Towards Identifying Internet Applications Using Early Classification of Traffic Flow

Ashish Gupta, Hari Prabhat Gupta, and Tanima Dutta
Department of Computer Science and Engineering
Indian Institute of Technology (BHU) Varanasi, India
ashishg.rs.cse16@iitbhu.ac.in, hprabhatgupta@gmail.com, dutta.tanima@gmail.com

***Abstract*—Network traffic classification has been an interesting topic of research for many years. It plays a crucial role in many network applications including resource allocation, intrusion detection, and quality of service. Network traffic is essentially a sequence or flow of time-stamped packets that are exchanged between two devices. The traffic flow also contains payload data along with information about packet statistics such as size, inter-arrival time, and direction. As these statistics are obtained from the time-stamped packets, they form a Multivariate Time Series (MTS). Such an MTS needs to be classified as early as possible to identify an Internet application associated with the generated traffic flow. In this paper, we propose an Early traffic Flow Classification (EFC) approach for identifying Internet applications using MTS. The approach estimates application-wise minimum required packets from the training data by employing k-means clustering and Long Short Term Memory model. We also develop a class forwarding method to utilize correlation that exists among different packet statistics. Additionally, we collect a real-world traffic flow dataset to evaluate the effectiveness of the approach. Experimental results show that EFC approach requires only the first 15 packets of the flow to achieve an accuracy of more than 90%.**

***Index Terms*—Early classification, Internet applications, multivariate time series, traffic flow.**

I. Introduction

Network traffic classification is one of the essential task for resource allocation, intrusion detection, Quality of Service (QoS) provisioning, and so on [1], [2]. It plays an important role in identifying the network problems and thus helps the Internet service providers to react quickly according to the occurred problem. Literature indicates that several approaches have been developed aiming to classify the network traffic into categories such as normal or malicious [3], [4], type of botnet [5], name of application (*e.g.,* Youtube, Gmail, Skype, *etc.*) [6]–[8], and so on [9], [10]. Network traffic refers to a flow of data packets that are moving across the connected devices. Such a flow contains sufficient amount of information (*i.e.,* port number, source and destination IP, packet size, inter-arrival time, and payload) to identify its category.

With the growth of diverse applications over the Internet, it becomes extremely important to identify the name or type of application using its traffic flow, to meet the QoS requirements. As the traffic flow contains a sequence of time-stamped packets (generated by the associated application), it can be treated as a time series [11]. Specifically, the traffic flow consists of multiple correlated statistics (*e.g.,* size, inter-arrival time, direction) against each packet. These statistics form a Multivariate Time Series (MTS) corresponding to a traffic flow. Classification of such an MTS refers to the prediction of its class label (or application) by using a training dataset [12].

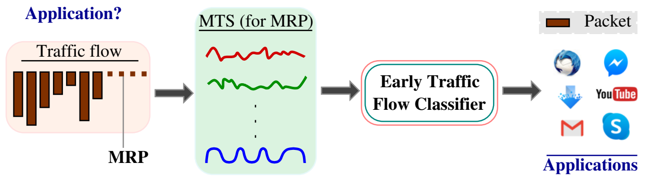


Fig. 1: An example of early classification for identifying an application by using the MTS of its traffic flow.

Classification of the traffic flow is a cumbersome and time consuming task if one wishes to use all the arrived packets (or end of the flow) [7]. However, for real-time traffic classification, it is essential to identify an application (associated with the flow) as early as possible, without waiting for all the data packets. So that the service provider can take appropriate action (*e.g.,* block unwanted application and allocate demanded resources) at the earliest, according to the identified application accessed by a device. A classification approach that can identify an application using an incomplete MTS (traffic flow with fewer data packets, not all) is called as *early classification of MTS* [13], [14]. As an early classification approach uses fewer packets of flow, it influences the accuracy of classifier. It suggests that the early classification approach should estimate Minimum Required Packets or data Points (MRP) from training data while maintaining an adequate level of accuracy. Fig. 1 illustrates early classification of a traffic flow using its MTS, to identify the associated Internet application that has generated the traffic flow.

Previous studies have proposed various classification approaches to identify the application or protocol. A comprehensive survey of the existing approaches can be found in [1]. We discuss some of the recent and novel contributions here. In [15], [16], the authors attempted to identify the network application or protocol by using the features extracted from the packets' contents of the traffic flow. The authors in [17] introduced a feature based traffic flow classification approach to distinguish the sensory data from multimedia. A wavelet

ISBN 978-3-903176-39-3 © 2021 IFIP

based approach is proposed in [18] that considered only packet size (within a specified time window) of the flow for classification. In a work [11], the authors developed a sampling strategy to filter out the zero-length packets (containing only control information) that are used to classify the encrypted TCP flows. Another work [8] also attempted to identify the applications such as Bing, Slack, Youtube, *etc.,* using encrypted packet contents. Slightly different from above studies, the authors in [3] developed an intrusion detection technique for protecting network devices by classifying the network protocols such as Message Queue Telemetry Transport (MQTT), Hyper Text Transfer Protocol (HTTP), and Domain Name Service (DNS). Further, deep neural networks have also been adopted for the traffic flow classification. For example, Rezaei *et al.* [7] employed a Convolutional Neural Network (CNN) to identify the applications by using the size and inter-arrival time information of packets. The work in [19] attempted to mitigate the problem of traffic burst in data center networks by classifying flows. Finally, the authors in [6] developed a CNN based traffic classifier, to identify the application using its Quick UDP Internet Connection (QUIC) traffic flow (may be generated through simulation [20]) while extracting thousands of packet-based deep features from the flow.

A. Motivation

Though the problem of traffic classification is quite mature, there are still some research issues that are not addressed yet by the existing work.

- As most of the existing approaches [7], [11], [15] used univariate time series of packet-based statistics to classify the traffic flow, they could not utilize the *correlation* among time series. Such a correlation helps to obtain better identifiable patterns in the flow and thus improves the interpretability and accuracy of the classification process with fewer packets.
- The content of first few packets of the flow can classify the associated application correctly [15], [16]. However, the approaches in [15], [16] fail to classify the flow if the packets are encrypted.
- The existing work [7], [8] managed to achieve an accuracy of around 90% by using first 30 to 60 packets of flow, they do not establish any meaningful *relationship between MRP and accuracy*. Such a relation can provide a crucial information to the service providers who is interested to classify the flow with minimum number of packets bu with some desired level of accuracy.
- In addition to above, some approaches [18], [21] can classify the traffic flow only when it is ended and thus do not provide any earliness in the classification.

With the goal of addressing the above issues, we solve the following problem: *how to identify an Internet application by using an MTS of statistics (packet size, inter-arrival time, and direction) of packets?* In particular, this work proposes an **E**arly traffic **F**low **C**lassification (EFC) approach.

