A novel statistical based feature extraction approach for the inner-class feature estimation using linear regression

Fannia Pacheco
UNIV PAU & PAYS ADOUR
LIUPPA, ANGLET, France 64600
Email: f.pacheco@univ-pau.fr

Ernesto Exposito
and Jose Aguilar
UNIV PAU & PAYS ADOUR
LIUPPA, ANGLET, France 64600
Universidad de Los Andes
Merida, Venezuela
Email: ernesto.exposito@univ-pau.fr
aguilar@ula.ve

Mathieu Gineste
and Cedric Baudoin
Thales Alenia Space
TOULOUSE, France
Email: mathieu.gineste,cedric.baudoin
@thalesaleniaspace.com

***Abstract*—Nowadays, statistical based feature extraction approaches are commonly used in the knowledge discovery field with Machine Learning. These features are accurate and give relevant information of the samples; however, these approaches consider some assumptions, such as the membership of the signals or samples to specific statistical distributions. In this work, we propose to model statistical computation through Linear Regression (LR) models; these models will be divided by classes, in order to increase the inner-class identification likelihood. In general, an ensemble of LR models will estimate a targeted statistical feature. In an online deployment, the pool of LR models of a given targeted statistical feature will be evaluated to find the most similar value to the current input, which will be as the estimated of the feature. The proposal is tested with a real world application in traffic network classification. In this case study, fast classification response has to be provided, and statistical based features are widely used for this aim. In this sense, the statistical features must give early signs about the status of the network in order to achieve some objectives such as improve the quality of service or detect malicious traffic.**

I. Introduction

Feature extraction (FE) is an important step for the application of Machine Learning (ML) in different contexts. These features allow caracterizing the problem, and building models that describe a system or process. Some approaches consider the FE process as the result of the subset of features that maximizes the class separation. Feature selection tries to select the set of most relevant features, and the main proposals in this domain are grouped into Filter, Wrapper and Embedded approaches; which in turn can be developed by supervised and unsupervised strategies. In the supervised strategy, the feature vector has a class label to guide the search process of relevant information. On the other hand, in the unsupervised strategy, the feature vector has no labels; and it is used to determine relationships between features to discover their relevance [1]. In addition, there are FE processes enbededed into the ML algorithm such as the deep learning approach [2]. Moreover, the idea of creating or extracting different features in a multi-class problem is explored by several works [3], [4]. For instance, the work in [5] apply LR for FE and dimentionality reduction in image processing.

Most of the Machine Learning (ML) applications rely on statistical based computations to extract relevant features from processes. Statistical based features are widely used in signal processing, image processing, economics, among others. Examples of the statistical metrics determined from the attributes, are commonly very accurate and give relevant information of the samples. Some approaches consider some assumptions to obtain these features, such as the membership of the samples to specific statistical distributions. Based on these assumptions, different metrics that describe these attributes are defined. For instance, the normallity of the data is usually assumed; however, there are many applications that do not fit into this common or any other distribution [6]. On the other hand, the way in which these features are extracted can add a cumulative error to the measure, given either because the sampling period is not correct or the online feature computation is not accurate. In addition, if we have a sample, conformed of a sequence of ordered events in time, the computation of statistical metrics, such as the mean or variance, can be affected by ouliers and noise; which add incertitude to the problem characterization.

In general, how these features behave in an incremental online calculation, has been largely studied in the statistic and signal processing fields [7], [8], [9], [6]. One solution is called the moving average, which consists in a stable procedure for incremental online calculation. This is an approach to compute the mean considering the previous mean and the current sample value; moreover, it tries to deal with the shifts in the sample values. From a classical point of view, the mean is computed in an incremental online scenario considering the current value of the mean penalized by an error among the current mean and an input value. This error can be either weighted, modeled by exponential distributions, or even by polinomial regression models [10]. Some other advanced approaches can be found

in the literature, such as the logarithmic moving average [11].

