FLAGB: Focal Loss based Adaptive Gradient Boosting for Imbalanced Traffic Classification

Yu Guo*†, Zhenzhen Li*†, Zhen Li*†, Gang Xiong*†, Minghao Jiang*†, Gaopeng Gou*†✉
*Institute of Information Engineering, Chinese Academy of Sciences
†School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China
{guoyu,lizhenzhen,lizhen,xionggang,jiangminghao,gougaopeng}@iie.ac.cn

***Abstract*—Machine learning (ML) is widely applied to network traffic classification (NTC), which is an essential component for network management and security. While the imbalance distribution exhibiting in real-world network traffic degrades the classifier's performance and leads to prediction bias towards majority classes, which is always ignored by exiting ML-based NTC studies. Some researches have proposed solutions such as resampling for imbalanced traffic classification. However, most methods don't take traffic characteristics into account and consume much time, resulting in unsatisfactory results. In this paper, we analyze the imbalanced traffic data and propose the focal loss based adaptive gradient boosting framework (FLAGB) for imbalanced traffic classification. FLAGB can automatically adapt to NTC tasks with different imbalance levels and overcome imbalance without the prior knowledge of data distribution. Our comprehensive experiments on two network traffic datasets covering binary and multiple classes prove that FLAGB outperforms the state-of-the-art methods. Its low time consumption during training also makes it an excellent choice for highly imbalanced traffic classification.**

***Index Terms*—machine learning, imbalanced traffic classification, security, focal loss, gradient boosting**

I. Introduction

With the explosive growth of Internet applications, network traffic classification (NTC) has become the fundamental component of network management and cybersecurity. At present, machine learning (ML) is the most mainstream and effective technology applying to NTC [1] [2]. However, the imbalance nature of real-world network traffic poses great challenges to ML-based NTC schemes [3]. ML algorithms are always designed to achieve the highest overall accuracy, which may lead to prediction bias towards majority classes [4]. The performance degradation on minority classes may be catastrophic in some scenarios such as malicious traffic identification and intrusion detection, where the malicious traffic accounts for a very small proportion. For example, in malicious robot detection tasks, a poor precision on the malicious robots will result in misclassifying a normal user as a malicious robot, seriously damaging the users' experience. While in intrusion detection tasks, a low detection rate on abnormal attacks will lead to severe security consequences to the system. Therefore, imbalance must be taken into account in future NTC researches.

Some studies have proposed several solutions to combat imbalance in NTC [5]. The most common solutions are to resample the training set to rebalance it [6]. The advanced solutions combine resampling techniques and ensemble algorithms to further improve the classifier's performance [7]. In addition, there are also proposals to consider the design of misclassification cost or class weight [8]. However, these studies have some problems. First, the resampling based methods may lose potentially useful information or increase the risk of overfitting as well as the time consumption. Secondly, most methods directly use the general techniques which have been designed to alleviate data imbalance without considering the particular characteristics of the network traffic, resulting in unstable effects and poor generalization capabilities on the imbalanced NTC tasks. An alternative idea is to design an end-to-end model that mitigating the traffic's imbalance during each iteration of training, i.e., combining a well-designed loss function and an efficient algorithm into a framework. No resampling is required in this framework, thus avoiding the drawbacks mentioned above.

Focal loss is proposed in the field of object detection for solving the extreme foreground-background class imbalance which degrades the first-stage detector's performance [9]. Through the analysis in Section III, we find that there are similarities between imbalanced traffic classification and object detection. Besides, gradient boosting is an excellent algorithm for its high accuracy [10]. Based on these considerations, this paper proposes a framework named focal loss based adaptive gradient boosting (FLAGB) for imbalanced traffic classification. FLAGB doesn't need to preprocess the training data, retaining the rich information in raw traffic data and avoiding the extra time consumption caused by resampling.

The main contributions of our work are summarized as follows:

- We propose the FLAGB framework to combat imbalance in network traffic classification. Considering the characteristics of imbalanced network traffic, FLAGB can reduce the weight of majority samples in disguise during the training phase and effectively compensate for the classifier's degradation caused by class imbalance.

✉ Corresponding author.
E-mail address: gougaopeng@iie.ac.cn

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

- Without the prior knowledge of data distribution, FLAGB can adapt to the imbalanced traffic dataset under different network scenarios. The classifier can achieve the best effect on the target metric with the optimal parameters automatically found by FLAGB.
- Our FLAGB achieves excellent results on a real-world network traffic dataset and the well-known KDD 99 dataset, and outperforms several state-of-the-art methods under various imbalance levels. Furthermore, FLAGB guarantees less time consumption in highly imbalanced NTC tasks.

The rest of the paper is organized as follows. Section II summarizes the related work. Our proposed framework is introduced in Section III. Section IV presents the experiments in detail. Finally, we conclude this paper in section V.

II. Related Work

Class imbalance has been widely studied as one of the most challenging problems in machine learning. The solutions can be divided into three categories: data-level methods, algorithm-level approaches, and cost-sensitivity methods [4]. Data-level methods, including oversampling, undersampling and hybrid algorithms, resample the dataset to diminish imbalance. Oversampling copies or synthesizes samples belonging to minority classes to rebalance the class distribution, while undersampling reduces samples of majority classes to achieve the same goal. Hybrid algorithms such as SMOTE-TL, SMOTE-ENN, combine two sampling techniques [11] [12]. Algorithm-level method is actually a hybrid model combining data-level approaches and ensemble algorithms, which uses resampling to mitigate data imbalance and boosting-like algorithms to enhance the classifier's performance. Cost-sensitive methods consider diverse costs for different classes, which directly modify the learning procedure to improve the classifier's sensitivity towards minority classes. It may bring better effect with a well-designed cost [4] [5].