B. Major contributions

We make following contributions in this paper:

- We propose an early classification approach (EFC) to identify an Internet application by using its traffic flow. EFC approach uses the MTS of packet-based statistics including size, inter-arrival time, and direction, without accessing the content of packets.
- EFC approach employs k-means clustering and deep neural networks, in particular Long Short Term Memory (LSTM) model, to estimate class-wise MRPs using a given training dataset. Such estimated MRPs are used to provides earliness in the classification.
- This work also develops a class forwarding method to incorporate the correlation between the time series during classification. The correlation helps in improving the accuracy as well as earliness.
- Finally, we collect a real-world traffic flow dataset for five common Internet applications. This work carried out several experiments to evaluate the performance of EFC approach on the collected and two existing datasets using accuracy, earliness, and F_1 score.

The rest of the paper is structured as follows. Next section discusses the terminologies and notations that are used throughout this work. Section III presents the proposed early classification approach for traffic flow using MTS. We report the experimental results in Section IV to show the performance of EFC approach independently and with comparison of recent existing approaches. Finally, Section V concludes the paper.

II. Preliminary

This section defines terminologies and notations to improve the readability of the paper. We also give an intuition behind the early classification of traffic flow later in this section. Table I presents the list of notations used in this work.

Let $\mathbf{D} = \{(\mathcal{X}^j, \mathcal{Y}_j) \mid 1 \leq j \leq N\}$ denote an MTS dataset where $\mathcal{X}^j$ is an instance of MTS belonging to class label $\mathcal{Y}_j$. Each MTS in $\mathbf{D}$ belongs to one of the l class labels in $\{y_1, y_2, \cdots, y_l\}$. In this work, an MTS corresponds to a traffic flow, which consists of correlated time series of packet-based statistics such as size, inter-arrival time, and direction. In other words, the traffic flow is a 3-dimensional time series. A class label corresponds an associated application or protocol that generated the traffic flow in the network.

Definition 1 (**Component**). *A time series is referred as component if it is a part of MTS. It is a sequence of temporal ordered data points taken for a fixed period of time. Let n be the number of components in the MTS of $\mathbf{D}$ and $\mathfrak{X}$ be a component of the MTS. Now, j^{th} MTS of $\mathbf{D}$ can be given as $\mathcal{X}^j = \{\mathfrak{X}_i^j \mid 1 \leq i \leq n\}$. A dataset with all N time series corresponding to one particular component of the MTS is denoted by $\mathcal{D} = \{\mathfrak{X}^j \mid 1 \leq j \leq N\}$.*

Definition 2 (**Complete MTS**). *An MTS is said to be complete if it consists of the data points (packet-based statistics) corresponding to all packets of a complete (or ended) flow. Let the* *length of a complete MTS is denoted by T. A complete MTS can now be given as $\mathscr{X} = \{\mathcal{X}_i \mid 1 \leq i \leq n\}$ where $\mathcal{X}_i \in \mathbb{R}^T$.*

Definition 3 (Earliness). *It is defined as the percentage of packets (of a complete traffic flow) that are not used in the classification of the flow. Earliness of an MTS (corresponding to a traffic flow) is mathematically expressed as*

$$Earliness(in\ \%) = \frac{T - t}{T} \times 100, \tag{1}$$

where T and t denote length of complete MTS and number of data points or packets used in the classification, respectively.

A. Long Short Term Memory (LSTM)

It is a type of recurrent neural network, which is capable enough to capture long term temporal dependencies in a given sequential input (*i.e.,* time series) [22]. A LSTM unit consists of different memory blocks known as cells. These cells retain the relevant information which has occurred at arbitrary time steps in the time series. LSTM regulates the flow of information from input to output using a gated mechanism. It uses three different gates (*i.e.,* input, output, and forget) to update the information of different cells and to decide whether the current information should be retained or not.

For a given input time series $\mathcal{X} \in \mathcal{D}$, let $\mathcal{X}_{(t)}, h_{t-1}$, and c_{t-1} denote a current input vector at time t, output state at time $t-1$, and cell state at time $t-1$, respectively. At time step t, a LSTM unit performs following operations:

$$\begin{aligned}
f_t &= \sigma(\mathbf{W}_f \mathcal{X}_{(t)} + \mathbf{V}_f h_{t-1}),\\
i_t &= \sigma(\mathbf{W}_i \mathcal{X}_{(t)} + \mathbf{V}_i h_{t-1}),\\
\tilde{c}_t &= tanh(\mathbf{W}_c \mathcal{X}_{(t)} + \mathbf{V}_c h_{t-1}),\\
c_t &= f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t,\\
h_t &= \sigma(\mathbf{W}_o \mathcal{X}_{(t)} + \mathbf{V}_o h_{t-1}) \otimes tanh(c_t),
\end{aligned} \tag{2}$$

where $\mathbf{W}_k$ and $\mathbf{V}_k$ denote weight matrices that are learned by LSTM during training and $k \in \{f, i, c, o\}$. The symbols σ and $\otimes$ denote sigmoid activation function and element-wise product, respectively.

TABLE I: Notations used in this work.

Symbol	Description	Symbol	Description
$\mathbf{D}$	Training MTS dataset	$\mathscr{X}$	A multivariate time series
$\mathcal{D}$	Dataset with only one component	$\mathscr{X}^j$	j^{th} MTS of dataset $\mathbf{D}$
T	Length of complete MTS	$\mathcal{Y}_j$	Class label of $\mathscr{X}^j$
N	Number of MTS in $\mathbf{D}$	$\mathcal{X}_i$	i^{th} component of MTS
n	Number of components in MTS	$\mathcal{X}_i^j$	i^{th} component of $\mathscr{X}^j$
l	Total number of classes in $\mathbf{D}$	$\mathcal{M}_c$	MRP for y_c class on $\mathcal{D}$
$\mathcal{X}$	An univariate time series	$\mathcal{W}_i$	MRP for y_c class on $\mathbf{D}$
$\mathcal{X}_{(t)}$	Time series of length t	$\boldsymbol{\beta}$	Desired level of accuracy

B. Intuition behind early classification of traffic flow

In this work, early classification of traffic flow mainly refers to identification of an application, associated with the traffic flow, by using minimum number of packets. Such early classification is required to make quick decisions about the services that are being provided to a particular application. In addition, it helps to save significant amount of processing and computational power during classification as only first few packets are to be processed. Further, MTS representation of the traffic flow allows the incorporation of correlation that may exist among the components. Such correlation improves the earliness as well as interpretability of the classification process.