In traffic analysis, ML is deployed to classify, comprehend, diagnose or observe the status of the network. Accurate identification and classification of network traffic is a key task for managing bandwidth budget, and ensuring Quality of Service (QoS) objectives. The internet traffic is represented by a sequence of events that are defined by communication protocols. In this domain, statistical based features are highly used, and play a key role in the analysis. They have been largely reported and demonstrated due to its simplicity [12]. For instance, for anomaly detection, the work in [13] lists all the features as classical statistical for unidirectional and bidirectional flows, as well as content type. The work in [14] presents a similar work, remarking the FE approaches from the flow and the packet level. The work in [15] presents a complete study about the statistical features; the authors tested ten classifiers to get into the conclusion that these features allow the classifier to get a good accuracy. However, the work in [16] presents the traffic classification problem from a different point of view, without an explicit FE process. Instead of using the classical statistical features, the authors build a deep learning architecture that learns from the packet content. The approach aims at learning new features for each application with the deep learning architecture. The FE process is embedded and these new features do not have a real meaning, instead are binary data with relationships found by the deep neural network. This last characteristic makes it suitable for encrypted traffic, and represents an alternative approach to statistical based features. However, it is not quantified its boundaries against classical FE and classification processes.

Through this proposal, we would like to exploit the way in which the statistical features are computed in an online manner, in particular with the moving average approach. In addition, the FE will be performed in a way that allows refining the actual features to get better inner-class separation. In this sense, in this paper is proposed:

- A mean model for each class.
- Inner-class distributions for each statistical feature(s) based on LR models.
- A penalization to the error computation when unusual sample values are captured.
- An online update scheme of FE process in presence of a new sample value, where the new feature value will be obtained by evaluating all the LR models. The final feature value will be the one that obtain the smallest error distance respect to the input.

Finally, the FE process will add this new feature computation (the mean, for our case) to perform the ML task. We will test the proposal over the traffic analysis case study, which has all the properties that make it suitable to show our approach.

II. Background

This section briefly presents the moving average approach, and the principles to perform traffic analysis over the internet network.

A. Moving average

From the ML point of view, the mean of a signal can represent a feature that will help to extract valuable knowledge from a process. This measure, with the standard desviation, is often used in diversity of problems. The moving average refers to a numerical calculation of the statistical mean for time series data. This metric aims at reducing the online flutations on the data, in order to obtain a reliable measure in the short and long term. The moving average tries to smooth out shifts in short sampling window scenarios; on the contrary, in large sampling windows, it will stress tendencies. Mathematically speaking, the moving average is interpreted as a convolution function, while in signal processing as a low-pass filter. Its main property is to average online streams for sampling processes in different applications [9].

The simplest procedure with all the samples is given by Eq. 1, while its computation in a moving window is given by Eq. 2. On the other hand, the exponentially weighted mean is given by Eq. 3, where α is the parameter of an exponential distribution. The effect of this parameter over the mean computation is to weight the error with an exponential decrease over time [17]. In brief, the weight is introduced mainly to smooth the average. For instance, the kernel average, the nearest neighbor and the local linear regression smoother, among others, also try capturing the most appropriate value of the mean [18]. Moreover, each of them are extended to compute the variance by using the expectation function, which is a generalized version of the mean.

$$\mu_n = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

$$\mu_n = \mu_{n-1} + \frac{1}{n}(x_n - \mu_{n-1}) \tag{2}$$

$$\mu_n = \mu_{n-1} + \alpha(x_n - \mu_{n-1}) \tag{3}$$

B. Traffic analysis

In order to understand how to perform traffic analysis, we formally define a traffic flow; which in turn is the most common representation of communication sessions in the network. Following, it is briefly explained the statistical based features that allows using ML for traffic analysis.

1) Traffic flows: In traffic analysis is commonly used the term flow to describe communication between peers. A traffic flow, according to [19], is a set of packets or frames in the network intercepted in a monitoring point during a time interval. The packets belonging to the same flow share several common properties, that is: a) transport or application header fields (e.g. destination IP address, destination port number, among others), b) characteristics of the packets such number of MPLS labels, c) additional fields, such as next-hop IP address, output interface, etc.

