IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 25, 2020, accepted November 30, 2020, date of publication December 3, 2020,
date of current version December 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3042382

Regression Based Dynamic Elephant Flow Detection in Airborne Network

PENGFEI LIU[1], NA LV[1], KEFAN CHEN[2], LIANG TANG[3], AND JIAXIN ZHOU[1]
[1]School of Information and Navigation, Air Force Engineering University, Xi'an 710077, China
[2]People's Liberation Army (PLA) 94860, China
[3]People's Liberation Army (PLA) 61932, China

Corresponding author: Na Lv (de_mer@163.com)

This work was supported by the National Natural Science Foundation of China Youth Program under Grant 61703427.

ABSTRACT As an important part of the spatial information network, airborne network (AN), which connects air platforms with upper satellites and ground devices, has been increasingly important right now. Due to the heavy-tailed distribution of network traffic, elephant flow detection is usually used to catch and control the key part of network traffic with low costs, which is a practical strategy to strengthen network management and improve network performance. In this paper, we consider the problem of dynamic threshold elephant flow detection in AN, and an intelligent method based on regression with pre-classification is proposed to adapt to the limited and dynamically changing bandwidth. The filtering mechanism with waiting-window is used firstly to filter out parts of small flows to decrease the detection cost. Then, the pre-classification is used to divide the range to be predicted and the flow size regression can be carried out in a compressed range, which makes the results more accurate. Finally, the predicted size is compared with the specific detection threshold related to the specific moment, and the elephant flow is identified. Numerical experiments demonstrate that the proposed method has a better adaptability to dynamic threshold and the performance is much better.

INDEX TERMS Elephant flow detection, dynamic threshold, machine learning, regression learning, airborne network.

I. INTRODUCTION

With the rapid development of information technology, the modes and forms of communication have been further evolved. An intelligent and interconnected spatial information network is coming. As an important part of the spatial information network, airborne network (AN), which connects air platforms with upper satellites and ground devices, has been increasingly important right now. Increasing and various services are being or will be transmitted over AN. In the civil field, AN can provide a convenient air access to the Internet, which can effectively cover the blind areas of ground wired network and further expand the range of communications [1], [2]. In the military field, AN can be used to link the air and ground combat platforms, which can realize a fast information sharing among all the combat platforms and establish an efficient cooperation between different combat platforms across different regions [3], [4]. Due to the heavy-tailed distribution of network traffic [5]–[8], a small number of elephant flows, such as video streaming in the civil field and surveillance and sensing message in the military field, contribute a significant amount of the traffic volume, which will occupy a large amount of the limited available bandwidth. If those elephant flows fail to be detected and all flows are treated equally without any difference, some elephant flows may converge on the same link, resulting in link congestion and message loss, while some links may be idle or with little flows, resulting in a waste of available bandwidth. Either link congestion or idle will greatly reduce the efficiency of information exchanging, resulting in poor network performance and user experience. Timely and accurate elephant flows detection has become an efficient and practical strategy to optimize network performance [9], [10]. Different from the traditional wired network, the electromagnetic environment in AN is more complex. Noise, interference and attenuation caused by meteorological or artificial factors are ubiquitous all the time. Coupled with the movement of the platforms and the directivity of antennas, communication connections

The associate editor coordinating the review of this manuscript and approving it for publication was Inês Domingues.

will be more vulnerable, the network topology and available bandwidth will also be dynamically changing. Moreover, owing to the burstiness and dynamics of network traffic, the numbers, types and transmission data volumes of carrying services and their QoS are also dynamically changing in AN. In this context, it is much meaningful to detect the elephant flows quickly, flexibly and dynamically, thus differentiated services can be provided under limited resources.

In traditional wired networks, elephant flow detection is mainly realized by counting [11], [12], sampling [13], [14] or LRU (Least Recently Used) [15], [16] queue management. The main idea is to compare the volume of passed data with the corresponding detection threshold or try to filter small flows and keep up elephant flows by entries update-exit mechanism in the queue. In literature [11], the strategies of counting and LRU are adopted, and an asymptotically optimal algorithm is proposed. In the algorithm, the storage is divided into active and inactive areas. Entries of flows are updated in the active area and small flows are removed from the inactive area. Before the capacity of active area reaches to the specific level, the areas of active and inactive are exchanged. By doing so, an accurate detection is achieved with lower storage. This method dose well in some traditional wired networks, but it may be not suitable for the real-time and dynamic scene, like AN. There are so many data need to be processed and the detection delay is relatively large.

With the introduction of artificial intelligence technology [16]–[19], the traditional strategy, which uses posterior statistics to detect the elephant flow, has changed. The historical traffic data are used for training with machine learning algorithms, and a classifier can be built to mine the mapping relationship between traffic class and early features of traffic flow [20], [21]. With the help of the classifier, elephant flow detection can be achieved in a short time with fewer data. Since elephant flow undetected is more serious than small flow mis-detected, literature [17] sets different costs for different kinds of misclassification, and builds a classification model based on the cost-sensitive decision tree to obtain a more accurate detection result. In literature [18], data mining technologies are applied to elephant flow detection under the framework of SDN, and a detection model with two-phase is suggested. In the first phase, a classifier is built on the switch with some features that could be easily obtained, and only suspected elephant flows are submitted to the controller for further confirmation. In the second phase, another classifier is constructed on the controller with more features extracted from the first few packets of traffic flow, and the suspected elephant flows are identified once again. After these two phases, a more accurate detection can be achieved with lower detection cost. Both the methods in [17], [18] are based on supervised learning with binary classification. Specifically, the training data are labeled with two classes according to the specific fixed threshold in advance. Then, the labeled dataset is used to train a binary classifier, which is applied to test new arrival data or detect new arrival elephant flow. During the process above, the threshold for elephant flow is fixed and unchangeable, which is only available for the scenes where the properties of traffic flows basically remain unchanged and the communication bandwidth keeps stable. Unlike the stable scenes, carrying traffic and available bandwidth in AN are dynamically changing with the missions, phases and communication environments. The fixed threshold is unavailable, and it is urgent to use the dynamic threshold to adapt the changing of bandwidth or other QoS constraints.

To solve the problem of dynamic threshold elephant flow detection in AN, we propose a regression method to adapt to the dynamically changing threshold. Firstly, a regression model is built to describe the relationship between the early features and the total sizes of traffic flows. Then, the size of the new arrival flow can be predicted by the regression model with the early features. By comparing the predicted flow size with the specific threshold, the elephant flow can be identified. In the proposed method, the filtering mechanism with waiting-window is used to eliminate parts of small flows and alleviate the problem of data imbalance in regression. And the strategy of pre-classification is adopted to compress the range of flow sizes to be predicted, and the accurate results can be got more easily.

The rest of this paper is structured as follows. Section II presents the models and assumptions related to our work. Section III describes the details of the proposed method. Extensive numerical experiments are presented in Section IV, and Section V concludes this article.

II. ELEPHANT FLOW DETECTION MODEL WITH DYNAMIC THRESHOLD

In most existing literatures [15], [22]–[25], elephant flow is often defined as the flow in which the number of packets or bytes carrying is greater than a certain value or a certain ratio of total traffic passed through the link. That is:

$$f \in F_{ele} | Fs_{total}(f) \geq Tr_c \tag{1}$$

where, f is the flow to be detected, F_{ele} is the set of elephant flows, $Fs_{total}(f)$ is the total size of flow f, and Tr_c is the fixed threshold to determine elephant flows.