III. Early Classification Approach for Traffic Flow using MTS

In this section, we propose an early classification approach (EFC) to identify the application associated with a traffic flow. The approach uses MTS of packet-based statistics of the flow for its classification. The approach consists of mainly two phases: training and testing. In training phase, the approach builds an early classifier by estimating class-wise MRPs using a given MTS training dataset $\mathbf{D}$. In testing phase, the classifier predicts the class label of an incomplete MTS (corresponding to a traffic flow) by using the obtained MRPs. The approach assumes that the training dataset contains only labeled and complete MTS. Fig. 2 illustrates an overview of EFC approach where an MTS dataset is given as input, consisting of packet-based statistics against the traffic flow generated by various Internet applications.

A. Training phase

This phase builds an early classifier by estimating MRPs for each class label. EFC approach employs k-means clustering and LSTM model to compute the MRPs. For a given dataset $\mathbf{D}$, EFC approach first computes class-wise MRPs for each component separately. Later, it estimates the MRPs for the MTS with the help of obtained MRPs of individual components using a correlation based similarity measure. Training phase is summarized from Lines 1 to 18 in Algorithm 1. Let $\mathcal{D} = \{\mathcal{X}^j, \mathcal{Y}_j | 1 \leq j \leq N\}$ be a part of given dataset $\mathbf{D}$, which contains all time series (*i.e.,* N) of a single component, where $\mathcal{X}^j$ and $\mathcal{Y}_j$ denote j^{th} time series and its class label $\mathcal{Y}_j \in \{y_1, y_2, \cdots, y_l\}$, respectively.

*1) **MRP estimation for** $\mathcal{D}$:* For a given dataset $\mathcal{D}$, the objective of EFC approach is to compute an MRP corresponding to each class label. In order to compute the MRPs, the approach learns a mapping $\mathcal{X}_{(t)} \rightarrow \mathcal{Y}$ with varying length of time series, *i.e.,* $1 \leq t \leq T$, where T denotes the length of complete time series. For a time series $\mathcal{X} \in \mathcal{D}$, EFC approach obtains the MRP using following steps:

(a) *Obtain prior probabilities using k-means:* In this step, EFC applies k-means clustering on the given dataset $\mathcal{D}$ with $k = l$, where l is the total number of class labels. k-means takes complete time series (*i.e.,* with length T) to obtain the clusters of similar instances. Let $\mathcal{H}$ denote the confusion matrix (of size $l \times l$) obtained after clustering. Prior probability of any class label y_c can be obtained using $\mathcal{H}$ as

$$Pr(\mathcal{G}_c) = \mathcal{H}_{cc}, \tag{3}$$

where $\mathcal{G}_c$ denotes the cluster obtained corresponding to the true class label y_c and $1 \leq c \leq l$. The probability that $\mathcal{X}_{(t)}$ belongs to $\mathcal{G}_c$ is computed by finding its similarity with centroids of all the l clusters using Euclidean distance. Let s_c be the similarity between time series $\mathcal{X}_{(t)}$ and cluster $\mathcal{G}_c$ using first t data points

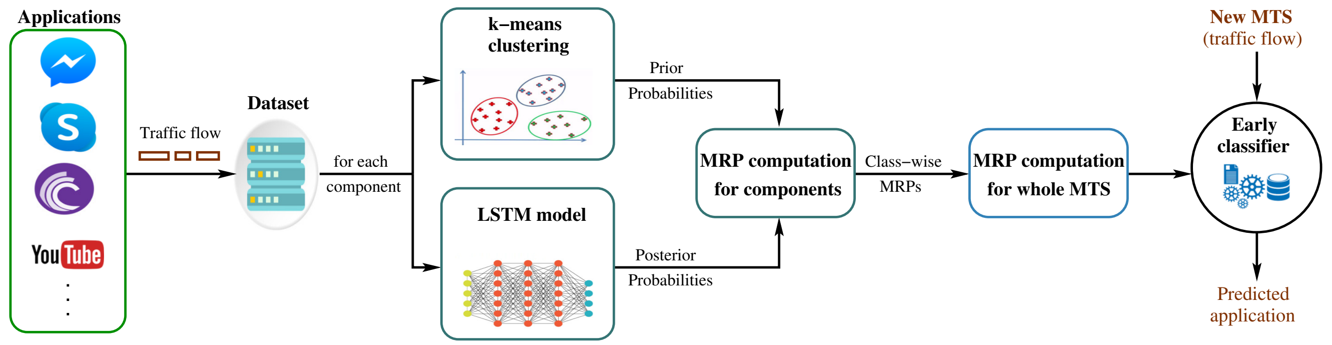


Fig. 2: An overview of EFC approach for traffic flow classification using MTS of packet-based statistics.

only. Such a similarity is given as $s_c = dist(\mathcal{X}_{(t)}, \mathcal{G}_c)$, where $dist(\cdot)$ denotes the Euclidean distance. Any other distance metric can readily be incorporated here. Next, EFC computes an average distance from $\mathcal{X}_{(t)}$ to the clusters, as follows

$$\bar{S} = \frac{1}{l}\sum_{c=1}^{l} s_c. \tag{4}$$

Now, the probability that $\mathcal{X}_{(t)}$ belongs to a cluster $\mathcal{G}_c$ is

$$Pr\left(\frac{\mathcal{G}_c}{\mathcal{X}_{(t)}}\right) = \frac{\delta_c}{\sum_{c=1}^{l}\delta_c}, \quad \text{where } \delta_c = \bar{S} - s_c. \tag{5}$$

(b) *Obtain posterior probabilities using LSTM model:* EFC approach employs a LSTM model to obtain posterior probabilities of the class labels. The model consists of 3 sequentially connected LSTM units (with 16 neurons each) followed by a Fully Connected (FC) layer and a softmax function, as shown in Fig. 3. The softmax computes class-wise posterior probabilities for a given training dataset $\mathcal{D}$. The LSTM model learns a mapping $\Gamma_t : \mathbb{R}^t \to \mathcal{Y}$, where $1 \leq t \leq T$. The posterior probability that a time series $\mathcal{X}_{(t)}$ belongs to a class label y_c, can be obtained using Γ_t as

$$Pr\left(\frac{y_c}{\mathcal{X}_{(t)}}\right) = \frac{e^{z_c}}{\sum_{c=1}^{l} e^{z_c}}, \tag{6}$$

where $z_c = \mathbf{W}_c\mathcal{X}_{(t)} + \mathbf{b}_c$, and the weight matrix $\mathbf{W}_c$ and bias matrix $\mathbf{b}_c$ are learned during the training of the model.

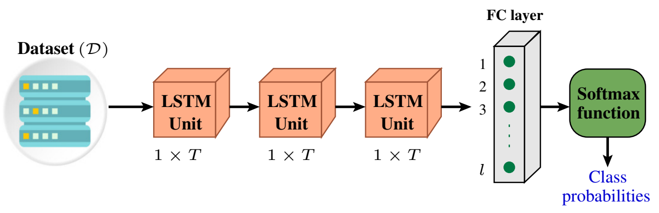


Fig. 3: Architecture of LSTM model.