Some works have sought solutions for combating imbalance in NTC, among which data-level methods are widely adopted. Seo et al. proposed an approach to find the optimal SMOTE ratio in imbalanced datasets for intrusion detection [6]. Oeung et al. put forward a clustering-based undersampling method called CTU in their NTC framework [13]. The key idea of CTU is to select the most informative samples from majority classes, which are determined by clustering. Data-level methods are easy to implement and effective to some degree. However, their performance cannot be guaranteed in real-world network traffic, because oversampling usually consumes much time and undersampling loses important information when reducing majority samples. Wei et al. compared several boosting-based ensemble algorithms for real-time imbalanced NTC and proposed a similar method called BalancedBoost [14]. They claimed that it outperformed other methods on the UNIBS dataset. However, Khoshgoftaar et al. found that ensemble learning was more time-consuming than the data-level approach and the cost-sensitive method [15]. Peng et al. proposed a cost-sensitive method called IDGC [8] and applied it to imbalanced NTC, achieving good results [16]. But they also pointed out that IDGC's computational complex was relatively high [17]. Another cost-sensitive called MetaCost [18] was used by Liu et al. and the authors demonstrated its effectiveness under different network scenarios [19]. Furthermore, Gomez et al. made a comprehensive review of imbalanced NTC and concluded several well-performing methods [5]. Unfortunately, these methods haven't considered the characteristics of network traffic. In this paper, we design a cost-sensitive and boosting combining method, which is fit for the real-world network traffic for its good performance.

III. Methodology

A. Problem Definition

In a given NTC task, for its dataset D composed of n ($n \geq 2$) categories, the sample size of the *i*th category is N_i. If there is a large difference in the sample size of n categories, i.e., $N_i \gg N_j$, then D is an imbalanced dataset. Classes with a larger sample size are called majority classes, and other smaller classes are minority classes. The NTC task on an imbalanced dataset is called imbalanced traffic classification. How to improve the classifier's performance on minority classes while maintaining the accuracy of majority classes in NTC tasks is our goal. In this research, we encode the majority class as 0, which is the negative class in binary classification, and the minority classes as 1,2,...,n-1, which corresponds to the positive class.

B. Focal Loss

In object detection tasks, Lin et al. believe the extreme foreground-background class imbalance hinders the first-stage detector from achieving a better performance. Therefore, they improve the traditional cross entropy (CE) loss function and devise Focal Loss (FL), which focuses training on hard examples, avoiding the vast number of easy negative examples from overwhelming the detector during training [9]. Hard example here refers to the examples in the training set that are poorly-predicted, i.e. being mislabeled by the current version of the classifier. Easy example is exactly the opposite.

The formula of focal loss for binary classification is as follows:

$$\mathrm{FL}\left(p_{\mathrm{t}}\right)=-\alpha_{\mathrm{t}}\left(1-p_{\mathrm{t}}\right)^{\gamma}\log\left(p_{\mathrm{t}}\right) \tag{1}$$

where p_{t} and α_{t} are defined as:

$$p_{\mathrm{t}},\alpha_{\mathrm{t}}=\begin{cases} p,\alpha & \text{if } y=1 \\ 1-p,1-\alpha & \text{otherwise} \end{cases} \tag{2}$$

Fig. 1 shows the FL function with different γ. When γ=0, it is CE loss. As can be seen, CE loss of the easy example (i.e, $p_{\mathrm{t}} \gg 0.5$) is still relatively large. While with the increasing of γ, FL values of easy examples are greatly reduced, but that of hard examples (i.e, $p_{\mathrm{t}} \ll 0.5$) are reduced to a maximum of one quarter (i.e. when $p_{\mathrm{t}} = 0.5$), making the classifier concentrate on hard examples.