Therefore, we can outline the following classical definition for an unidirectional flow F_i:

Definition 1: A flow F_i is described by the set,

$$F_i = \{H_i, P_i\} \quad (4)$$

where H_i is the header of the flow, and $P_i = \{p_{i1}, ..., p_{in}\}$ is a set of packets belonging to the flow.

Definition 2: A flow header H_i is described by the tuple,

$$H_i = (IP_{src}, IP_{dest}, port_{src}, port_{dest}, proto) \quad (5)$$

where IP_{src} and IP_{dest} are the IP source and destination addresses. $port_{src}$ and $port_{dest}$ are the source and destination transport port, respectively; and $proto$ is the transport protocol.

For a bidirectional flow, assuming that F_{src} is the source flow and F_{dst} is the destination flow, the definition above is extended in the sense that, F is bidirectional when $F_{scr} \cup F_{dst}$ and some elements of the header match. One of the main properties of the traffic flows is that they are ordered sequences of packets between the source and destination; this order is defined by the communication protocols.

2) Statistical based features: The features extracted from packet flows are mainly statistical based features, which are defined under the assumption that traffic at the network layer has statistical properties (such as the distribution of the flow duration, flow idle time, packet inter-arrival times and packet lengths) that are unique for certain type of applications, and enable different source applications to be distinguished from each other. Under this assumption, the work in [12] proposes 249 statistical features, which can be extracted from flow network traffic. Traffic characterization based on statistical features has been largely reported [13].

The properties such as inter-arrival time (IAT) and packets length are the most important characteristics considered, with their metrics, such as maximum, minimum, mean and standard deviation, among others. In practice, statistical features from flows are largely used due to its simplicity and non-intrusive property in the packet payload. Particularly, this last property is desirable due to the proliferation of encrypted communications. Additionally, in traffic analysis, the moving average is deployed to extract features from sessions communication.

III. Inner-class feature estimation proposal

This section shows the assumptions and algorithms defined to apply our proposal. Generally speaking, the main aim of the proposal is to estimate feature values through modeling the dynamic behavior of the original features, in particular the moving average. The main activities cover creating the feature models, and performing a FE with them. In order to create the feature models, several LR models will be trained with a selected labeled raw dataset, and only one model for class will generalize the feature value estimation. Following, in the FE process, the classical statistical features will be computed; moreover, the new features estimated by the LR models will be added. The new features will be computed through evaluating the current value by all the LR models, and the final feature value will be the one that obtains the minimum error.

To correctly apply this proposal, the following assumptions are set,

Assumption 1. We can differentiate a raw input, which will be interpreted as an ordered sequence of samples in a finite period of time, from another raw input.

Assumption 2. The statistical behavior of a variable is different from class to class, enabling to differentiate them from one another.

Assumption 3. Given the previous statements, it is possible to model statistical feature behaviors for each class separately.

Let $X = [X_1, X_2, ...X_n]$ be the raw data from a historical dataset with class labels $L = [l_1, l_2, ..., l_n]$, F the current statistical based features, and A the selected features to be modeled. In Table I is depicted the general algorithm. This algorithm starts selecting a subset of raw samples $\{X_s, L_s\}$ from the original raw data $\{X, L\}$. The next step is to select the features to be estimated A; this proposal chose the moving average. Following, the models M are created and a FE process is executed.

In the following sections, the main components of the proposal are detailed.

TABLE I: Pseudo code of the proposal

Macro algorithm
Input: X and F. Procedure: *% Proposal* 1. Select a subset X_s from the original raw data X 2. Select the feature set A to be estimated/modeled from F, *% Create the feature models* 2.1. Create the models M with X_s using the algorithm in Table II *% Feature extraction* 3. Perform the FE process with X and M using the algorithm in Table III End For Output: M

A. Create the feature models

The main aim is to find a model that can predict the current value of a feature, given the raw sample just measured and the past value of the feature. This proposal select a simple and classical strategy for modeling a feature behavior, i.e., a LR model. A LR model is created for each class c in L_s with the raw data $X^c \in X_s$. This model m^c is described as follows.