(c) *MRP computation for $\mathcal{X}$:* This step first computes a relative accuracy $\mathcal{A}_t$ using t data points of the time series (*i.e.,* $\mathcal{X}_{(t)}$). The relative accuracy that $\mathcal{X}_{(t)}$ belongs to a class label y_c can be computed using Eqs. 3, 5, and 6 as

$$\mathcal{A}_t = \frac{Pr\left(\frac{y_c}{\mathcal{X}_{(t)}}\right)\cdot Pr\left(\frac{\mathcal{G}_c}{\mathcal{X}_{(t)}}\right)}{\beta Pr(\mathcal{G}_c)}, \tag{7}$$

where β denotes a desired level of accuracy of the classification. Now, we can estimate MRP of a time series $\mathcal{X} \in \mathcal{D}$, by maximizing the tradeoff between relative accuracy $\mathcal{A}_t$ and earliness (*i.e.,* $\frac{T-t}{T}$) as

$$\text{MRP}(\mathcal{X}) = \underset{t}{\text{argmax}}\left\{\frac{2\times\mathcal{A}_t\times(\frac{T-t}{T})}{\mathcal{A}_t + (\frac{T-t}{T})}\right\}. \tag{8}$$

After estimating the MRPs for all the time series of $\mathcal{D}$, EFC computes a representative MRP for each class label in $\{y_1, y_2, \cdots, y_l\}$. Let $\mathcal{M}_c$ be the MRP for y_c class label, which is estimated using following expression

$$\mathcal{M}_c = \frac{1}{N_c}\sum_{j=1}^{N_c}\text{MRP}(\mathcal{X}^j), \tag{9}$$

where N_c denotes the number of time series that belongs to class label y_c. The estimated class-wise MRPs for the dataset $\mathcal{D}$ (consists the time series of i^{th} component only), are stored into a vector $\boldsymbol{\mathcal{M}}_i = \{\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_l\}$. For the dataset $\mathbf{D}$, the sets of class-wise MRPs is obtained as $\boldsymbol{\mathcal{M}} = \{\boldsymbol{\mathcal{M}}_1, \boldsymbol{\mathcal{M}}_2, \cdots, \boldsymbol{\mathcal{M}}_n\}$.

2) ***MRP estimation for whole MTS dataset* $\mathbf{D}$:** Once MRPs are computed for each of individual component, EFC approach attempts to estimate class-wise MRPs for the whole dataset $\mathbf{D}$ by proposing a correlation based similarity measure. Here, the approach incorporates the correlation among the components of MTS while computing the MRPs. In this work, the correlation is incorporated by transforming each MTS of the dataset $\mathbf{D}$ into a matrix. As the MTS has n components, it provides a correlation matrix of size $n \times n$. Such a matrix is constructed by computing a correlation coefficient between each pair of components of the MTS. For any two components $\mathcal{X}_a$ and $\mathcal{X}_b$, the correlation coefficient is defined as

$$\mathcal{C}_{\mathcal{X}_a,\mathcal{X}_b} = \frac{\sum_{t=1}^{T}(\mathcal{X}_{a,(t)} - \mu_{\mathcal{X}_a})(\mathcal{X}_{b,(t)} - \mu_{\mathcal{X}_b})}{(T-1)\sigma_{\mathcal{X}_a}\sigma_{\mathcal{X}_b}}. \tag{10}$$

As the dataset $\mathbf{D}$ consists of N MTS, we get N corresponding correlation matrices. The EFC approach utilizes these matrices to compute a similarity score between the MTS. For any two MTS $\mathcal{X}^j$ and $\mathcal{X}^k$ in $\mathbf{D}$, similarity score is defined as

$$Sim(\mathcal{X}^j, \mathcal{X}^k) = \sqrt{\frac{1}{n \times n} \sum_{i=1}^{n} \sum_{i'=1}^{n} (\mathcal{C}^j[i, i'] - \mathcal{C}^k[i, i'])}, \quad (11)$$

where $\mathcal{C}^j$ denotes an obtained confusion matrix for j^{th} MTS. Using Eq. 11, EFC constructs a similarity matrix $\mathcal{S}$ of size $N \times N$ which contains similarity scores between each pair of the MTS using T data points. Given the set of MRPs (*i.e.,* $\boldsymbol{\mathcal{M}}$) for the individual component and similarity matrix $\mathcal{S}$, we estimate the class-wise MRPs for MTS using following steps:

(a) Build a classifier Γ using LSTM model on similarity matrix $\mathcal{S}$ to compute the accuracy $\boldsymbol{A}_{T,c}$ for label y_c.
(b) Find minimum length of MTS to start the estimation of MRPs as $start = \min\{\boldsymbol{\mathcal{M}}\}$.
(c) For $t = start$ to T:
   - Obtain similarity matrix $\mathcal{S}$ using Eq. 11 by taking first t data points of the MTS.
   - Compute accuracy $\boldsymbol{A}_{t,c}$ for y_c using the classifier Γ_t.
   - If $\beta \boldsymbol{A}_{T,c} \leq \boldsymbol{A}_{t,c}$ satisfies then t becomes the MRP for class label y_c.

By repeating above steps for all l classes, EFC gets a separate MRP for each class label, which are denoted as $\boldsymbol{\mathcal{W}} = \{\mathcal{W}_1, \mathcal{W}_2, \cdots, \mathcal{W}_l\}$.

B. Testing phase

In this phase, EFC approach predicts class label of a new incomplete MTS which is generated corresponding to a traffic flow. Let $\mathcal{X}^p$ denotes the new MTS of the packet-based statistics of the traffic flow. For the given class-wise MRPs, EFC approach uses a class forwarding method to incorporate correlation among the components of $\mathcal{X}^p$ in the classification. The approach utilizes trained classifier Γ_t to predict the class label of $\mathcal{X}^p$ using only first t packets of the traffic flow. Let t be the number of packets arrived in the traffic flow and t' be the minimum required packets (MRP) that are estimated during training phase for the correct prediction with β. The class forwarding method consists of following steps:

(a) Obtain minimum number of packets of the MTS as $t' = \min\{\boldsymbol{\mathcal{W}}\}$.
(b) If $t' < \min\{\boldsymbol{\mathcal{M}}_1\}$ then update $t' = \min\{\boldsymbol{\mathcal{M}}_1\}$.
(c) For component $i = 1$ to $n - 1$:
   - If $t' < t$ then predict the class label of i^{th} component of $\mathcal{X}^p$ using Γ_t else wait for more packets. Let $\hat{\mathcal{Y}}_i$ be the predicted class label.
   - Update $t' = \boldsymbol{\mathcal{M}}_{i+1}[\hat{\mathcal{Y}}_i]$ for next component.
(d) Find the class label $\mathcal{Y}_p$ predicted by majority of components and assign it to the MTS $\mathcal{X}^p$.
(e) Obtain earliness by the number of packets that are used for predicting the class label $\mathcal{Y}_p \in \{y_1, y_2, \cdots, y_l\}$.