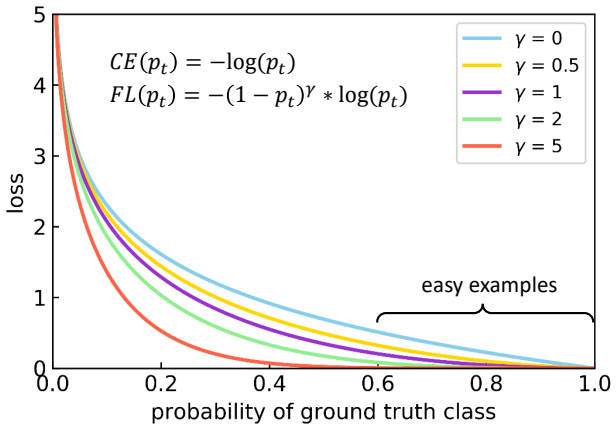

Fig. 1. Focal loss versus predicted probability under different γ values

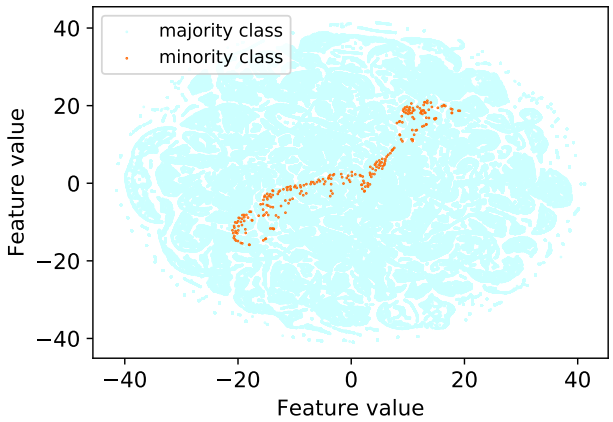

Fig. 2. Visualization of imbalanced traffic data by TSNE

C. Gradient boosting algorithm

Gradient boosting is also called GBDT because it is often an ensemble model of decision trees [10]. In each iteration, GBDT uses the negative gradient to fit the approximation of the current loss and learn the new tree. This method achieves state-of-the-art performances in many machine learning tasks. However, due to its high computational complexity, it is difficult to balance accuracy and efficiency when the data size is large. Ke et al. proposed a novel GBDT algorithm called LightGBM, which made a big breakthrough in terms of computational speed and memory consumption [20]. It solves the efficiency problem and maintains superior performance. So we choose LightGBM as the concrete implementation of the gradient boosting algorithm in our framework.

D. Focal Loss based Adaptive Gradient Boosting Method

Imbalanced traffic classification tasks have certain similarities with the object detection scenarios described in Section III-B. Fig. 2 visualizes a real-world network traffic dataset used in this paper. We use TSNE to reduce the samples' dimensions and display them in a 2D image. From Fig. 2, most majority samples(light blue) are quite far and clearly distinguishable from the minority samples(orange), which can thus be considered as easy examples. The samples which really deserve attention are those located around the decision boundary and indistinguishable from other classes. Based on this observation, we infer that FL may be effective to help solve the imbalance problem in NTC tasks.

FLAGB is a hybrid method of cost-sensitive and ensemble algorithms. Fig. 3 illustrates the framework of FLAGB. It is mainly divided into three parts according to the different functions and processing phases, including the data preparation part in the middle of Fig. 3, the adaptive tuning part on the left, and the classifier generation part on the right, of which the latter two parts contain core technologies for mitigating the imbalance in traffic data.

First, after the feature extraction and sample labeling, the raw imbalanced network traffic is partitioned into a training set and a validation set according to a certain ratio without the need for any resampling operations. In the adaptive tuning phase, all traffic data are put into use to find the corresponding optimal parameters in this scenario. Then, the training set is used to train the classifier, while the validation set is utilized for observing and assisting the training. The size of the validation set should not be too small to ensure the generalization ability of the classifier.

In the framework, LightGBM is chosen to be the gradient boosting algorithm as mentioned in III-C. Especially, the loss function is replaced with FL, so that the model puts more attention on few hard examples, which equates to reducing the weight of the majority classes and mitigating the degree of traffic imbalance in the training set.

More concretely, the default loss function of LightGBM for binary classification tasks is CE:

$$\mathrm{CE}(p, y) = -(y \log p + (1-y) \log(1-p)) \tag{3}$$

where $y \in \{0, 1\}$ is the ground truth label and $p \in [0, 1]$ is the model's predicted probability for the class with label $y = 1$. Adding a modulating factor related to p and γ and a balanced factor α to CE, FL is obtained:

$$\begin{aligned} \mathrm{FL}(p, y) = &-(\alpha y + (1-\alpha)(1-y)) \\ &\cdot ((1 - (yp + (1-y)(1-p)))^{\gamma}) \\ &\cdot (y \log p + (1-y) \log(1-p)) \end{aligned} \tag{4}$$

For binary NTC tasks, assume that the prediction output of LightGBM is *pred*. Then the corresponding probability p is *sigmoid(pred)*, written as *s(pred)* for short. Substituting p into (4), we get:

$$\begin{aligned} \mathrm{FL}(pred, y) = &-(\alpha y + (1-\alpha)(1-y)) \\ &\cdot (1 - (y \cdot s(pred) + (1-y) \cdot (1 - s(pred))))^{\gamma} \\ &\cdot (y \log(s(pred)) + (1-y) \log(1 - s(pred))) \end{aligned} \tag{5}$$

Calculate the first-order and second-order partial derivative of $\mathrm{FL}(pred, y)$ with respect to *pred* and take them as the return value of objective loss function.