The definition with fixed threshold [15], [22]–[24] is used and appropriate for stable wired networks, where the bandwidth and the distribution of carrying traffic are almost unchanged. However, for AN, the stable scene has changed and the fixed threshold is no longer suitable. The threshold should be a dynamically adjustable value, which can be used to adapt to the changing of the available bandwidth and carrying traffic. Besides, the definition above is defined from the perspective of the total size of the flow, including the data that has passed and the data that is coming. But in most cases, the volume of data that is coming is unknown at the detection moment, so is the total size of the flow. And only when the flow ends, the total size can be obtained. Therefore, this definition is much suitable for a post-event network traffic analysis. Sometimes, the volume of data passed before the detection is also used to approximately evaluate the total size of the flow, and the result can be used for a real-time traffic

scheduling. But this approximation will not probably work well due to the unknown of coming data, which are exactly the data that needs to be scheduled. Actually, by introducing the artificial intelligence, it is possible to learn from the historical data and make a prediction of flow size before the flow ends. In this case, the passed data are used to extract specific features, which are sent to the intelligent model, learned from the historical data, to discriminate traffic flows or predict the flow size. Then, the volume of the coming data can be obtained from the predicted flow size and the volume of passed data. Obviously, the coming data has great influence on the coming status of network, and should receive much more focus in real-time traffic scheduling of AN. Based on the above analysis, we modify the definition as follow:

$$f \in F_{ele} | Fs_{total}(f) - Fs_v(f) \geq Tr_v \tag{2}$$

where, $Fs_v(f)$ is the volume of data used for prediction, and Tr_v is the detection threshold that can be dynamically changed.

It is worth noting that, in this paper, small flows serve as a complement to elephant flows, and then small flows can be identified with the same threshold.

$$f \in F_{mice} | Fs_{total}(f) - Fs_v(f) < Tr_v \tag{3}$$

Different from (1), the volume of data used for prediction $Fs_v(f)$ and the dynamic threshold Tr_v are taken into consideration in (2). If both $Fs_v(f)$ and Tr_v are constants, that is, the volume of data used for prediction and the detection threshold of elephant flow are fixed. Then equation (2) can be simplified to (1), where Tr_c is substituted by the sum of $Fs_v(f)$ and Tr_v. In this case, elephant flows and small flows can be identified only according to the relationship between the flow size $Fs_{total}(f)$ and the constant threshold of the sum of $Fs_v(f)$ and Tr_v. Thus, the historical data flows, with known flow sizes, can be labeled according to the fixed threshold, and a binary labeled training dataset can be obtained. And after establishing the connections between the features and the classes of data flows, a binary classifier can be trained from the labeled training dataset to detect the elephant flow easily.

According to (2), the classes of data flows are still related to the flow size, the volume of data used for prediction and the dynamic threshold. However, for AN, the detection threshold Tr_v is a variable, and the volume of data used for prediction $Fs_v(f)$ may also change. Thus, the relationships between the features and the classes of historical flows are inconsistent and changing, and the binary labeled training dataset with the uniform threshold may no longer be applicable. In this case, a static binary classifier is incompetent, and a regression model with dynamic threshold is needed. Different from the binary classification, the regression model does not need the labeling of historical flows. It is constructed only based on the features and the sizes of the historical data flows, which is no difference with the dynamic detection threshold. For a new arrival data flow, the predicted flow size can be obtained from the regression model inputted with the features extracted from the data passed. The dynamic threshold elephant flow detection is achieved by comparing the predicted flow size and the dynamic threshold.

Suppose D_n is the training dataset, which contains n samples $(\boldsymbol{x}_i, y_i), i = 1, \cdots, n$, where $\boldsymbol{x}_i=(x_{i1}, x_{i2}, \cdots, x_{im})$ is the features of the ith sample in the dataset, x_{im} is the mth dimension feature of $\boldsymbol{x}_i$, and y_i is the corresponding flow size. For the fixed detection threshold, as the detection threshold is fixed and knowable, the sample $(\boldsymbol{x}_i, y_i)$ can be labeled with the formula:

$$l_i = \begin{cases} l_{ele}, & y_i \geq Tr_c \\ l_{mice}, & y_i < Tr_c \end{cases} \tag{4}$$

where, y_i is the size of the flow and Tr_c is the detection threshold.

After the labeling, the binary labeled training dataset $(\boldsymbol{x}_i, l_i), i = 1, \cdots, n$ can be obtained, where l_i is either l_{ele} or l_{mice}. Based on the binary labeled training dataset, a mapping or a binary classifier $M_C : \boldsymbol{x} \rightarrow \{l_{ele}, l_{mice}\}$ can be obtained with a machine learning strategy. When a new flow f_* arrivals, the corresponding features x_* are sent to the model M_C, and the class of the flow can be directly obtained by $l_* = M_C(\boldsymbol{x}_*)$

While, for the dynamic threshold detection, as the detection threshold is dynamic changing, then the samples cannot be labeled with a threshold. In this case, the sizes of flows are regarded as labels. Regression learning is directly taken on this consistent label dataset, and a regression model or predictor $M_R : \boldsymbol{x} \rightarrow y$ is used to predict the flow size. When a new flow f_* arrivals, the corresponding features $\boldsymbol{x}_*$ are sent to the model M_R, the predicted flow size can be obtained by $y_* = M_R(\boldsymbol{x}_*)$. By substituting the detection threshold Tr_v and the predicted flow size y_* into (2), the class of the flow can be determined. The processes of detection with fixed threshold and dynamic threshold are shown in Fig. 1.

As can be seen in Fig. 1, the difference between the fixed threshold detection and the dynamic threshold detection lies in the labeling. In the fixed threshold detection, the classification is adopted, and the labeling is placed before training. While, in the dynamic threshold detection, the regression is adopted, and the labeling is placed after the prediction of flow size. It is obvious that the results of elephant flow detection with dynamic threshold are seriously influenced by the results of regression prediction. Therefore, the key of the dynamic elephant flow detection proposed in this paper is the regression for the flow size.

III. FLOW SIZE REGRESSION WITH PRE-CLASSIFICATION

Researches [26]–[28] show that the flow size of network traffic is distributed in a wide range, and the distribution is usually imbalanced. It is difficult to do the regression learning on the original training dataset. In order to reduce the difficulty and improve the accuracy, here we introduce a strategy of pre-classification for the flow size regression. Before the regression learning, a filtering mechanism with waiting-window is firstly used to filter out parts of