Algorithm 1 summarizes the testing phase at Lines 20 and 21.

Time complexity: In Algorithm 1, as the **for** loop at Line 1 can be executed parallelly for different components, it can be removed from the algorithm while computing complexity.

Algorithm 1: Early traffic flow classification approach

```
Input: A dataset $\mathbf{D}$ consists of $N$ labeled MTS instances
       corresponding to $N$ traffic flows and each has $n$
       components with $T$ data points. A testing MTS
       $\mathcal{X}^p \notin \mathbf{D}$ for which class label is to be predicted;
Output: Set of class-wise MRLs for each component as $\boldsymbol{\mathcal{M}}_i$
        and for whole dataset $\mathbf{D}$ as $\boldsymbol{\mathcal{W}}$, and predicted class
        label $\mathcal{Y}_p$;
1  for $i \leftarrow 1$ to $n$ do
       /* Dataset $\mathcal{D}$ contains all time series of $i^{th}$ component */
2      Apply $k$-means clustering on $\mathcal{D}$ with $k = l$.
       /* $\mathcal{H}$ is a confusion matrix of size $l \times l$ */
       /* $\mathcal{G}_c$ is a cluster for $y_c$ class label */
3      Obtain prior probability as $Pr(\mathcal{G}_c) = \mathcal{H}_{cc}$.
       /* $\mathcal{X} \in \mathcal{D}$ is a time series with class label $y_c$*/
4      for $t \leftarrow 1$ to $T$ do
           /* $\mathcal{X}_{(t)}$ is a time series with first $t$ data points */
5          Compute $Pr\left(\frac{\mathcal{G}_c}{\mathcal{X}_{(t)}}\right)$ using Eq. 5.
6          Build LSTM model on $\mathcal{D}$ using $t$ data points.
7          Compute $Pr\left(\frac{y_c}{\mathcal{X}_{(t)}}\right)$ using Eq. 6.
           /* $\beta$ represents the desired level of accuracy */
8          Compute relative accuracy $\mathcal{A}_t$ using Eq. 7.
9          Compute $\mathcal{U}_t = \frac{2 \times \mathcal{A}_t \times (\frac{T-t}{T})}{\mathcal{A}_t + (\frac{T-t}{T})}$.
10         Append $\mathcal{U} \leftarrow \mathcal{U}_t$.
       /* $t'$ is an index of maximum utility value in $\mathcal{U}$ */
11     Estimate MRP($\mathcal{X}$) $= t'$.
12     Compute MRPs of all time series of $\mathcal{D}$ using Lines $4 - 12$.
       /* Obtain class-wise MRPs for $\mathcal{D}$ using Eq. 9 */
13     $\boldsymbol{\mathcal{M}}_i = \{\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_l\}$.
14 Class-wise MRPs for $\mathbf{D}$ as $\boldsymbol{\mathcal{M}} = \{\boldsymbol{\mathcal{M}}_1, \boldsymbol{\mathcal{M}}_2, \cdots, \boldsymbol{\mathcal{M}}_n\}$.
   /* Obtaining MRPs for $\mathbf{D}$ using all components together */
15 for $j \leftarrow 1$ to $N$ do
16     for $k \leftarrow 1$ to $N$ do
17         Compute similarity score $Sim(\mathcal{X}^j, \mathcal{X}^k)$ using Eq. 11.
18         Append $\mathcal{S} \leftarrow Sim(\mathcal{X}^j, \mathcal{X}^k)$.
   /* Using steps (a) to (c) shown in Section III-A2 */
19 Compute class-wise MRPs as $\boldsymbol{\mathcal{W}} = \{\mathcal{W}_1, \mathcal{W}_2, \cdots, \mathcal{W}_l\}$.
   /* Given a testing MTS $\mathcal{X}^p \notin \mathbf{D}$ */
   /* Using steps (a) to (e) shown in Section III-B */
20 Predict a class label $\mathcal{Y}_p$ for $\mathcal{X}^p$.
   /* $t'$ is the number of packets of $\mathcal{X}^p$ that are used for $\mathcal{Y}_p$ */
21 Compute earliness as $\frac{T-t'}{T} \times 100$.
```

Now, the time complexity of Algorithm 1 mainly depends on the **for** loop at Line 4 and the complexity of LSTM model. The time complexity of LSTM per time step is O($\mathbf{W}$) [22], where $\mathbf{W}$ is the number of weight parameters that are learned during training. At this point, the time complexity of Algorithm 1 can be given as $O(T\mathbf{W})$.

IV. Experimental Evaluation

We carried out several experiments to evaluate the performance of EFC approach on three datasets: one collected and two existing including QUIC [23], [24] and ISCX [25]. The approach is also compared with four existing approaches including [3], [6], [7], [15] and the results are reported later in this section.

A. Datasets

We describe each of the used datasets in detail for clear understanding of the reported results.

*1) **Collected dataset**:* We collected traffic flow data for five Internet applications using *tcpdump* utility on Linux operating system (in particular Ubuntu 20.04). The considered applications are Skype (**A1**), KTorrent (**A2**), Gmail desktop (**A3**), Facebook messenger (**A4**), and Youtube viewer (**A5**). We captured 1000 traffic flows for each of the applications to keep uniform distribution of instances among the five classes. After preprocessing, the dataset consists of 5000 labeled MTS instances that are obtained by computing three statistics: size, inter-arrival time, and direction. We refer the created dataset as MTS of Flow (MTSF).

*2) **QUIC dataset**:* This dataset contains labeled traffic flow data for five Google applications (*i.e.,* class labels) including Drive (**B1**), Youtube (**B2**), Docs (**B3**), Search (**B4**), and Music (**B5**). The authors in [24] collected this dataset to classify the Google applications by using their QUIC traffic flows. In the experiments, we use pretraining directory of the available dataset, where total 6589 labeled flows are given. The flow consists of three statistics (packet size, inter-arrival time, and direction) corresponding to each packet. Since a flow contains several packets in a temporal order, it corresponds to an MTS of correlated statistics. In order to have uniform class distribution, this work considers 600 flows from each application for the experimental evaluation of EFC approach. In total, we use the QUIC dataset with 3000 labeled flows.