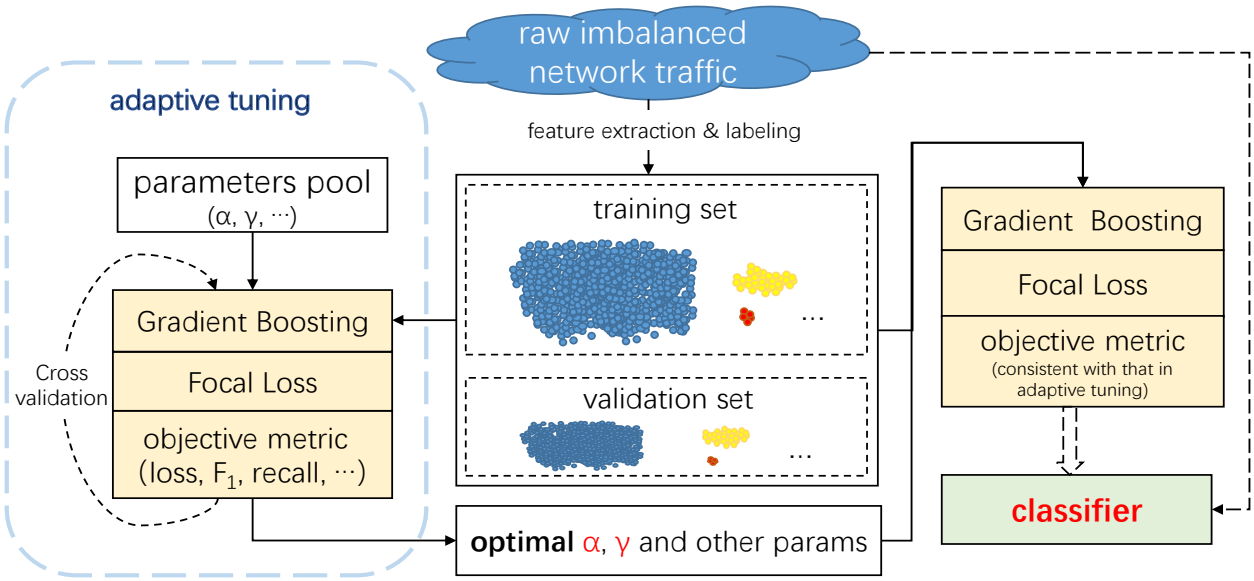

Fig. 3. The framework overview of FLAGB. Its core technology is the combination of cost-sensitive and gradient boosting algorithms. Adaptive tuning function can help find the optimal parameters and adapt the model to fit different scenarios automatically.

For multiclass classification tasks, (5) is also applicable. We extend binary to multiple classification using the idea of one-versus-all. Assume that the training set D has a total of m samples and n classes, and each class is represented by C_i. Then $D = \{C_1, C_2, ..., C_n\}$. The prediction output of LightGBM for multiclass task is a $(1, m * n)$ array. Reshape it to a $m * n$ matrix, i.e. *pred*. In addition, encode the ground-truth label to get the one-hot label matrix of $m * n$ size, which corresponds to y in (5).

For Internet network traffic, the imbalance level varies depending on the specific scenario. The adaptive tuning function is deployed in FLAGB to adapt the model to fit different scenarios automatically. For a given dataset, there is no need to know about the class distribution. The user only needs to set the ranges of all parameters in the parameters pool and the objective metric value required by the task. Referring to [9], the ranges of α and γ are set to be (0,1) and (0.5,5), respectively. For a given task, select the target metric, such as precision, recall, etc., so that the classifier attempts to reach the highest metric result for each class. We use hyperopt and cross-validation library in Python to search for the best α, γ and other parameters.

After getting the optimal parameters, a classifier can be trained on the training set. The current target metric should be consistent with that in adaptive-tuning phase. When the metric results are no longer promoted on the verification set, the training is stopped and the final classifier is obtained.

IV. Experiments

A. Dataset

In order to prove the effectiveness and versatility of our proposed method, we select two datasets from different Internet network environments, covering binary classification and multi-classification tasks. One is a malicious cloudbot dataset called BOT and the other is the KDD99' dataset.

BOT dataset was collected by Guo et al. to study the identification of malicious cloudbots [21]. It was extracted from the raw traffic from online trading servers of a large Internet company. BOT consists of the human-user class and the malicious-cloudbot class, including 141-dimensional statistical features. More information can be found in [21].

KDD Cup 1999 Data, briefly called KDD 99, is widely used for intrusion detection and machine learning research [22]. It consists of 5 main categories, including DOS(Denial of Service), Probe, R2L(Root 2 Local), U2R(User 2 Root) and Normal, among which the first four categories are abnormal. Each sample is described by 41 features. DOS occupies 79.27% of the entire dataset, while Normal only accounts for 19.85%. This is extremely unreasonable, so we remove DOS class. We also remove some fine-grained classes in each category that appear in the testing set but do not appear in the training set. Then the second dataset used in this study is formed, called KDD99'. Since the problem we are exploring is the impact of class imbalance on network traffic classification, removing some of the data mentioned above does not affect the validity of the experimental conclusions.

B. Experiment settings

a) Imbalance level: We use imbalance ratio per label ($IRLbl$) defined by (6) to measure the imbalance level of each dataset [5]. $IRLbl$ is the ratio between the sample size of majority class and that of class i. It is 1 for majority class and the fewer samples a minority class has, the larger its $IRLbl$ will be.

$$IRLbl_i = \frac{N_{majority\ class}}{N_i} \tag{6}$$

TABLE I
CLASS DISTRIBUTION IN KDD99' DATASET. NORMAL IS THE MAJORITY CLASS

KDD99'	Normal(0)		Probe(1)			U2R(2)			R2L(3)		
	s	**%s**	**s**	**%s**	$IRLbl_1$	**s**	**%s**	$IRLbl_2$	**s**	**%s**	$IRLbl_3$
Training set	97278	95.80	4107	4.04	23.7	52	0.05	1870.7	104	0.10	935.4
Testing set	60593	87.81	2377	3.44	25.5	39	0.06	1553.7	5993	8.69	10.1

For BOT dataset, we set up 5 experimental groups by randomly selecting a certain number of samples from two original classes. We set $IRLb_1$ to 50, 100, 200, 500, 1000 respectively to cover different imbalance level, namely BOT-$IRLbl_1$. Sample size of majority class (i.e. negative class) in each training set is set to 250000. We assume that the sample size of testing set is one-tenth of that of training set, and its $IRLb_1$ is consistent with the training set.