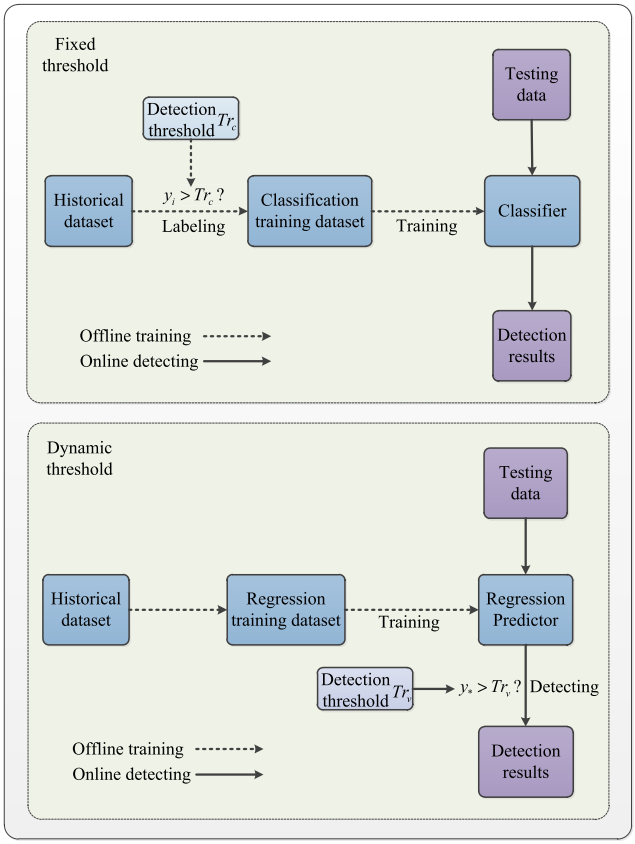


FIGURE 1. The processes of elephant flow detections with fixed and dynamic threshold.

small flows, which can compress the prediction range and alleviate the phenomenon of imbalance. Since fewer samples need to be further processed, the detection cost will decrease. Then, the pre-classification method is adopted to divide the range of flow size to be predicted. Classifiers are trained on the dataset labeled with dividing borders and regression predictors are trained on the divided dataset. Thus, the regression of flow size can be carried out in a compressed range and implemented much easier. After the regression, the predicted flow size is compared with the specific detection threshold related to the specific communication condition to detect the elephant flows. The entire process of flow size regression with pre-classification is shown in Fig. 2.

As can be seen in Fig. 2, the entire process of flow size regression with pre-classification consists of the offline training part and the online detecting part. In the training stage, waiting-window filtering mechanism is used to screen out available training samples, and pre-classification is used to pre-train the standby classifiers and regression predictors. While, in the testing stage, waiting-window filtering mechanism is used to filter out and detect parts of small flows, and pre-classification is used to select specific classifiers or regression predictors for elephant flow detection.

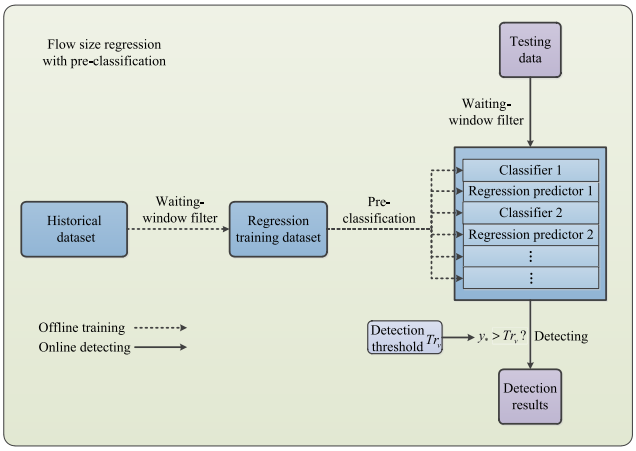


FIGURE 2. The process of flow size regression with pre-classification.

A. WAITING-WINDOW FILTERING

Due to the heavy-tailed distribution in network traffic, there are many small flows either in the training dataset or the testing data. Although the volume of carrying data is small, the number of small flows is huge. In the training stage, the huge number of small flows will lead to the sample imbalance of the training dataset, which will seriously affect the preferences of the prediction model to be generated. In the stage of prediction, since the number of packets used for feature extraction is very limited, it is almost impossible to predict the flow size of such small flows. These unpredictable small flows will lead to a lot of unnecessary prediction overhead. Even if we can make an accurate prediction at great cost, it is also not cost-effective to control the remaining data.

In order to reduce the negative impact of such small flows, a filtering mechanism with waiting-window is adopted. According to the property of fewer packets and relatively lower packets frequencies, parts of small flows can be eliminated with a time stack, together with the feature extraction. By setting a waiting-window, small flows that do not meet the packet number required for feature extraction within the specific time are eliminated, and potential elephant flows represented by the features are retained. In the training stage, this filtering mechanism can be used to alleviate the imbalance of the training dataset used for classification or regression, and then a relatively balanced modified training dataset can be obtained. In the detecting stage, it can be used to reduce the number of flows that need to be further processed by classification or regression, thus improving detection efficiency and reducing detection cost. The process of waiting-window filtering is shown in Fig. 3.

As can be seen in Fig. 3, in the waiting-window, packets of flows are collected until a sufficient number is satisfied. If the number of packets is sufficient within the waiting-window, the flow is retained, and the collected packets are sent for feature extraction. Otherwise, the flow is discarded.

In the module of feature extraction, the desired features to represent the flow are extracted based on the collected packets. Usually, the header information and statistical parameters

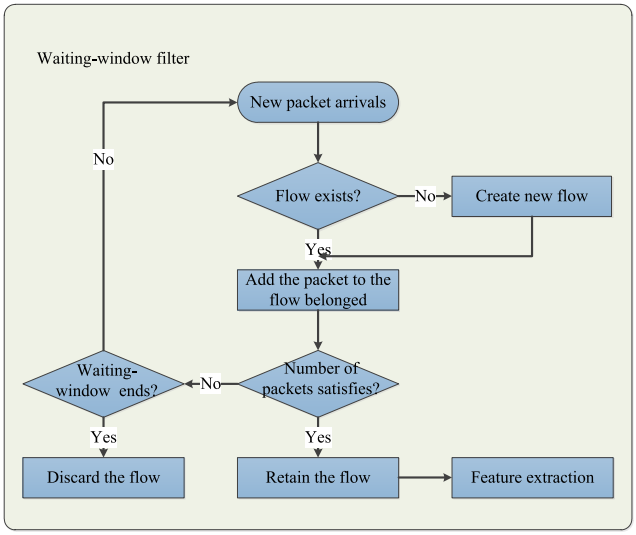


FIGURE 3. The process of waiting-window filtering.

of packets are adopted as features to deal with the encryption technology. These features can be obtained with Network data analysis tools and numerical calculation tools.

After the process of waiting-window filtering, features and the size of the flow are saved as a sample in the training dataset.

B. PRE-CLASSIFICATION

After the preprocessing of the waiting-window filtering, the number of small flows in the modified training dataset is greatly reduced, which alleviates the phenomenon of imbalance. However, as the flow size is distributed in a relatively large range, it is still difficult to make a regression prediction in the entire range. In order to further reduce the regression difficulty and improve the accuracy of prediction, we divide the range of flow size into several small ranges by means of pre-classification, and further compress the range that needs to be predicted.

1) DIVISION OF PREDICTION RANGE

Before dividing the range of flow sizes to be predicted, the concepts about the ranges of the flow sizes are necessary to be clarified. One is the range of flow sizes of the training dataset, covering the minimum and maximum flow sizes of the samples in the training dataset; the other is the range of the dynamic thresholds, used to detect the elephant flows, covering the minimum and maximum flow size to be further processed. Intuitively, the range of dynamic threshold is more desirable than the flow size range of the training dataset. But it is closely related to the changing of bandwidth and carrying traffic, and cannot be known in advance. Therefore, we have to settle for the second best, i.e., selecting the range of flow sizes of the training dataset.