*3) **ISCX dataset**:* This dataset was captured to identify various categories of traffic flow including email, streaming, voice over IP, and so on. It contains traffic flow data of 14 different applications such as Gmail chat, Mail, Torrent, *etc.* As the EFC approach builds a LSTM model, the number of instances of each application must be sufficient to train the classifier. We therefore consider following applications in the experiments: Gmail chat (**C1**), Mail (**C2**), Torrent (**C3**), Vimeo (**C4**), Youtube (**C5**), Facebook audio (**C6**), and Handgout audio (**C7**). This work uses 400 instances of each of seven applications for the performance evaluation of EFC approach. We use the ISCX dataset with total 2800 labeled traffic flows.

B. Experimental results

This work first divides the datasets into training and testing data with 70% and 30% instances, respectively. The proposed approach uses training data for building EFC approach by learning class-wise MRPs and testing data for its performance evaluation. We preferred Python language with Keras library to implement the approach. The deep learning classifier is trained with following hyperparameters: batch size = 100, optimizer = 'sgd', and learning rate = 10^{-2}. This work uses following evaluation metrics:

- *Accuracy:* It is percentage ratio of number of correctly predicted flows (*i.e.,* MTS instances) to the total number of flows in the testing data.
- *Earliness:* It is computed using Eq. 1.
- *F_1 score:* It measures the balance between quality and relevance of the classification by taking harmonic mean of precision and recall.

With above metrics, we attempt to answer following questions:

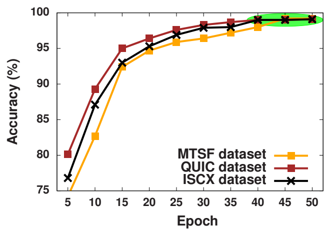
- What is the minimum number of packets in complete MTS to justify the meaning of earliness? (Section IV-B1)
- What is the minimum number of epochs required to converge the learning of classifier? (Section IV-B2)
- How does EFC approach perform on the testing data? (Section IV-B3)
- What is the class-wise performance of the approach for different datasets? (Section IV-B4)
- How does the desired level of accuracy (β) impact on the performance? (Section IV-B5)

*1) **Obtaining number of packets in complete MTS** (T):* At first, the approach carried out an experiment to find the minimum number of packets in the flows that can provide maximum accuracy (*i.e.,* above 99%). The EFC approach is tested by taking varying number of packets with a difference of 5 packets, $\beta = 1$, and epochs $= 50$. The obtained accuracy results are shown in Table II. We observe that EFC approach is able to classify the traffic flows with an accuracy of more than 99%, for all the used datasets, by using only first 25 to 30 packets of the flow. This work therefore sets the length of complete MTS to first 30 packets, *i.e.,* $T = 30$. Such complete MTS is later used to estimate the class-wise MRPs for different values of β.

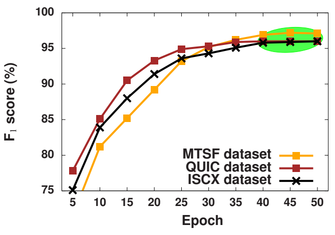
TABLE II: Accuracy with varying number of packets.

	Dataset	Number of packets					
		5	**10**	**15**	**20**	**25**	**30**
Accuracy (%)	MTSF	80.7	91.6	97.5	98.9	99.5	99.6
	QUIC	83.5	94.2	98.9	99.3	99.8	99.8
	ISCX	81.2	92.3	95.5	99.0	99.3	99.3

*2) **Impact of number of epochs**:* Next, we conduct experiments with different number of epochs using training data and evaluate the learning performance of the classifier on the same data. Fig. 4 shows the obtained accuracy and F_1 score results with varying number of epochs. It is clear from the results that there exists a substantial rise in performance up to 40 epochs for the datasets. Later, from 40 to 50 epochs, we see a negligible change in the accuracy and F_1 score, as indicated by an ellipse. We therefore carried out all the subsequent experiments with 50 epochs.



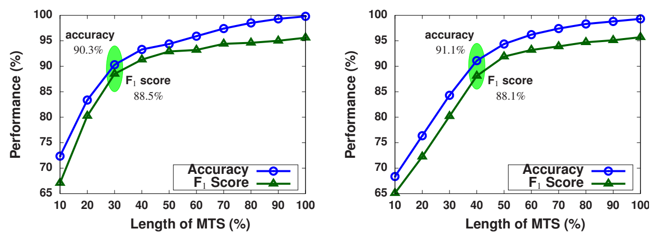
(a) Impact on accuracy



(b) Impact on F_1 score

Fig. 4: Impact of number of epochs on the performance.

*3) **Performance results on testing data:*** As the main objective of EFC approach is to classify a traffic flow as early as possible by using minimum number of packets, this work conducts an experiment to validate the performance of EFC along the progress of MTS. Fig. 5 shows the performance results along the traffic flow with an interval of 10%. It is clear from part (a) of Fig. 5 that EFC achieved an accuracy of more than 90% by using 30% length of MTS (*i.e.,* just 9 packets of the flow) as indicated by an ellipse. Moreover, here the value of F_1 score is 88.5%, which indicates that EFC approach is able to maintain good balance between quality and relevance of classification for MTSF dataset. Further, for QUIC dataset, the approach required 40% of packets (*i.e.,* 12 packets) of complete MTS to obtain an accuracy of 91.1%.



(a) MTSF dataset (b) QUIC dataset

Fig. 5: Performance of EFC approach along the length of traffic flow.

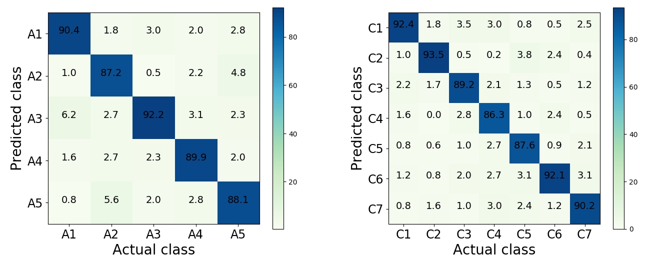
*4) **Class-wise performance of EFC approach:*** In order to understand the class-wise behavior of EFC approach, this work presents the performance results for different Internet applications in Table III. We observe that the approach achieved maximum accuracy for *Gmail desktop*, *Docs*, and *Mail* applications of MTSF, QUIC, and ISCX datasets, respectively. It indicates that traffic flow of these applications have more identifiable patterns in their MTS. Similar observations can be made about F_1 scores. However, the applications that are classified with maximum accuracy, have shown poor performance on earliness metric. An interesting observation in the results is about *Youtube* application where the approach obtained almost equal accuracy using the traffic flow for all the datasets. Moreover, the earliness is maximum (*i.e.,* 59.7%, 61.2%, and 58.9) for *Youtube* application.