Definition 3: The LR of a raw sample $X_j^c \in X^c$ with $X_j^c = [x_{ji}, ..., x_{jh}]$ is defined as,

$$m_j^c = g(\Theta_j^c) \quad (6)$$

where Θ_j^c is a vector holding the parameters of LR model, and g the function that specifies the relationship among them. In this particular case, the type of LR model fixed will depend on the number of the historical and current raw features that will be used to make the prediction. In the present work, we propose to use the model of Eq. 7.

$$a_i^c = m^c([a_{i-1}, x_i]) = \theta_0^c + \theta_1^c * a_{i-1} + \theta_k^c * x_i \quad (7)$$

where $a \in A$ is the attribute targeted to be estimated and $x_i \in X_j^c$. Therefore, a_i^c is the current predicted value for the sample i by the model c.

This LR function is inspired by the moving average parameters, due to this case is selected to compare the proposal with. We can also compute the training error for further use, by using Eq. 8.

Definition 4: The training error of m_j^c will be measured through the mean squared error as follows,

$$e_j^c = \frac{1}{h} * \sum_{i=1}^{h} (f(x_{ji}^c) - a_j^c)^2 \tag{8}$$

where $f(x)$ is the feature computation using the classical statistical expression, interpreted as the ground truth; and h the amount of the raw values in a time series sample.

We can notice from the previous definitions that a LR model will be created for each time series sample in the raw dataset X^c. In order to find only one model for class, we use **Assumption 3** to assume that the model parameters should be alike because the raw samples belong to the same class. In this sense, we propose a weighted mean computation of the parameters Θ^c by using the following definition.

Definition 5: The final parameters for the class c with k samples are given by Eq. 9, where e_j^c in Eq. 8 will weight the impact of each parameter. This weight is scaled from 1 to 0, meaning that values of $\bar{e_j^c}$ close to 0 had a big error when building the LR model; the contrary case will occur if $\bar{e_j^c}$ is close to 1.

$$\Theta^c = \frac{1}{k} * \sum_{i=1}^{k} \bar{e_j^c} * \Theta_j^c \tag{9}$$

Table II shows the algorithm to create the feature model for all the class. The selected raw set X_s is divided into subsets by means of the unique class labels in L_s; i.e., the rows in X_s that belong to the class c are in $X^c = [X_1, ..., X_k]$, with k the amount of samples labeled with class c. This selection is performed in order to improve the inner-class differentiation based on **Assumption 2**; consequently, a LR model will be built for each class in the set X_s. The LR models are trained by means of the raw time series samples and their ground truth feature value (for this case, the mean of the series). Following, to obtain the parameters that generalize the model, the normalized error is used to weight the mean computation of Θ. At the end of this process, we have a model for each class in the raw data set; defined as $M = [m^1, ..., m^c]$. These models will feed the next process, in order to add one or more attributes a to the feature extraction process.

TABLE II: Pseudo code to build the metric model for the class c

Macro algorithm

Input: X_s and L_s.
Procedure:

% Feature model
1. Separate the raw dataset X_s into subsets X^c, using L_s
2. For X^j in $\mid X^c \mid$,
   2.1. For X_i^j in X^j,
      2.1.1. Train a regression model m_i with X_i^j and $f(X_i^j)$
      2.1.2. Compute the error e_i by using Definition 4
      2.1.3. Save the pair e_i and Θ_i in the sets E and Θ, respectively
   2.2. Normalize the errors in E
   2.3. Compute the parameters Θ^c through Definition 5, and obtain the final models M with the parameters.

Output: M

B. Feature extraction

In this step, we can extract the classical statistical based features F, such as the mean and variance, previously defined for the case study. Moreover, new attributes can be computed by estimating its value through the models in M.

The algorithm to perform such task is given in Table III. The FE process starts with the computation of the original statistical features. The estimation of the feature value will compute the output of all the models for class in M. The selection of the best feature value will be the one that originates the lowest distance between the prediction and the current value as follows.