For the range division, two important parameters need to be determined. One parameter is the number of the divided sub-ranges, which corresponds to the number of classes that need to be classified with pre-classification. Since the number of the classes to be classified increases with the number of sub-ranges, the finer the classification granularity the smaller the range. When the classification granularity is fine enough, the results of multiple classifications can even be regarded as the prediction value. But it is worth noting that the finer granularity leads to the higher cost. The other one is the specific division thresholds. These thresholds are determined based on the specific distribution of the training dataset on the premise of giving the sub-range number. The most simple and convenient method to determine the division thresholds is to divide the range of flow size or the sample number of the training dataset by isometric division.

In this paper, the entire range of the flow sizes of training dataset is not the target to be predicted, and we only select a subset of the dataset for the prediction. Here we select the 90th percentile and 99th percentile of the flow sizes in the training dataset as the lower and upper limits that need to be further processed, that is, only 1% to 10% of the data flows in the training dataset will be considered in the prediction model. Usually, flows over the 99th percentile are treated as elephant flows, and flows under the 90th percentile are treated as small flows. From the property of heavy-tailed distribution, it can be seen that even if only one-tenth or even one-hundredth of the flows at the top of the distribution are predicted and further processed, the actual volume of traffic packets is still considerable. After determining the range of flow size to be predicted, the method of equal quantity division is adopted to determine the division thresholds, and thus avoid the class imbalance between different ranges. In order to avoid unnecessary division of ranges caused by too small interval between percentiles, the minimum division interval is set in advance to reduce classes of classification and simplify the complexity of pre-classification.

2) MULTI-CLASS CLASSIFIER AND CLASSIFICATION

After dividing the training dataset into sub-ranges or classes, a multi-class classifier can be obtained by means of training or learning. Usually, the multi-class classifier can be achieved directly from a multi-class training, or obtained by the combination of multiple binary classifiers. Due to the mature skills and methods of feature selection and data preprocessing in binary classifier, a good binary classification is relatively easy to obtain. Therefore, we combine multiple binary classifiers to achieve the multi-class classifier, and compress the prediction range. In order to reduce the complexity of pre-classification, we choose the decision tree C4.5, which is simple and fast, as the basic classifier. The performance of this algorithm has been verified in many network traffic classification studies [29]–[31]. In accordance with the aforesaid method of prediction range division, lots of classification training datasets, labeled by the division thresholds, can be obtained, and a number of decision trees can be trained based on the training datasets. By combining decision trees of adjacent division thresholds, a multi-class classifier and prediction range compression can be achieved.

For example, suppose that the dynamic detection threshold of a new arrival flow is Tr_v. If the classifier of the detection threshold Tr_v belongs to the standby classifiers trained in advance, the new arrival flow can be classified and detected directly with the corresponding binary classifier. Otherwise, we can combine the classifiers of division thresholds Tr_i and Tr_{i+1}, where Tr_i and Tr_{i+1} are nearest to Tr_v and satisfy $Tr_i < Tr_v < Tr_{i+1}$.

If the new arrival flow is classified as a small flow by the classifier Tr_i, that means:

$$Fs_{total}(f) - Fs_v(f) < Tr_i \tag{5}$$

then

$$Fs_{total}(f) - Fs_v(f) < Tr_v \tag{6}$$

For the detection threshold Tr_v, the new arrival flow is still a small flow.

If the new arrival data flow is classified as an elephant flow by the classifier Tr_{i+1}, that means:

$$Fs_{total}(f) - Fs_v(f) \geq Tr_{i+1} \tag{7}$$

then

$$Fs_{total}(f) - Fs_v(f) \geq Tr_v \tag{8}$$

For the detection threshold Tr_v, the new arrival flow is still an elephant flow.

Besides, if the new arrival flow is classified as an elephant flow by the classifier Tr_i and classified as a small flow by the classifier Tr_{i+1}, that means:

$$Tr_i \leq Fs_{total}(f) - Fs_v(f) < Tr_{i+1} \tag{9}$$

Although a further regression processing is still needed to detect the elephant flow, the range of flow size to be predicted has been compressed, in other words, the sub-range has been obtained by the pre-classification. The entire process of pre-classification is shown in Fig. 4.

As can be seen in Fig. 4, in the training stage, the range of flow size is divided into several small ranges. Based on the dividing borders, many binary classification training datasets can be labeled, and binary classifiers are trained on the datasets to discriminate different sub-ranges. At the same time, a lot of regression predictors are trained on the divided datasets within the sub-ranges. In the detecting stage, the dynamic detection threshold of elephant flow is given, and the two classifiers, whose dividing borders are closest to the detection threshold, are selected. The results of classifiers are combined to determine whether a regression predictor is further needed to detect the elephant flow.

C. REGRESSION PREDICTION

With the processing of waiting-window filtering and pre-classification, the classes of some flows have been identified. For the ones that are not identified yet, the flow sizes to be predicted have also been compressed by the pre-classification. Thus, regression prediction can be carried out in the compressed range for further identification. Different from regression on the entire range of flow size, the complexity of regression on the compressed range has greatly decreased, and a more accurate prediction can be achieved. Currently, there are many algorithms available for regression prediction. Any algorithm with excellent performance can be adopted here.

In this paper, the Gaussian process regression [32], [33] is selected for the flow size prediction. This algorithm can be easily implemented and has strong generalization ability. The prediction of flow size in the Gaussian process regression is treated as a part of Gaussian process, in which any number of outputs is assumed to be consistent with the joint Gaussian distribution. Suppose D_n is the training dataset, which contains n samples $(\boldsymbol{x}_i, y_i)$, $i = 1, \cdots, n$, where $\boldsymbol{x}_i$ is the features of the ith sample in the dataset, and y_i is the corresponding flow size. Let $\boldsymbol{X}$ be the matrix composed of all $\boldsymbol{x}_i$, and $\boldsymbol{y}$ be the vector composed of all y_i, then the training dataset D_n can be expressed as $(\boldsymbol{X}, \boldsymbol{y})$. For a new arrival flow $f_*, \boldsymbol{x}_*$ represents the input features, and the output flow size y_* satisfies:

$$\begin{pmatrix} \boldsymbol{y} \\ y_* \end{pmatrix} \sim N\left(0, \begin{bmatrix} K(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 \boldsymbol{I} & K(\boldsymbol{X}, \boldsymbol{x}_*) \\ K(\boldsymbol{x}_*, \boldsymbol{X}) & K(\boldsymbol{x}_*, \boldsymbol{x}_*) \end{bmatrix}\right) \tag{10}$$

where, $N(\cdot)$ represents the joint Gaussian distribution, $K(\cdot, \cdot)$ represents the covariance matrix between the input vectors, and σ_n^2 is the noise variance.

From (10), the posterior probability density function of the flow size of the new arrival flow can be obtained as follows:

$$y_* | \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y} \sim N(\mu, \sigma^2) \tag{11}$$

where

$$\mu = K(\boldsymbol{x}_*, \boldsymbol{X}) \cdot (K(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 \boldsymbol{I})^{-1} \cdot \boldsymbol{y} \tag{12}$$

$$\sigma^2 = K(\boldsymbol{x}_*, \boldsymbol{x}_*) - K(\boldsymbol{x}_*, \boldsymbol{X}) \cdot (K(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 \boldsymbol{I})^{-1} \cdot K(\boldsymbol{X}, \boldsymbol{x}_*) \tag{13}$$

Since the probability density function of the Gaussian distribution is symmetric about the mean μ and has the greatest probability at the mean μ, the mean in (12) is generally regarded as an estimate of the flow size y_*.