No additional operation is done for KDD99'. Each class's sample size(s) and its percentage(%s) are presented in Table I.

b) Performance metrics: Appropriate metrics are very important to objectively assess the effectiveness of various methods and compare them for combating imbalance in NTC. Since it is hard for traditional overall accuracy to reflect the classifier's performance on minority classes, we use overall metrics and individual metrics. Overall metrics can measure the classifier on the whole dataset. While individual metrics assess it on each class, providing us a clear observation about if a given method strengthens minority classes.

Individual metrics: Precision (P) and Recall (R) are adopted as metrics for individual classes. Recall is also known as accuracy or detection rate. They are defined by (7) and (8) separately. For class i, TP_i is the number of samples correctly predicted as class i, FP_i is the number of samples misclassified as class i, TN_i is the number of samples correctly predicted as non-class i, and FN_i is the number of samples that are misclassified as non-class i. Besides, F_1 score defined by (9) is also adopted in the binary classification task, which is the harmonic mean of P_1 and R_1.

$$P_i = \frac{TP_i}{TP_i + FP_i} \tag{7}$$

$$R_i = \frac{TP_i}{TP_i + FN_i} \tag{8}$$

$$F_1 = 2 \cdot \frac{P_1 \cdot R_1}{P_1 + R_1} \tag{9}$$

Overall metrics: We choose overall accuracy (OA) and G-mean (GM) as the overall metrics. OA is the ratio between the number of correctly predicted samples and the total size of the dataset, which is sensitive to class imbalance. While GM is not sensitive to class imbalance. It is the geometric mean of the per-class recall and treats all classes equally. They are separately defined in (10) and (11).

$$OA = \frac{\sum TP_i}{\#\text{ Samples}} \tag{10}$$

$$GM = \sqrt[n]{\prod R_i} \tag{11}$$

c) Estimator for baseline: Decision Tree (DT) has been widely used in NTC for its good performance and interpretability. Moreover, Gomez et al. indicate that CART DT is quite sensitive to class imbalance, which makes it a good base estimator to assess the effects of different methods [5]. So we choose CART DT as the base learner. The results DT generated on different datasets are baselines and all methods in our study will be implemented on this basis.

d) Comparison methods: [5] compared 28 methods for combating imbalance in NTC. We choose the best 9 methods in their study, which also represents the state-of-the-art methods for imbalanced NTC, including 5 data-level methods, 2 algorithm-level techniques, and 1 cost-sensitive approach. Among 5 data-level methods, there are 2 oversampling methods, namely ROS and ADASYNC, 2 undersampling methods, NCR and TL, and 1 hybrid method, SMOTETL. The algorithm-level techniques include SMOTEboost and TLboost. The cost-sensitive approach is MetaCost. Data-level methods are implemented by the Python library, imbalanced-learn. The rest techniques are collected from the project published by [5] on GitHub. Additionally, some researches prove that ensemble algorithms can achieve better results on imbalanced NTC [3] [13]. So we select two typical ensemble algorithms, namely random forest (RF) and LightGBM. Their implementations are from scikit-learn, a Python library.

C. Experiments results and analysis

a) Baseline: First, we use the base estimator, i.e., CART DT from Python scikit-learn library, to build the baseline. We perform 10-fold cross-validation on each experimental group of the BOT dataset. For KDD99 ', we perform ten repeated experiments on the given training set and testing set to avoid interference from some random factors. The baseline results are averaged and shown in Table II and Table III.

From Table II, in binary NTC tasks, as the minority class's IRLbl (i.e. $IRLbl_1$) increases, OA and GM show the opposite trend. OA gradually grows, while GM decreases continuously.

TABLE II
BASELINE RESULTS ON MULTIPLE BOT GROUPS. THE RESULTS ARE EXPRESSED IN %

	Overall metric		Individual metric			
$IRLbl_1$	**OA**	**GM**	P_0	R_0	P_1	R_1
BOT-50	98.49	80.31	99.30	99.15	60.54	65.06
BOT-100	99.04	75.26	99.51	99.51	57.12	56.91
BOT-200	99.47	70.47	99.75	99.72	47.37	49.80
BOT-500	99.69	62.77	99.86	99.83	34.09	39.47
BOT-1000	99.85	56.61	99.93	99.92	30.91	32.08