Definition 6: The error of m_j will be measured through the Euclidean distance between the previous value computation and the current sample value,

$$d_j^c = [(a_j^c + w) - x_j]^2 \tag{10}$$

This error can be affected by shifts in the sample behavior. In this sense, a momentum w is added at this stage for balance the final decision. The momentum is given by,

$$w = (a_{j-1} - x_j)/t \tag{11}$$

where t is the number of samples currently evaluated. Our approach considers a shifting backwards with w for improving the current prediction.

Once each the estimated metric value is computed, evaluating each LR model, the selection of the model will use the following definition.

Definition 7: The best approximation to the attribute of interest a, is given by the LR model that obtains the lowest distance value, as follows.

$$a_j = \{a_j^i \in P \mid argmin(d_j^i)\} \tag{12}$$

IV. Experimental evaluation

In this section, we will start presenting the case study selected to test our proposal. Following, the experiments are described. Finally, the evaluation of our proposal.

A. Case study

As we previously discussed, the case study selected is the traffic analysis problem; where the main objective is to perform traffic classification for improving the QoS in satellite communications. We have used different datasets, which are:

- Captured: This data was captured by the authors to record internet traffic for particular applications. In this sense,

TABLE III: Pseudo code to extract the features from the raw data

Macro algorithm

Input: X, M, F.
Procedure:

% Feature extraction
1. For X_d in $\mid X \mid$,
  1.1. For x_j in $\mid X_d \mid$,
    1.1.1. Compute the original statistical features S by using the functions in F
    1.1.2. For m^i in M
      1.1.2.1. Compute $a_j^i = m^i([a_{j-1}, x_j])$ by using Equation 7
      1.1.2.2. Compute $d_i(a_j^i + w, x_j)$ by using Definition 6
      1.1.2.3. Add to the set $P(i) = (a_j^i, d_j^i)$
    1.1.3. Select the feature value estimation a_j by using Definition 7
    1.1.4. $a_{j-1} = a_j$
    1.1.5. Append to the data set $data(j) = [s_1, s_n, ..., a_j]$

Output: $data$

this dataset contains flow sequences of Youtube, Skype and web browsing application. These applications were launched from a personal computer, and all the flows of the opened sessions are captured and saved into a binary file.

- PAM : This dataset was developed by [20]. In real world scenarios in traffic network is very difficult to inspect end-to-end communications due to several aspects, mainly concerning to privacy matters. Therefore, the authors in [20] created an emulated environment that allows them to acquire complete flows from several end-to-end communications. They used 4 hardware machines, 2 with Windows 7 and 2 with Ubuntu, plus 3 virtual machines with Windows 7, Windows XP, and Ubuntu, as data generating stations. A server machine was used for data storage. VBS was used to collect the information about the flows, such as start time of the flow, number of packets contained by the flow, local and remote IP addresses, local and remote ports, transport layer protocol, along with detailed information about each packet.

Additionally, these datasets were processed by a DPI tool, in order to obtain the name of the application and the category of each flow. Among the categories identified are: Web protocols, File sharing (P2P), Social network, Streaming, network communication protocols, real-time communication (VOIP), System level applications, File transport protocol, VPNs and Protocols for database communication. Some categories are more relevant to detect for satellite resource managing to improve the QoS; for instance, streaming and the real-time communications . These last categories count with applications such as Netflix, Flash, YouTube, and Skype, among others. Table IV summaries the raw data characteristics.

TABLE IV: Features extracted from the packets flows.

Name	# of sequences	# of classes
Captured	3793	6
PAM	173429	17

B. Feature models and feature extraction

For both case studies, the settings in Table V were established for creating the models. In this particular proof of concept, a stream sequence is considered as a flow of packet lengths and inter-arrival times (IATs), due to they represent the raw data that can be extracted from communication sessions, and most of the statistical based features are computed over them in traffic analysis. Consequently, four features were selected for the flow directions str and $dest$ (source and destination, respectively). It is important to mention that, for labeling the flows, *nDPI* was used to obtain their categories. Four estimated means will be modeled; these means will represent the packet length and the IAT means for the directions str and $dest$. Therefore, the number of LR will be the number of classes multiply by the number of features to be modeled.