IV. NUMERICAL EXPERIMENTS

A. DATASET AND SETTINGS

In order to verify the performance of proposed method for the dynamic threshold elephant flow detection in airborne network, an evaluation dataset from an airborne network is needed. However, currently, there is no public and available airborne network traffic datasets, and traffic generation and emulation for airborne network have not been fully studied. Fortunately, the focus of this paper is on the problem of elephant flow dynamic detection, and the phenomenon of heavy-tailed distribution is universal for both airborne network and ground wired computer network. In this paper, a modified dataset, which is modified from the UNIBS-2009 dataset [34], is adopted for the numerical experiments. The original traces of UNIBS-2009 are collected on the edge

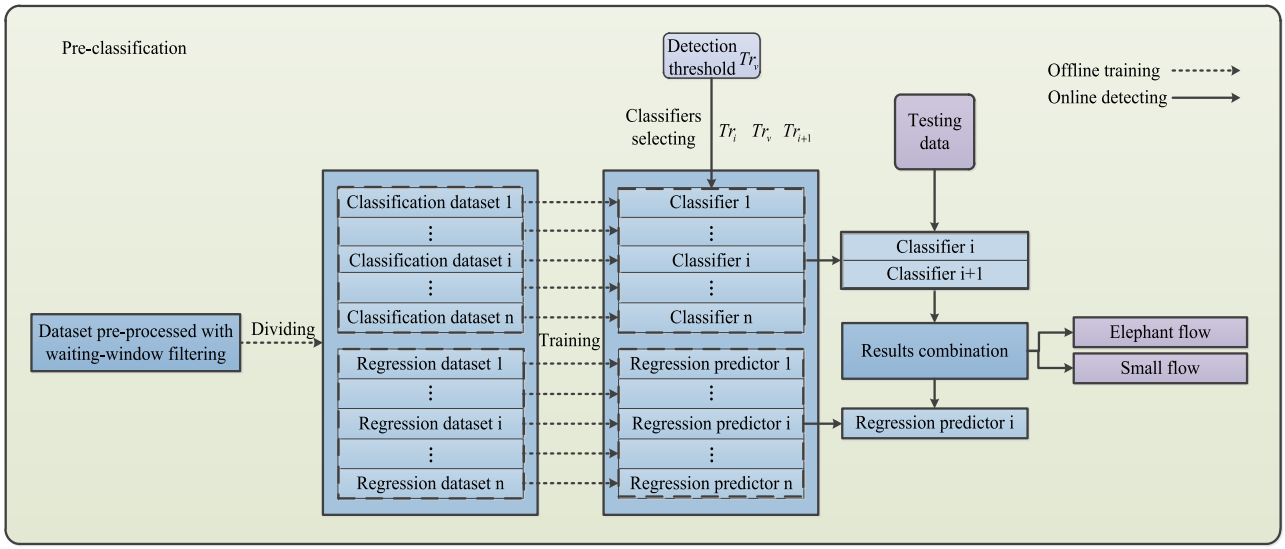


FIGURE 4. The process of pre-classification.

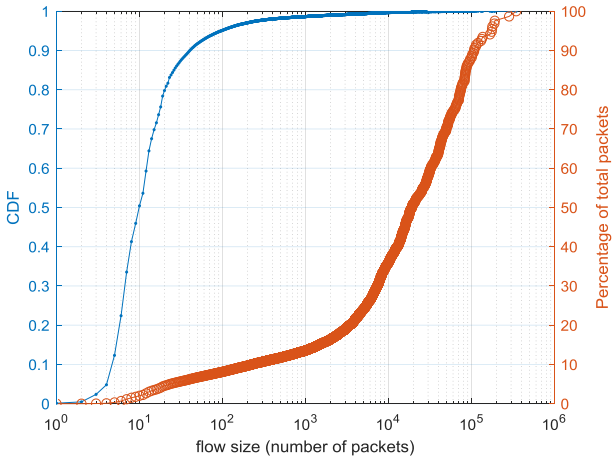


FIGURE 5. Distribution of flow size and percentage of total packets over flow size. Left axis: CDF; Right axis: percentage of total packets.

router of the campus network of the University of Brescia on three consecutive working days, generated by a set of twenty workstations. The traffic includes Web, Mail, Skype, traffic generated by Peer-to-Peer applications and other protocols, around 79000 conversations in total. For airborne network, most services [35]–[38] are very similar to the UNIBS-2009, such as, instant messaging, voice communication, transmission and sharing of pictures and radios, and so on. Therefore, similar traffic flows are selected into the modified dataset to evaluate the proposed method. 26114316 data packets and 154714 flows are included. The cumulative distribution function of flows and packets over the flow size are shown in the Fig. 5.

It can be seen from Fig. 5 that flow sizes in the selected dataset is widely distributed, ranging from a few packets to millions of packets. The numbers of flows with different sizes vary greatly. The numbers of small flows, containing fewer packets, are very huge, while the numbers of elephant flows, containing more packets, are very small. The phenomenon of imbalance and the characteristics of heavy-tailed distribution are very obvious.

In this paper, 75% of the samples in the dataset are randomly selected as the training dataset, and the remaining 25% are the testing dataset. To achieve the early detection of elephant flow, the first ten packets of the data flow are used for feature extraction. Here we extract some parameters related to the first ten packets as features, including the packet size, inter-arrival time (IAT), and statistics of packet size and IAT. Additionally, the source port, destination port, protocol type and duration of first ten packets are also extracted. Among the desired features, source port, destination port, and protocol type are extracted from the header of the first packet. The size and inter-arrival time (IAT) of the first ten packets are extracted from the header and timestamp of each packet. The statistics of size and IAT of the first ten packets are obtained based on the size and inter-arrival time (IAT) of the first ten packets. The duration of the first ten packets is also extracted from the timestamp of the first ten packets. As the basis for these features, both the header and the timestamp of packets are extracted with the tool Wireshark, and statistics and numerical calculations are conducted with MATLAB. The numerical experiments are running on a DELL XPS8930 with an Intel i7-8700 3.2 GHz CPU and 16GB RAM. Weka 3.8.4 and MATLAB 2018b are used as software frameworks, which are running on Windows 10 64-bit OS. The Weka is used for feature selection for pre-classification, and the MATLAB is used for regression prediction with its own Regression Learner tool. The results of the feature selection in adjacent binary classifiers are used for regression. It is assumed that the dynamic threshold obeys the normal distribution and changes every 20 samples. The simulation is repeated 20 times and the average result can be obtained.

B. PERFORMANCE EVALUATION

In this paper, precision, recall and f-score are used to evaluate the performance of dynamic elephant flow detection. They

are defined as follows:

$$precision = \frac{TP}{TP + FP} \tag{14}$$

$$recall = \frac{TP}{TP + FN} \tag{15}$$

$$f - score = \frac{2 \times precision \times recall}{precision + recall} \tag{16}$$

where, TP represents the positive examples correctly classified, FP represents the negative examples incorrectly classified as positive examples, and FN represents the positive examples incorrectly classified as negative examples. Therefore, the precision represents the ratio of the true elephant flows within the elephant flows detected, while the recall represents the ratio of the true elephant flows detected within the entire true elephant flows. F-score is the harmonic mean of precision and recall.