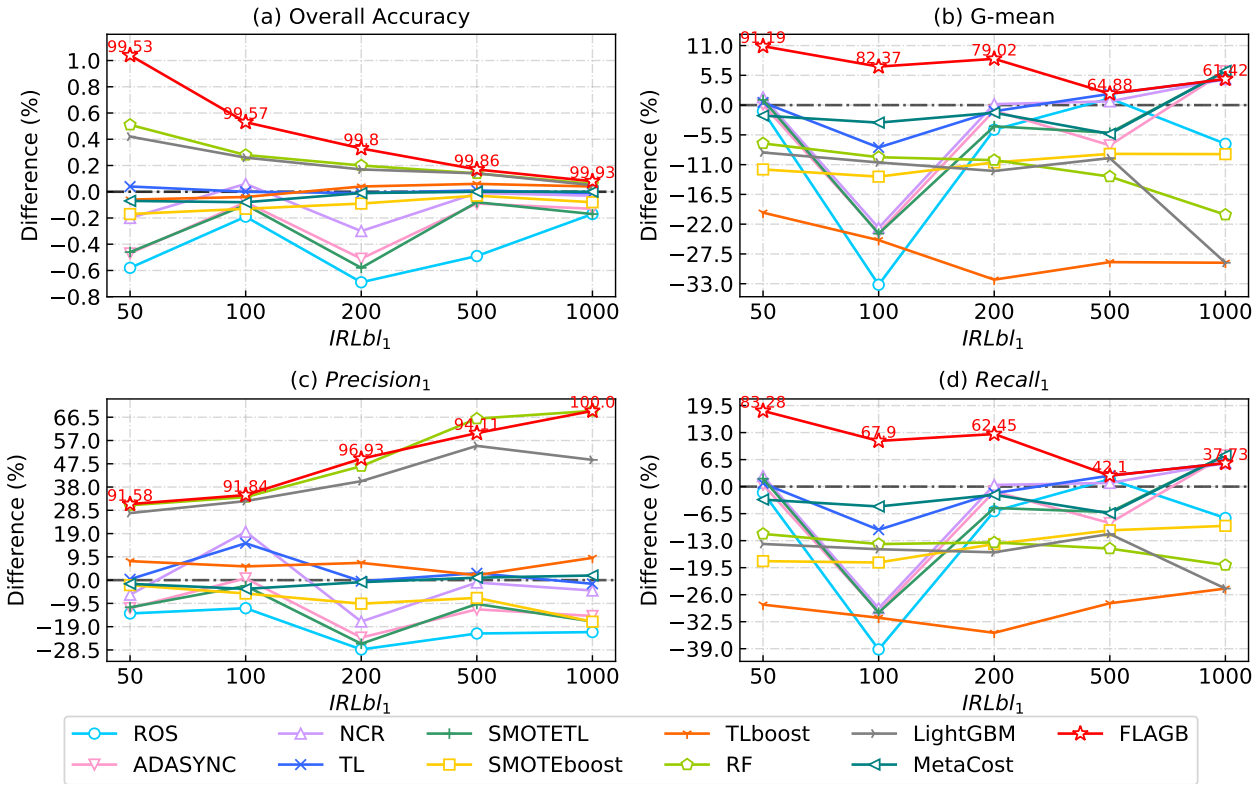

Fig. 4. Overall and individual metrics obtained on BOT groups with different imbalance levels. The results are expressed as percentage differences with the baseline. The red numbers additionally indicate the actual result values of FLAGB.

This is because OA is dominated by the majority class. When $IRLbl_1$ reaches a large value, OA will be approximately equal to R_0. However, GM is not affected by class imbalance and does not bias towards the majority class.In terms of individual metrics, P_0 and R_0 rise as $IRLbl$ grows, while P_1 and R_1 decline, so does F1 score. On KDD99', as shown in Table III, because of the larger IRLbl of class 2 and class 3, their P and R are very poor. Especially for class 2, its sample size is only 0.05% in the training set, far less than that of class 0, causing its P and R to be only 1.51% and 12.82%, respectively.

The baseline results demonstrate that class imbalances can degrade the classifier's performance and the greater the imbalance, the worse the classifier performs on minority classes.

b) Results on BOT dataset: We evaluate the comparison methods mentioned in Section IV-B on all experimental groups. The best parameters for each method are used and the results are averaged over ten experiments. Due to the limited space, the parameters are not presented. To intuitively show how each method improves the baseline on imbalanced datasets, we take the difference between the results of each method and the baseline. Positive values indicate an improvement to the baseline, 0 represents the baseline, and negative values indicate making the effect worse.

The results on series of BOT groups are shown in Fig. 4. Fig. 4(a) illustrates the difference on OA. FLAGB shows the highest improvement. However, the gaps between all methods and baseline are not large, i.e., within $\pm 1\%$, and it is obvious that the improvement of FLAGB decays with the increase of $IRLbl_1$. The reason is that OA is mainly controlled by the majority class, and the enhancement on minority class has little influence on it. What's more, the baseline OAs are all close to 1, which leads to its promotion space approaching 0. Fig. 4(b) shows the difference on GM. (c) and (d) respectively depict the difference on P and R of the minority class. Since the P_0 and R_0 do not change much, they are no longer displayed. It is worth noting that (b) and (d) have almost the same trend. Since GM is the geometric mean of R_0 and R_1 and R_0 is approximately equal to 1, GM basically represents the recall of minority classes. It can be seen that FLAGB has the greatest improvement on R_0 when $IRLbl_1 \leq 500$. When $IRLbl_1 = 1000$, ADASYNC, SMOTETL and MetaCost enhance the baseline by 7.54%, which is higher than FLAGB's 5.65%, and NCR and TL preform similarly to FLAGB. However, according to Fig. 4(c), except for FLAGB, the performance of other methods just mentioned are not optimistic. Only MetaCost has a 1.9% promotion, while others have lowered the baseline to varying degrees. Interestingly, RF and LightGBM have greatly improved P_1 at the expense of severely compromising R_1. Therefore, from a comprehensive perspective, FLAGB is the best method to solve the imbalanced binary NTC under different imbalance levels.