TABLE V: Features extracted from the packets flows.

Flow sequence property	**F**	**A**	$\{X_s, L_s\}$
Packet length IAT	mean std min max	mean	30% of the raw data, randomly selected

C. Evaluation

Fist at all, we measure the accuracy of a classifier using the original feature extraction process with a simple moving average, following with an exponetially weigthed moving average, and finally our proposal. This test will allow us to evaluate the macro performance of the new added features compared with the classical ones. Additionally, we will remark the micro recall values, in order to see how the inner-class dectection behaves for certain classes. The classifier selected was Random Forest (RF), which offered high rates of performance compared with other classical approaches [21]. Finally, in the evaluation proposed will count with some variations in the length of the sequence; consequently, we emulate a real scenario of incoming raw inputs.

We present the results of the proposal as follows, the accuracy of the selected classifier varying the amount of samples in the raw sequence in Table VI using the classical approach (Cl), exponentially weighted (Exp) with $\alpha = 0.1$, and our proposal (Prop) to compute the mean. Additionally, we show the recall metric of the three approaches for certain classes in the tables VII and VIII.

In Table VI, a sampling window (W) equal to 10 means that the number of packets taken to compute the metrics was 10 packets. The overall performance in Table VI of our approach is good and it shows a big improvement in the accuracy with respect to it counterpart when the sampling window is small (W=10 or W = 30). The early identification in traffic

analysis is very important and assures a fast action response. In this sense, these experiments offer a good outlook to the framework proposed. In general, the inner-class performance was improved.

V. Further analysis

In this section, we will try to interpret the feature estimation given by our proposal. In this sense, we take two flows where one flow belong to the class Streaming in Figure 1, and the other to the class VoIP in Figure 2. They are two particular classes that are normally hard to detect. The tables VII and VIII shows the goodness of our proposal overpassing the classical approaches with the variation of the raw data length in these flows. In this case, recall means how many of the true positives the model returns. Particularly, we obtained very good results for a small sampling window (W=10), which shows the early identification capabilities in traffic flows by our proposal.

From the figures, we can notice that the best method to estimate the mean of the packet length is given by the exponentially weighted. However, the question is why our approach give satisfactory results; in particular, when the sampling window is small. We can address this answer based on the design of the proposal. We know that the proposal estimate the mean of the sequence by selecting the most suitable LR model output, in the next iteration the same operation is performed with the same or a different LR model. Therefore, there is not memory about which LR model output was selected as the most suitable in the previous iteration. In this sense, our approach aim at creating a new feature value that tends to their class mean behavior, and not to the raw sample mean. It is for this reason, that in the figures, the proposal does not follow the raw mean value, and starts with trends that can change sharply according to LR model outputs.

From the traffic analysis point of view, in a standard streaming connection, the communication protocol starts with the client making a request and if the server agrees, this last one will send most of the session workload at its maximum capacity (packet lengths with 1500 bytes). Figure 1(a) and Figure 1(b) are reflecting this behavior. A different case occurs with VoIP session, where both parties can interact in the same way interchanging their roles, as we can notice in the figures 2(a) and 2(b).

VI. Conclusion

In this work, we present a novel approach to extract statistical based features that support the classical computation of the mean. We outlined some procedures to create an ensemble of LR models for each class under defined assumptions. Generally speaking, the workflow reached better results than two classical approaches; in particular, when the length of the raw sample is small. Additionally, inner-class discrimination was improved by our approach.

In future works, we are looking forward improving the selection of the LR models for class, as well as defining a more optimum procedure to build them. Additionally, modeling the error penalization will be considered by using the existing principles in the moving average such as the exponential weighted. Another important point, that we will address, is modeling another statistical based feature, such as the standard deviation, which at a first glance depends on the mean's LR model.

In satellite communications, correctly detecting some types of applications in the beginning of the communication is very important due to enable taking fast actions to guarantee QoS; thus, our results improve the inner-class early detection. Finally, we will test our approach over different databases from traffic analysis, and from other applications that fit into the assumptions proposed.

References

[1] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[2] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85 – 117, 2015.

