digital Systems Testing. His research interests include: Wireless Communications Security, Cloud Computing Security, Multimedia Security, Trust in Distributed Computing, Low Energy VLSI Design, and VLSI Testing. He has more than 180 publications. He is a senior member of IEEE and a senior member of ACM.



**Ayman Kayssi** studied electrical engineering and received the BE degree, with distinction, in 1987 from the American University of Beirut (AUB), and the MSE and Ph.D. degrees from the University of Michigan, Ann Arbor, in 1989 and 1993, respectively. In 1993, he joined the Department of Electrical and Computer Engineering (ECE) at AUB, where he is currently a full professor. From 2004 to 2007, he served as chairman of the ECE Department at AUB. He teaches courses in electronics and in networking and has received AUB's Teaching Excellence Award in 2003. His research interests are in information security and networks, and in integrated circuit design and test. He has published more than 200 articles in the areas of security, networking, and VLSI. He is a senior member of IEEE, and a member of ACM, ISOC, and the Beirut OEA.