We also present the confusion matrices for MTSF and ISCX datasets in Fig. 6. The EFC approach has maximum confusion of 6.2% between **A1** and **A3** classes, indicating some of the testing instances for these classes have similar patterns. On the other hand, EFC approach can clearly distinguish **C2** class from **C4** with 0% confusion, as shown in part (b) of Fig. 6.

*5) **Impact of β on performance:*** Next, we performed experiments to analyze the impact of desired level of accuracy β on the performance of the EFC approach and the results are illustrated in Fig. 7. As the desired level of accuracy increases, the earliness decreases, however the approach is still able to manage more than 50% of earliness even at $\beta = 0.9$ for all three datasets. The results clearly indicate that the proposed approach always maintained the desired level of accuracy by compromising the earliness, *i.e.,* by using more number of packets of the flow.

TABLE III: Performance of the proposed approach for different applications (classes) with $\beta = 0.9$.

Dataset	**Application**	**Performance metrics**		
		Accuracy (%)	**Earliness (%)**	**F_1 score (%)**
MTSF	Skype	90.4	52.3	89.5
	KTorrent	87.2	53.2	85.7
	Gmail desktop	**92.2**	50.4	**91.7**
	Facebook messenger	89.9	54.7	90.1
	Youtube viewer	88.1	**59.7**	86.8
QUIC	Drive	91.5	50.3	87.8
	Youtube	87.4	**61.2**	86.1
	Docs	**94.2**	49.1	**92.5**
	Search	87.1	52.3	89.1
	Music	87.8	59.2	90.7
ISCX	Gmail chat	92.4	49.6	89.5
	Mail	**93.5**	45.4	**91.2**
	Torrent	89.2	51.2	86.5
	Vimeo	86.3	46.6	88.2
	Youtube	87.6	**58.9**	86.7
	Facebook audio	92.1	51.1	87.6
	Handgout audio	90.2	47.2	89.2



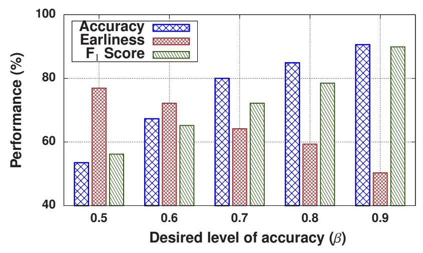
(a) MTSF dataset (b) ISCX dataset

Fig. 6: Illustration of confusion matrices (in %) at $\beta = 0.9$.

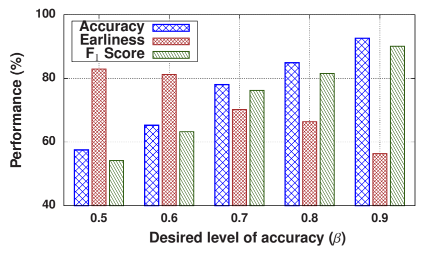
Next, this work evaluates EFC approach based on the number of packets, used during the classification, for different applications. Fig. 8 shows the obtained results for the considered datasets at different values of β. In part (a) of Fig. 8, we observe that the approach requires maximum number of packets (*i.e.,* 15) to identify **A3** class (*i.e.,* Gmail desktop) with $\beta = 0.9$. On the other hand, if desired level of accuracy reduces to 0.6 then only 6 packets are sufficient to classify various applications of QUIC dataset. Similarly, 7 packets are sufficient to classify the applications of ISCX network traffic dataset for $\beta = 0.5$, as shown in part (c) of Fig. 8.

C. Comparison with existing approaches

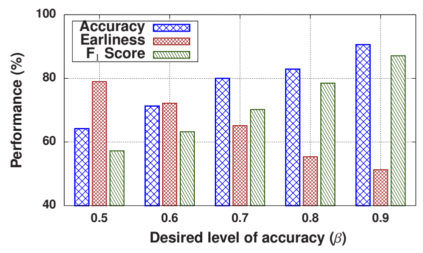
Finally, we compare the proposed approach with four existing approaches including Early Classification of network Traffic (ECT) [15], CNN based Traffic Classifier (CTC) [6], Encrypted Application Classification (EAC) [8], and Ensemble Intrusion Detection (EID) technique [3]. ECT approach claimed that it can classify a traffic flow with first few packets but it needs to access the packet contents and thus unable to classify an encrypted traffic flow. The CTC approach can recognize various Google services using QUIC dataset without accessing the packet contents. Next, the EAC approach in [8] used encrypted payload and inter-arrival time for classifying various applications. Finally, the authors in EID approach classified applications using their traffic flow data.



(a) MTSF dataset

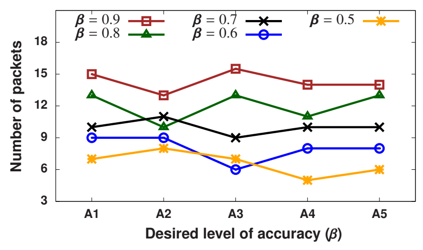


(b) QUIC dataset

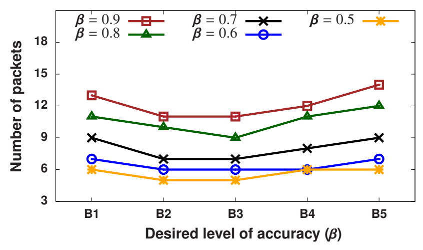


(c) ISCX dataset

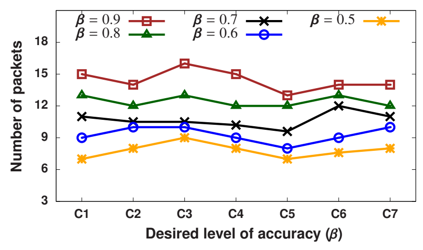
Fig. 7: Impact of β on the performance of EFC approach.



(a) MTSF dataset

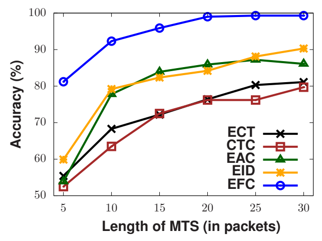


(b) QUIC dataset

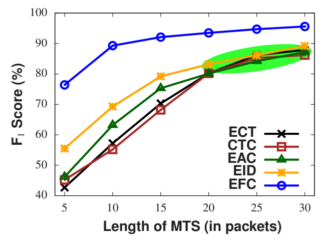


(c) ISCX dataset

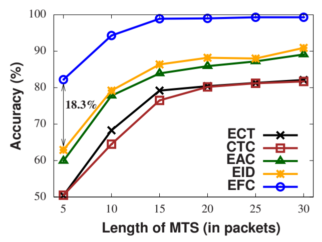
Fig. 8: Number of packets used by EFC approach for different applications.