[3] H. Wang, Y. Zhang, N. R. Waytowich, D. J. Krusienski, G. Zhou, J. Jin, X. Wang, and A. Cichocki, "Discriminative feature extraction via multivariate linear regression for ssvep-based bci," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 5, pp. 532–541, May 2016.

[4] P. Huang, T. Li, Z. Shu, G. Gao, G. Yang, and C. Qian, "Locality-regularized linear regression discriminant analysis for feature extraction," *Information Sciences*, vol. 429, pp. 164 – 176, 2018.

[5] Y. Chen and Z. Jin, "Feature extraction using class-oriented regression embedding," in *The First Asian Conference on Pattern Recognition*, Nov 2011, pp. 520–524.

[6] G. R. Arce, *Nonlinear Signal Processing*. John Wiley & Sons, Inc., 2005.

[7] M. Nakano, A. Takahashi, and S. Takahashi, "Generalized exponential moving average (ema) model with particle filtering and anomaly detection," *Expert Systems with Applications*, vol. 73, pp. 187 – 200, 2017.

[8] M. Menth and F. Hauser, "On moving averages, histograms and time-dependentrates for online measurement," in *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering*, ser. ICPE '17, 2017, pp. 103–114.

[9] G. E. P. Box and G. Jenkins, *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated, 1990.

[10] T. Finch, "Incremental calculation of weighted mean and variance," University of Cambridge Compuring Service, Tech. Rep., 2009.

[11] N. Bingham and B. Gashi, "Logarithmic moving averages," *Journal of Mathematical Analysis and Applications*, vol. 421, no. 2, pp. 1790 – 1802, 2015.

[12] A. Moore, M. Crogan, A. W. Moore, Q. Mary, D. Zuev, D. Zuev, and M. L. Crogan, "Discriminators for use in flow-based classification," University of London, Tech. Rep., 2005.

[13] J. J. Davis and A. J. Clark, "Data preprocessing for anomaly based network intrusion detection: A review," *Computers & Security*, vol. 30, no. 6, pp. 353 – 375, 2011.

[14] A. Marnerides, A. Schaeffer-Filho, and A. Mauthe, "Traffic anomaly diagnosis in internet backbone networks: A survey," *Computer Networks*, vol. 73, pp. 224 – 243, 2014.

[15] L. Peng, B. Yang, Y. Chen, and Z. Chen, "Effectiveness of statistical features for early stage internet traffic identification," *International Journal of Parallel Programming*, vol. 44, no. 1, pp. 181–197, 2016.

[16] M. Lotfollahi, R. Shirali, M. Jafari Siavoshani, and M. Saberian, "Deep Packet: A Novel Approach For Encrypted Traffic Classification Using Deep Learning," *ArXiv e-prints*, Sep. 2017.

[17] J. M. Lucas, M. S. Saccucci, R. V. Baxley, Jr., W. H. Woodall, H. D. Maragh, F. W. Faltin, G. J. Hahn, W. T. Tucker, J. S. Hunter, J. F. MacGregor, and T. J. Harris, "Exponentially weighted moving average control schemes: Properties and enhancements," *Technometrics*, vol. 32, no. 1, pp. 1–29, Jan. 1990.

[18] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer, 2009.

TABLE VI: Accuracy of the RF classifier varying the number of samples.