C. RESULTS AND ANALYSIS

1) PERFORMANCE EVALUATION OF REGRESSION WITH PRE-CLASSIFICATION

To evaluate the performance of the proposed method, we compare the performance of the existing binary classifier (C1), multi-class classifier (CM), global regression predictor (R1) and the proposed regression predictor with pre-classification (RC). The precision, recall, f-score and detecting time of the four methods are compared respectively.

Fig. 6 shows the results of the comparison. Influenced by the dynamic threshold, all the precision, recall and f-score of binary classifier, which is trained on the fixed threshold, are not very good. In contrast, in the multi-class classifier and regression predictor, the negative influence of dynamic threshold can be mitigated to some extent and better results are achieved. Compared with multi-class classifier, regression predictor with pre-classification has better performance. This is because the proposed modified regression predictor makes a further regression on the multi-classification results instead of directly selecting the nearest classification results, thus the misclassification caused by the difference between actual threshold and training threshold in the multi-class classifier is improved. However, the addition of regression step also increases the testing time of the proposed method. Comparing the results of global regression predictor and regression predictor with pre-classification, it can be found that the performance of global regression is much poorer. This is because it is hard to construct a global model on the larger range. If the model is not reasonable or the parameters are not well adjusted, the performance of the global regression will be greatly reduced. Different from the global regression, the proposed regression with pre-classification compresses the range of flow size to be predicted in advance, which results in better performance. In addition, early detection of elephant flows in the pre-classification stage also helps to reduce detection time. It is worth noting that, in this paper, the model and parameters of the global regression are almost same with the pre-classification regression. And under the same parameter settings, the pre-classification regression is much better. Global regression prediction is a complicated problem. If we have enough training data and could adjust the model and parameters regardless of the cost, global regression may get a better result.

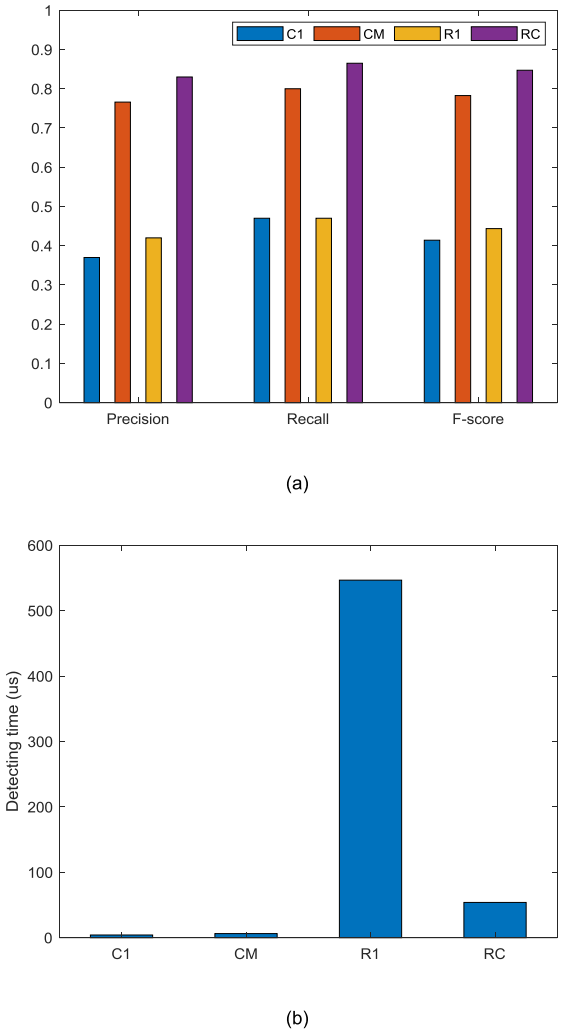
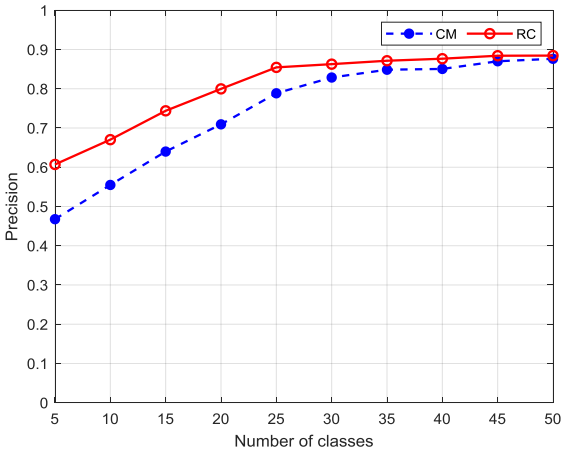


FIGURE 6. The performance comparison of various elephant flow detection methods: (a) Comparison of precision, recall and f-score; (b) Comparison of detecting time.

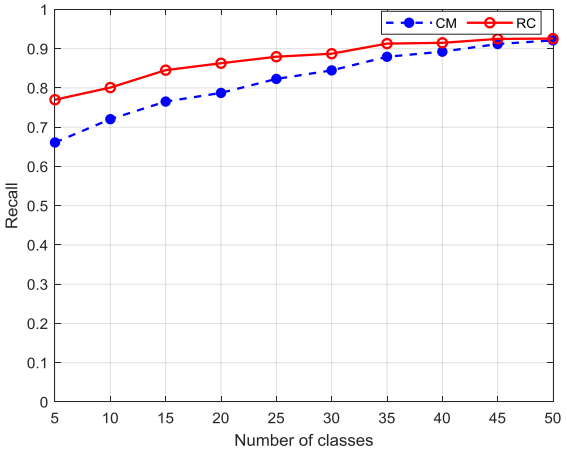
2) EFFECTS OF THE NUMBER OF CLASSES IN PRE-CLASSIFICATION

In order to investigate the effects of the number of classes in pre-classification, we compare the multi-class classifier and the proposed regression predictor under different numbers of classes. The results are shown in Fig. 7.

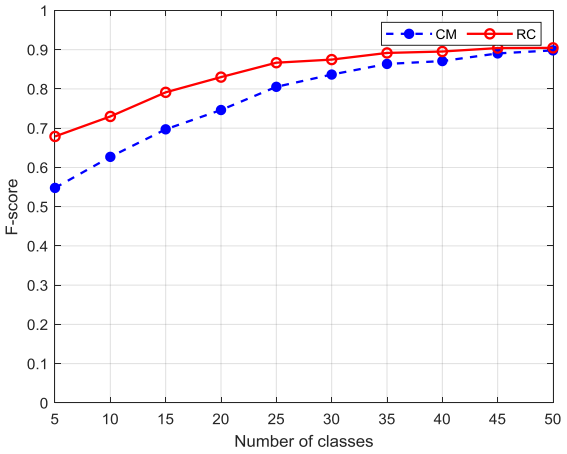
As shown in Fig. 7, with the increase of the number of classes, the performances of multi-class classifier and regression predictor with pre-classification continue to improve, and finally tend to almost the same high level. The simulation results indicate that the increase of the number of classes is conducive to the improvement of the performance of the



(a)