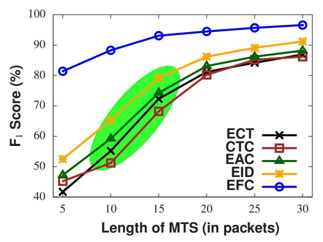
(a) Accuracy on MTSF dataset



(b) F_1 score on MTSF dataset



(c) Accuracy on QUIC dataset



(d) F_1 score on QUIC dataset

Fig. 9: Comparison of EFC approach with existing approaches.

Fig. 9 shows the performance comparison results using accuracy and F_1 score for MTSF and QUIC datasets. We observe the following points in the results:

- EFC approach performs remarkably well on accuracy and F_1 score metrics by using just a few packets of traffic flow. It outperforms the existing approaches on various evaluation metrics on both the datasets.
- With just 5 packets of flow, EFC approach is able to achieve an accuracy of 82.2% which is substantially higher (*i.e.,* 18.3%) than existing approaches as shown in part (c) of Fig. 9. It is due to the presence of correlation between components of MTS which is utilized by EFC.
- The existing approach EID performs significantly better than other existing approaches as it utilizes the advantage of ensemble learning technique during classification.
- Similarly, EAC approach performs better than CTC and ECT, which indicates that it is able extract more identifiable patterns in the traffic flows for recognizing the various applications. Further, a substantial rise is also seen in F_1 scores between 10 to 15 packets as indicated by an ellipse in part (d) of Fig. 9.

V. Conclusion

In this paper, we proposed an early classification approach for traffic flow using the MTS of packet-based statistics such as size, inter-arrival time, and direction. Unlike existing approaches, the proposed approach considered the network traffic flow as MTS and utilized the correlation between its time series, which helps in recognizing the associated application by using only first few packets. The approach built an early classifier by estimating class-wise MRPs using the given labeled network traffic dataset. The MRPs are later used to classify a traffic flow as early as possible. Additionally, we collected a labeled dataset using *tcpdump* utility to evaluate the effective of the proposed approach for five common Internet applications such as Gmail desktop, Youtube viewer, *etc.*

We carried out several experiments to validate EFC approach using the collected and two real-world existing datasets and reported results showed that the approach is able to maintain an adequate level of accuracy by using first 15 packets of the traffic flows. This work motivates further research towards incorporating the concept of transfer learning for identifying an Internet application by the classifier trained on other applications, without accessing the packet contents.

References

[1] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar, "Towards the deployment of machine learning solutions in network traffic classification: a systematic survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1988–2014, 2018.

[2] T. Taleb, A. Ksentini, M. Chen, and R. Jantti, "Coping with emerging mobile social media applications through dynamic service function chaining," *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, pp. 2859–2871, 2016.

[3] N. Moustafa, B. Turnbull, and K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815–4830, 2019.

[4] B. Anderson and D. McGrew, "Machine learning for encrypted malware traffic classification: accounting for noisy labels and non-stationarity," in *Proceedings of SIGKDD*, 2017, pp. 1723–1732.

[5] D. Wu, B. Fang, J. Wang, Q. Liu, and X. Cui, "Evading machine learning botnet detection models via deep reinforcement learning," in *Proceedings of ICC*, 2019, pp. 1–6.

[6] V. Tong, H. A. Tran, S. Souihi, and A. Mellouk, "A novel quic traffic classifier based on convolutional neural networks," in *Proceedings of GLOBECOM*, 2018, pp. 1–6.

[7] S. Rezaei and X. Liu, "Multitask learning for network traffic classification," *ACM SIGCOMM Computer Communication Review*, vol. 48, no. 1, 2019.

[8] K. Yang, L. Xu, Y. Xu, and J. Chao, "Encrypted application classification with convolutional neural network," in *Proceedings of IFIP Networking*, 2020, pp. 499–503.

[9] R. Wang, L. Shi, and B. Jennings, "Training traffic classifiers with arbitrary packet sets," in *Proceedings of ICC Workshops*, 2013, pp. 1314–1318.

[10] R. Li, Z. Zhao, J. Zheng, C. Mei, Y. Cai, and H. Zhang, "The learning and prediction of application-level traffic data in cellular networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3899–3912, 2017.

[11] J. Kampeas, A. Cohen, and O. Gurewitz, "Traffic classification based on zero-length packets," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 1049–1062, 2018.

[12] W. Pei, H. Dibeklioğlu, D. M. J. Tax, and L. van der Maaten, "Multivariate time-series classification using the hidden-unit logistic model," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 4, pp. 920–931, 2018.

[13] A. Gupta, H. P. Gupta, B. Biswas, and T. Dutta, "A fault-tolerant early classification approach for human activities using multivariate time series," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1747–1760, 2021.

[14] K. Li, S. Li, and Y. Fu, "Early Classification of Ongoing Observation," in *Proceedings of ICDM*, 2014, pp. 310–319.

[15] A. Dainotti, A. Pescapé, and C. Sansone, "Early classification of network traffic through multi-classification," in *Proceedings of TMA*, 2011, pp. 122–135.

[16] L. Bernaille, R. Teixeira, and K. Salamatian, "Early application identification," in *Proceedings of the ACM CoNEXT*, 2006.

[17] S. Egea, A. Rego Mañez, B. Carro, A. Sánchez-Esguevillas, and J. Lloret, "Intelligent iot traffic classification using novel search strategy for fast-based-correlation feature selection in industrial environments," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1616–1624, 2018.

[18] H. Shi, H. Li, D. Zhang, C. Cheng, and W. Wu, "Efficient and robust feature extraction and selection for traffic classification," *Computer Networks*, vol. 119, pp. 1–16, 2017.

[19] K. Zhu, G. Shen, Y. Jiang, J. Lv, Q. Li, and M. Xu, "Differentiated transmission based on traffic classification with deep learning in datacenter," in *Proceedings of IFIP Networking*, 2020, pp. 599–603.

[20] T. Völker, E. Volodina, M. Tüxen, and E. P. Rathgeb, "A quic simulation model for inet and its application to the acknowledgment ratio issue," in *Proceedings of IFIP Networking*, 2020, pp. 737–742.

[21] J. Zhang, Y. Xiang, Y. Wang, W. Zhou, Y. Xiang, and Y. Guan, "Network traffic classification using correlation information," *IEEE Transactions on Parallel and Distributed systems*, vol. 24, no. 1, pp. 104–117, 2012.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] QUIC Dataset 2019. [Online]. Available: https://drive.google.com/drive/folders/1Pvev0hJ82usPh6dWDlz7Lv8L6h3JpWhE

[24] S. Rezaei and X. Liu, "How to achieve high classification accuracy with just a few labels: A semi-supervised approach using sampled packets," *arXiv preprint arXiv:1812.09761*, 2018.

[25] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and vpn traffic using time-related," in *Proceedings of ICISSP*, 2016, pp. 407–414.