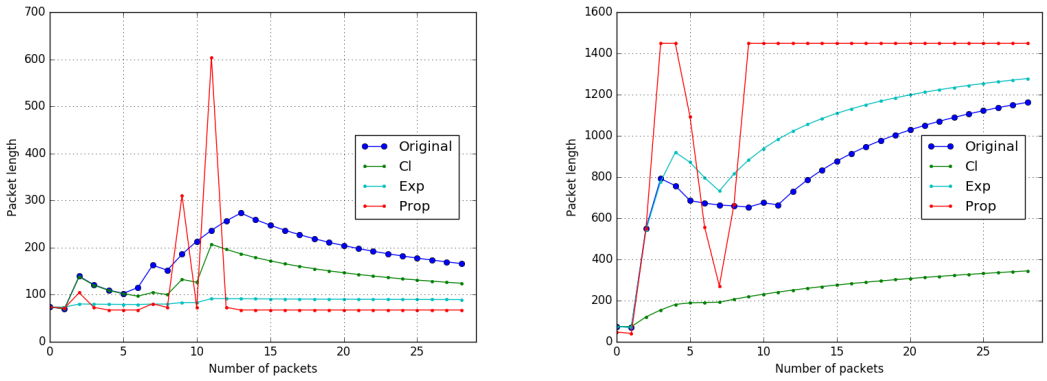
	W = 10			W = 30			Complete sequence		
	Cl	Exp.	Prop	Cl	Exp.	Prop	Cl	Exp	Prop
Captured	0.7859	0.7747	0.9728	0.8067	0.8083	0.9640	0.8027	0.9664	0.9624
PAM	0.4107	0.5300	0.9464	0.7209	0.8123	0.9450	0.9426	0.9511	0.9508

TABLE VII: Recall of the capture data for two classes.

	W = 10			W = 30			Complete raw sample		
Class	Cl	Exp.	Prop	Cl	Exp.	Prop	Cl	Exp.	Prop
Streaming	0.15	0.16	0.94	0.32	0.85	0.94	0.96	0.96	0.97
VoIP	0.17	0.37	0.85	0.83	0.17	0.87	0.94	0.94	0.94

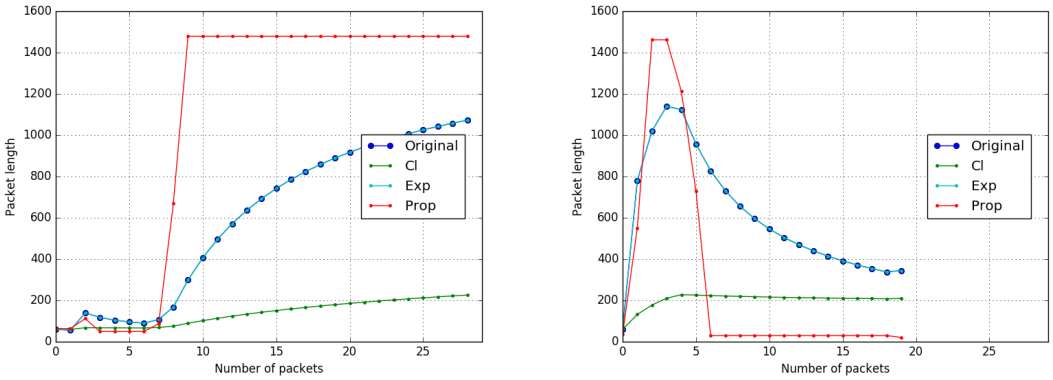
TABLE VIII: Recall of the PAM data for two classes.

	W = 10			W = 30			Complete sequence		
Class	Cl	Exp.	Prop	Cl	Exp.	Prop	Cl	Exp.	Prop
Streaming	0.03	0.42	0.65	0.04	0.48	0.63	0.64	0.66	0.66
VoIP	0.45	0.47	0.47	0.45	0.47	0.47	0.46	0.47	0.48



(a) (b)

Fig. 1: (a) Statistical feature values computed for a packet sequence in the client with the Streaming class. (b) Statistical feature values computed for a packet sequence in the server with the Streaming class.



(a) (b)

Fig. 2: (a) Statistical feature values computed for a packet sequence in the client with the VoIP class. (b) Statistical feature values computed for a packet sequence in the client with the VoIP class.

[19] B. Claise, B. Trammell, and P. Aitken, "Specification of the ip flow information export (ipfix) protocol for the exchange of flow information," Internet Engineering Task Force (IETF), Tech. Rep., 2013.
[20] T. Bujlow, V. Carela-Español, and P. Barlet-Ros, "Independent comparison of popular DPI tools for traffic classification," *Computer Networks*, vol. 76, pp. 75 – 89, 2015.
[21] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *Journal of Machine Learning Research*, vol. 15, pp. 3133–3181, 2014.