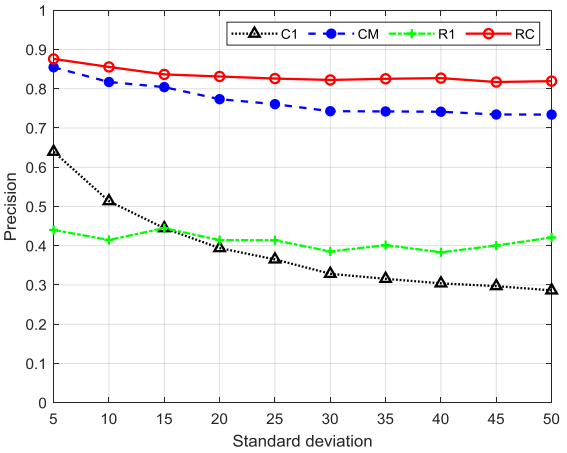


(b)

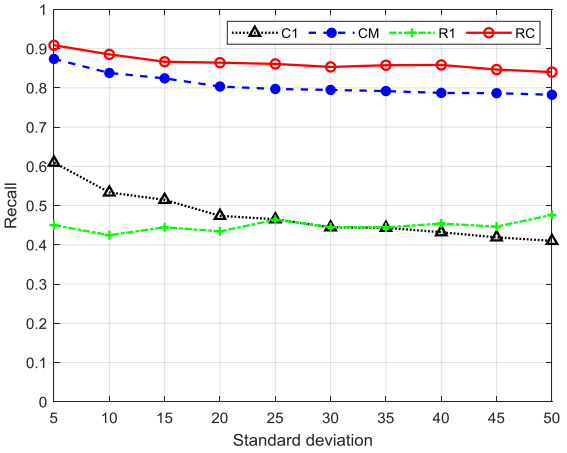


(c)

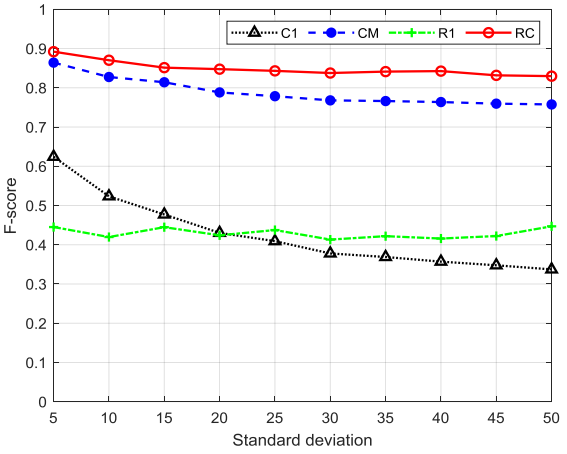
FIGURE 7. The performance comparison of multi-class classifier and regression predictor with pre-classification under different numbers of classes: (a) Comparison of precision; (b) Comparison of recall; (c) Comparison of f-score.



(a)



(b)



(c)

FIGURE 8. The performance comparison of various elephant flow detection methods under different standard deviations of dynamic threshold: (a) Comparison of precision; (b) Comparison of recall;(c) Comparison of f-score.

elephant flow detection under dynamic thresholds. Besides, in terms of the precision, recall and f-score, the regression predictor with pre-classification is always better than the multi-class classifier. With the increase of the number of classes, the gaps between them decrease. This is because that, the classification granularity is continuously refined with the

increase of the number of classes and the role of regression has been weakened. Under the circumstances, the better performance can be achieved only with multi-class classifier.

3) INFLUENCE OF DYNAMIC THRESHOLD

In order to verify the robustness of the proposed method, we evaluate the performance of the method under different changing intensities of the detection threshold. As mentioned earlier, the changing of dynamic threshold is assumed to be normally distributed. Therefore, we can change the intensity of the detection threshold by changing the standard deviation of normal distribution.

Fig. 8 shows the performance comparison of four different elephant flow detection methods under different standard deviations of dynamic threshold. It can be found that, with the increase of standard deviation of the dynamic threshold, the detection performance of binary classifier (C1) degrades greatly. Different from continuous degradation of binary classifier, the performances of both the regression predictor (R1 and RM) and the multi-class classifier (CM) decrease slightly with the increase of standard deviations of the dynamic threshold. Comparing the regression predictor with pre-classification and the multi-class classifier, the former is even lesser. In contrast, limited by the accuracy of flow size prediction, the performance of the global regression method keeps at a low level.

Through the comparisons above, it can be seen that the proposed method of pre-classification regression can be well applied to the problem of dynamic threshold flow detection, and the performance is relatively good.

V. CONCLUSION

In this paper, we propose a regression method to deal with the dynamic elephant flow detection in AN. Flow size regression is regarded as an intermediate to adapt to the dynamic change of detection thresholds. The elephant flows are identified by comparing the regression result with the specific detection threshold. In order to reduce the detection cost and improve the accuracy of flow size regression, waiting-window filtering mechanism and pre-classification strategy are used to filter out most small flows and compress the range of flow size to be predicted. The simulation results verify the proposed method, and the performance is relatively good.

For future work, it is necessary to make further studies of traffic generation and emulation for AN, as the actual data of AN is hard to collected. In addition, further studies of more fine-grained traffic classification, such as, the combination of regular traffic classification and elephant flow detection, and more general frameworks or models of traffic classification are also necessary to be studied to improve network performance.

REFERENCES

[1] D. Medina, F. Hoffmann, F. Rossetto, and C.-H. Rokitansky, "A geographic routing strategy for North Atlantic in-flight Internet access via airborne mesh networking," *IEEE/ACM Trans. Netw.*, vol. 20, no. 4, pp. 1231–1244, Aug. 2012.

[2] N. Gupta and A. Aggarwal, "Airborne Internet the Internet in the air," in *Proc. 7th Int. Conf. Cloud Comput., Data Sci. Eng.*, Jan. 2017, pp. 441–444.

[3] W. Pan and W. Li, "Dynamic network management and service integration for airborne network," in *Proc. Int. Conf. Space Inf. Technol.*, Dec. 2009, Art. no. 76511A.

[4] X. He and X. Zhang, "Network scheme for airborne weapon-cooperation data link," *Command Inf. Syst. Technol.*, vol. 2, no. 3, pp. 19–22, 2011.

[5] A. Gudibanda, J. Ros-Giralt, A. Commike, and R. Lethin, "Fast detection of elephant flows with Dirichlet-categorical inference," in *Proc. IEEE/ACM Innovating Netw. for Data-Intensive Sci. (INDIS)*, Nov. 2018, pp. 10–22.

[6] J. Tao, Y. Li, Z. Wang, P. Xu, and C. Su, "Self-adaptive probabilistic sampling for elephant flows detection," in *Proc. IEEE Global Commun. Conf.*, Dec. 2019, pp. 1–6.

[7] N. Lyu, J. Zhou, X. Feng, K. Chen, and W. Chen, "A timeliness-enhanced traffic identification method in airborne network," *Xibei Gongye Daxue Xuebao/J. Northwestern Polytechnical Univ.*, vol. 38, no. 2, pp. 341–350, Apr. 2020.

[8] T. Yang, H. Zhang, J. Li, J. Gong, S. Uhlig, S. Chen, and X. Li, "HeavyKeeper: An accurate algorithm for finding top-k elephant flows," *IEEE/ACM Trans. Netw.*, vol. 27, no. 5, pp. 1845–1858, Oct. 2019.