More intuitively, Fig. 5 shows the difference on F_1 score, which fully measures P_1 and R_1 at the same time. FLAGB

TABLE III
OVERALL AND INDIVIDUAL METRICS OBTAINED BY DIFFERENT METHODS ON KDD99'. THE BASELINE RESULTS ARE EXPRESSED IN %, WHILE THE OTHER RESULTS ARE EXPRESSED AS PERCENTAGE DIFFERENCES WITH THE BASELINE

	Overall metric		**Individual metric**							
	OA	**GM**	P_0	R_0	P_1	R_1	P_2	R_2	P_3	R_3
baseline	91.50	31.77	92.62	99.51	76.98	99.03	1.51	12.82	94.34	8.06
ROS	-0.53	-11.63	-0.95	-0.15	1.25	0.59	3.25	-7.69	-20.01	-4.82
ADASYNC	-0.15	-1.00	-0.50	-0.01	-1.16	-2.23	5.74	0.00	2.11	-0.80
NCR	-0.07	7.24	-0.01	-0.01	-0.52	0.59	2.03	**20.51**	2.21	-1.05
TL	-0.05	-0.55	-0.67	-0.02	**12.95**	0.55	0.01	0.00	-0.81	-0.58
SMOTETL	-0.61	-2.71	-1.11	-0.05	0.00	-12.24	7.42	0.00	0.27	-1.62
SMOTEboost	-3.69	-3.85	-1.31	-2.96	-21.78	-18.47	0.86	15.39	-77.72	-5.29
TLboost	-2.22	-6.12	-0.61	-2.35	-9.33	-7.99	-0.94	-7.69	-55.48	1.48
RF	-0.50	-11.25	-0.95	**0.11**	-0.16	**0.63**	51.82	7.69	-1.48	-7.19
LightGBM	0.29	2.19	**0.03**	-0.19	-1.45	0.42	6.82	-2.56	-4.20	**5.06**
MetaCost	-0.39	-3.43	-0.01	0.03	-1.08	-0.04	-0.10	7.69	-2.51	-4.87
FLAGB	**0.37**	**9.89**	-0.09	0.01	-0.61	0.51	**57.31**	12.82	**4.68**	3.80

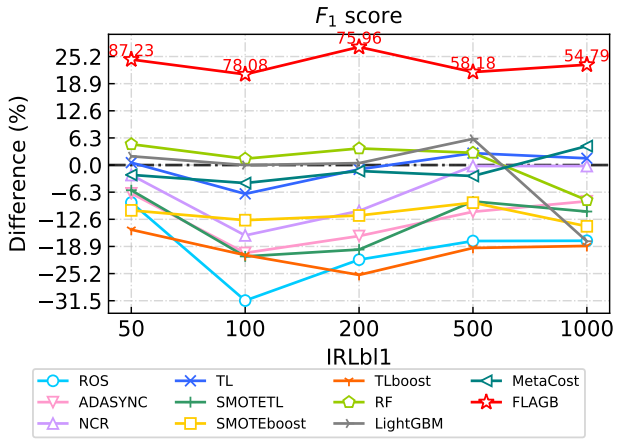

Fig. 5. F_1 score obtained on BOT groups with different imbalance levels

leads other methods with absolute advantage and the effect is relatively stable. Among resampling methods, undersampling is slightly better than oversampling. Two ensemble algorithms, RF and LightGBM, have a slight improvement on baseline when $IRLbl_1 \leq 500$, but their performances drop sharply when $IRLbl_1 = 1000$, indicating that highly imbalanced data is a great challenge to classifiers. However, FLAGB's enhancement on F_1 score is still as high as 23.3%, making it reach 54.8%.

c) Results on KDD99' dataset: Table III shows the difference between the results and the baseline on KDD99'. From the perspective of overall metrics, FLAGB and LightGBM are the only algorithms for improving the baseline on both OA and GM. NCR slightly reduces OA but its GM is greatly improved. In terms of individual metrics, we mainly focus on the precision and recall of class 2 and class 3. Although class 1 is also one of the minority classes, its $IRLbl$ is less than 25 and the baseline P_1 and R_1 are 77% and 99%, respectively, which is a relatively good performance. For class 2, P_2 is in urgent need of a big upgrade as its baseline is only 1.51%. RF and FLAGB bring more than 50% improvement to P_2, and R_2 also has a good promotion, especially for FLAGB. As for other methods, NCR has the largest enhancement on R_2, while its increase on P_2 is small.

For class 3, since the training set size is far smaller than the testing set, resulting in an inadequate learning on class 3, R_3 is difficult to be significantly upgraded. However, RF has a 7.19% drop on R_3, which is catastrophic, because the baseline R_3 is only 8.06%, which means RF's R_3 is 0.87%, almost unable to recognize class 3 in testing set. LightGBM's improvement on R_3 is the maximum among all methods, but its promotion to class 2 is too slight. In contrast, FLAGB takes all classes into account and makes the most reasonable improvements. To sum up, FLAGB is the most efficient method, followed by NCR.

d) Time consumption analysis: We also evaluate the time consumption of several well-performing methods. According to the above analysis, in addition to FLAGB, MetaCost, TL, and NCR also perform relatively well in highly imbalanced situations and multi-classification tasks. Among them, TL and NCR are undersampling methods. Their time consumptions during training are mainly spent in the resampling operations, i.e., the removal of samples. MetaCost, as a cost-sensitive method, mainly consumes time in re-assigning labels to samples according to a certain strategy. For these three methods, we use the default parameters in the standard library or existing implementations. For FLAGB, the early-stopping strategy is adopted, that is, stop training when the score on the validation set no longer continues to grow for 20 iterations. This will ensure the best classification performance of FLAGB. We conduct multiple rounds of training and average the time consumption for each method on BOT-1000 and KDD99'. The results are presented in Fig. 6.

On BOT-1000, FLAGB only spends 37s to obtain the well-trained classifier, while other methods take at least 130s to train. On KDD99', FLAGB consumes almost the same amount of time as MetaCost but its performance is much better than that. Therefore, FLAGB guarantees excellent time consumption while performing much better than other methods, which makes it especially suitable for big and imbalanced traffic data

in real-world network environments.