[9] F. Tang, H. Zhang, L. T. Yang, and L. Chen, "Elephant flow detection and differentiated scheduling with efficient sampling and classification," *IEEE Trans. Cloud Comput.*, early access, Feb. 26, 2019, doi: 10.1109/TCC.2019.2901669.

[10] R. Jordi, "A mathematical framework for the detection of elephant flows," *ArXiv abs/1701*, vol. 01683, pp. 1–13, May 2017.

[11] R. Ben Basat, G. Einziger, R. Friedman, and Y. Kassner, "Optimal elephant flow detection," in *Proc. IEEE Conf. Comput. Commun.*, May 2017, pp. 1–9.

[12] M. Chen, S. Chen, and Z. Cai, "Counter tree: A scalable counter architecture for per-flow traffic measurement," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 1249–1262, Apr. 2017.

[13] Y. Afek, A. Bremler-Barr, S. Landau Feibish, and L. Schiff, "Sampling and large flow detection in SDN," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 345–346, Sep. 2015.

[14] Y. Sun, W. Liu, Z. Liu, and C. Liu, "Comparison of five Packet-Sampling-Based methods for detecting elephant flows," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2016, pp. 2018–2023.

[15] Z. Zhang, B. Wang, and J. Lan, "Identifying elephant flows in Internet backbone traffic with Bloom filters and LRU," *Comput. Commun.*, vol. 61, pp. 70–78, May 2015.

[16] K. Lou, Y. Yang, and C. Wang, "An elephant flow detection method based on machine learning," in *Proc. SmartCom*, 2019, pp. 212–220.

[17] P. Xiao, W. Qu, H. Qi, Y. Xu, and Z. Li, "An efficient elephant flow detection with cost-sensitive in SDN," in *Proc. 1st Int. Conf. Ind. Netw. Intell. Syst.*, 2015, pp. 24–28.

[18] S.-C. Chao, K. C.-J. Lin, and M.-S. Chen, "Flow classification for software-defined data centers using stream mining," *IEEE Trans. Services Comput.*, vol. 12, no. 1, pp. 105–116, Jan. 2019.

[19] Y.-H. Huang, W.-Y. Shih, and J.-L. Huang, "A classification-based elephant flow detection method using application round on SDN environments," in *Proc. 19th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Sep. 2017, pp. 231–234.

[20] L. Peng, B. Yang, and Y. Chen, "Effective packet number for early stage Internet traffic identification," *Neurocomputing*, vol. 156, pp. 252–267, May 2015.

[21] E. Arestrom and N. Carlsson, "Early online classification of encrypted traffic streams using multi-fractal features," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Apr. 2019, pp. 84–89.

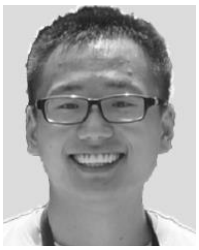
[22] K.-C. Lan and J. Heidemann, "A measurement study of correlations of Internet flow characteristics," *Comput. Netw.*, vol. 50, no. 1, pp. 46–62, Jan. 2006.

[23] A. R. Curtis, "DevoFlow: Scaling Flow Management for High-Performance Networks," in *Proc. ACM SIGCOMM*, 2011, pp. 254–265.

[24] C.-Y. Lin, C. Chen, J.-W. Chang, and Y. H. Chu, "Elephant flow detection in datacenters using OpenFlow-based hierarchical statistics pulling," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 2264–2269.

[25] Y. Afek, A. Bremler-Barr, S. Landau Feibish, and L. Schiff, "Detecting heavy flows in the SDN match and action model," *Comput. Netw.*, vol. 136, pp. 1–12, May 2018.

[26] M. Kiran and A. Chhabra, "Understanding flows in high-speed scientific networks: A netflow data study," *Future Gener. Comput. Syst.*, vol. 94, pp. 72–79, May 2019.
[27] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Acm Sigcomm Conf. Internet Meas.*, pp. 267–280, 2010.
[28] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, Nov. 2016.
[29] N. Williams, S. Zander, and G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 5, pp. 5–16, Oct. 2006.
[30] M. Shafiq, X. Yu, A. A. Laghari, L. Yao, N. K. Karn, and F. Abdessamia, "Network traffic classification techniques and comparative analysis using machine learning algorithms," in *Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, Oct. 2016, pp. 2451–2455.
[31] J. Kampeas, A. Cohen, and O. Gurewitz, "Traffic classification based on zero-length packets," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 3, pp. 1049–1062, Sep. 2018.
[32] C. Rasmussen and C. Williams, "Gaussian Processes for Machine Learning," in *Adaptive Computation and Machine Learning*, Cambridge, MA, USA: MIT Press, 2006.
[33] N. Zhang, J. Xiong, J. Zhong, and K. Leatham, "Gaussian process regression method for classification for high-dimensional data with limited samples," in *Proc. 8th Int. Conf. Inf. Sci. Technol. (ICIST)*, Jun. 2018, pp. 358–363.
[34] *UNIBS: Data Sharing*. [Online]. Available: http://www.ing.unibs.it/ntw/tools/traces/
[35] R. Trafton and S. V. Pizzi, "The joint airborne network services suite," in *Proc. Mil. Commun. Conf.*, Oct. 2006, pp. 1–5.
[36] S. Adams, B. Cain, K. Olds, and P. Griessler, "A comparison of FDD and TDD/TDMA architectures for airborne backbone network traffic," in *Proc. IEEE Mil. Commun. Conf.*, Nov. 2008, pp. 1–7.
[37] Y. Liang, "Research progress on architecture and protocol stack of the airborne network," *Ruan Jian Xue Bao/Joumal Softw.*, vol. 27, no. 1, pp. 96–111, 2016.
[38] X. Li, "Systematic medium access control in hierarchical airborne networks with multi-beam and single-beam antennas," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 2, pp. 706–716, May 2019.



PENGFEI LIU received the B.S. degree from Xi'an Jiaotong University, Xi'an, China, in 2012, and the M.S. degree from the National University of Defense Technology, Changsha, China, in 2014. He is currently pursuing the Ph.D. degree with Air Force Engineering University.

His research interests include airborne networks, traffic classification, machine learning, and aviation data link systems.

NA LV received the B.S. degree in testing technology and instrumentation, the M.S. degree in control theory and applications, and the Ph.D. degree in armament science and technology from Northwestern Polytechnical University, Xi'an, China, in 1992, 1995, and 2010, respectively.

She is currently a Full Professor with Air Force Engineering University, Xi'an. Her current research interests include aviation datalink systems, military air communications, and software-defined networking.

KEFAN CHEN received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2013, and the M.S. and Ph.D. degrees from Air Force Engineering University, Xi'an, China, in 2016 and 2019, respectively.

His research interests include airborne tactical networks, software-defined networking, and aviation data link systems.

LIANG TANG received the Ph.D. degree in information and communication engineering from the National University of Defense Technology, in 2018.

His research interests include big data and radar signal processing.

JIAXIN ZHOU received the B.S. degree from Hunan University, Changsha, China, in 2016, and the M.S. degree from Air Force Engineering University, Xi'an, China, in 2019.

His research interests include airborne tactical networks, traffic classification, machine learning, and aviation data link systems.