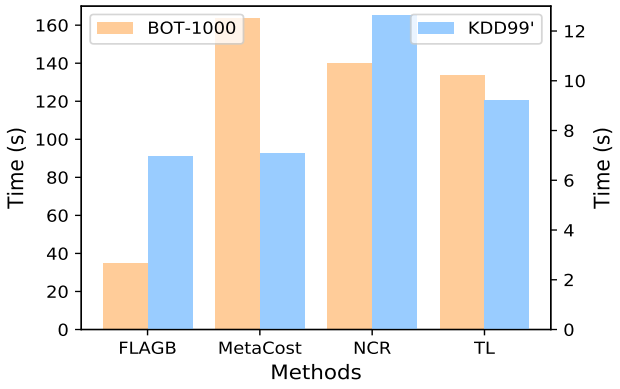

Fig. 6. Time consumption during training of the top 4 well-performing methods for imbalanced traffic classification. Time on BOT-1000 corresponds to the left ordinate while time on KDD 99' corresponds to the right ordinate.

V. Conclusion

The imbalance nature of Internet traffic poses a great challenge to the machine learning based network traffic classification, while researches in this area are not sufficient. Based on the observation of traffic characteristics, we propose a cost-sensitive and ensemble algorithm combination method, named FLAGB. It imports focal loss in the process of learning, which reduces the weight of majority samples in disguise, alleviating the imbalance of training data. Besides, it can give the best adaptation to different imbalanced datasets without manual operation. The comprehensive experiments demonstrate the effectiveness and low time-consumption of FLAGB comparing with other state-of-the-art methods.

Acknowledgment

This work is supported by The National Natural Science Foundation of China (No. U1636217) and The National Key Research and Development Program of China (No.2016QY05X1000 and No. 2018YFB1800200) and Key research and Development Program for Guangdong Province under grant No. 2019B010137003. Additionally, we sincerely appreciate the discussion and kind help provided by Javier Rodriguez Zaurin in code implementation.

References

[1] C. Liu, L. He, G. Xiong, Z. Cao, and Z. Li, "Fs-net: A flow sequence network for encrypted traffic classification," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1171–1179.

[2] Z. Zhang, C. Kang, G. Xiong, and Z. Li, "Deep forest with lrrs feature for fine-grained website fingerprinting with encrypted ssl/tls," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 851–860.

[3] S. S. M. Amina, B. Abdolkhalegh, N. K. Khoa, and C. Mohamed, "Featuring real-time imbalanced network traffic classification," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2018, pp. 840–846.

[4] Q. Dong, S. Gong, and X. Zhu, "Imbalanced deep learning by minority class incremental rectification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 6, pp. 1367–1381, 2018.

[5] S. E. Gómez, L. Hernández-Callejo, B. C. Martínez, and A. J. Sánchez-Esguevillas, "Exploratory study on class imbalance and solutions for network traffic classification," *Neurocomputing*, vol. 343, pp. 100–119, 2019.

[6] J.-H. Seo and Y.-H. Kim, "Machine-learning approach to optimize smote ratio in class imbalance dataset for intrusion detection," *Computational Intelligence and Neuroscience*, vol. 2018, 2018.

[7] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "Smoteboost: Improving prediction of the minority class in boosting," in *Knowledge Discovery in Databases: PKDD 2003, 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, September 22-26, 2003, Proceedings*, 2003.

[8] L. Peng, H. Zhang, B. Yang, and Y. Chen, "A new approach for imbalanced data classification based on data gravitation," *Information Sciences*, vol. 288, pp. 347–373, 2014.

[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[10] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[11] G. E. Batista, A. L. Bazzan, M. C. Monard *et al.*, "Balancing training data for automated annotation of keywords: a case study." in *WOB*, 2003, pp. 10–18.

[12] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[13] P. Oeung and F. Shen, "Imbalanced internet traffic classification using ensemble framework," in *2019 International Conference on Information Networking (ICOIN)*. IEEE, 2019, pp. 37–42.

[14] H. Wei, B. Sun, and M. Jing, "Balancedboost: A hybrid approach for real-time network traffic classification," in *2014 23rd International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2014, pp. 1–6.

[15] T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Comparing boosting and bagging techniques with noisy and imbalanced data," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 41, no. 3, pp. 552–568, 2010.

[16] Z. Chen, Q. Yan, H. Han, S. Wang, L. Peng, L. Wang, and B. Yang, "Machine learning based mobile malware detection using highly imbalanced network traffic," *Information Sciences*, vol. 433, pp. 346–364, 2018.

[17] L. Peng, H. Zhang, Y. Chen, and B. Yang, "Imbalanced traffic identification using an imbalanced data gravitation-based classification model," *Computer Communications*, vol. 102, pp. 177–189, 2017.

[18] P. Domingos, "Metacost: A general method for making classifiers cost-sensitive," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999, pp. 155–164.

[19] Q. Liu and Z. Liu, "A comparison of improving multi-class imbalance for internet traffic classification," *Information Systems Frontiers*, vol. 16, no. 3, pp. 509–521, 2014.

[20] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in neural information processing systems*, 2017, pp. 3146–3154.

[21] Y. Guo, J. Shi, Z. Cao, C. Kang, G. Xiong, and Z. Li, "Machine learning based cloudbot detection using multi-layer traffic statistics," in *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 2019, pp. 2428–2435.

[22] S. Stolfo *et al.*, "Kdd cup 1999 dataset," *http://kdd. ics. uci. edu*, 1999.