IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 3, 2021, accepted June 22, 2021, date of publication June 28, 2021, date of current version July 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3093094

An Effective Cost-Sensitive XGBoost Method for Malicious URLs Detection in Imbalanced Dataset

SHEN HE, BANGLING LI, HUAXI PENG, JUN XIN, AND ERPENG ZHANG
Department of Security Technology, China Mobile Research Institute, Beijing 100053, China

Corresponding author: Bangling Li (libangling@chinamobile.com)

This work did not involve human subjects or animals in its research.

ABSTRACT Imbalanced class has been a common problem encountered in the modeling process, and has attracted more and more attention from scholars. Biased classifiers, which limit the classifiers' performance for minority classes, will be produced if the imbalanced ratio between the number of positive labels and negative labels is ignored. The synthetic minority over-sampling technique (SMOTE) is a very classic and popular over-sampling method, which is widely used to address this problem. However, SMOTE increases label noise and the training time during the over-sampling process. To improve the detection rate of minority classes while ensuring efficiency, we propose a cost-sensitive XGBoost (CS-XGB) for the imbalanced data problem. The CS-XGB method can reduce the classifiers' preference for most classes without changing the distribution of the original data. 600000 Uniform Resource Locators (URLs) were collected to validate the CS-XGB method. We compare XGBoost (XGB), SMOTE+XGB and CS-XGB, and the experimental results confirm that the CS-XGB is robust and efficient for imbalanced cases.

INDEX TERMS Cost-sensitive learning, malicious URLs detection, SMOTE, XGBoost.

I. INTRODUCTION

With the development of the Internet, cyber attacks have become an increasingly important security issue. Many types of attacks, such as phishing, Trojan horses, and malware, often use malicious URLs as a means. Now that the URL generation algorithm is mature, a large number of malicious URLs appear every day. Therefore, identifying malicious URLs is of great significance to prevent various network attacks and maintain network security.

The most traditional method of detecting malicious URLs is the blacklist method [1]. Although the method is simple and straightforward, with high accuracy, it cannot identify new malicious URLs that are constantly created. With the development of artificial intelligence, machine learning methods have been applied to malicious URL detection. Basnet and Sung [2] have proposed a machine learning based approach to detect phishing Web pages. Kuyama *et al.* [3] have proposed a method to detect malicious domains by using support vector machine (SVM) and neural network with WHOIS and DNS information. Patil and Patil [4] have evaluated the performance of 6 decision tree learning algorithms J48 Decision Tree, Simple CART, Random Forest (RF), Random Tree, ADTree and REPTree for detecting malicious URLs on a balanced dataset. A combination of linear and non-linear space transformation methods has been applied to improve the performances of classifiers in identifying malicious URLs [5]. And the new features in their paper, which improved the efficiency of classifiers, were generated using five space transformation models (singular value decomposition, distance metric learning, Nyström methods, DML-NYS, and NYS-DML). Vinayakumar *et al.* [6] evaluated various deep learning architectures specifically recurrent neural network (RNN), identity-recurrent neural network (I-RNN), long short-term memory (LSTM), convolution neural network (CNN), and convolutional neural network-long short-term memory (CNN-LSTM) architectures by modeling the real known benign and malicious URL's in character-level language. Afzal *et al.* [7] proposed a hybrid deep-learning approach named URLdeepDetect for time-of-click URL analysis and classification to detect malicious URLs.

Chen and Guestrin [8] proposed a new ensemble learning method, which is Extreme Gradient Boosting (XGB). Compared with Gradient Boosted Decision Tree (GBDT), XGB adds a regular term to the objective function to prevent overfitting, and the speed of parallel processing of data is faster. XGB has advantages over other integrated learning algorithms in generalization performance, speed and

The associate editor coordinating the review of this manuscript and approving it for publication was Victor S. Sheng.

accuracy [9]–[11]. For instance, Wang *et al.* [12] established a type 2 diabetes classification model based on XGB, which had the best prediction effect than SVM, RF and K-Nearest Neighbor (KNN). Mahmud *et al.* [13] verified the reliability of the XGB classifier in the prediction of drug-target interaction.

In actual applications, it is found that the ratio of malicious URLs to benign URLs is unbalanced. If the imbalanced ratio between the number of malicious URLs and benign URLs is ignored, a biased classifier will be produced, causing the prediction results to be more inclined to the larger category. For example, in credit card fraud detection, it is maybe to have 99.9% of the customers without fraud and only 0.1% with it. If the model simply predicts everyone as 'no fraud', then the accuracy is 99.9%, which is remarkably high. However, missing to spot any fraud customer can lead to huge financial losses. The current common solutions are data augmentation and cost-sensitive learning [14]. Shen *et al.* [15] proposed a novel approach based on GANs to generate various mass images, which achieved an improvement of 5.03% in detection rate over the same model trained on original real lesion images. GANs are very famous for synthetic data generation, however, models designed for synthetic data generation have notable limitations [16]. Shaikh and Patil [17] proposed a role-based interactive model for data aggregation, in which tuning of privacy loss will be according to the role. The synthetic minority over-sampling technique (SMOTE) is the most common sampling method. Tan *et al.* [18] proposed that RF combined with the SMOTE is an effective solution to solve the problem of class imbalance and improve the performance of intrusion detection. Dong *et al.* [19] improved the classification effect of minority categories by using SMOTE in flotation method classification.

However, SMOTE increases label noise and the training time during the over-sampling process. The second method is cost-sensitive learning, which is adjusting the threshold toward classes to make misclassification of positive class examples harder for minimizing misclassification cost [20]–[23]. Jabeur *et al.* [24] confirmed cost-sensitive decision tree outperforms the other types of ensemble and single classifiers for bond rating prediction. A cost-sensitive convolution neural network (CSCNN) for imbalanced control chart pattern recognition (CCPR) problem, was proposed by Fuqua and Razzaghi [25]. And the performance of CSCNN on both simple and complex abnormal patterns is better than the existing CNN algorithm.

As abovementioned, we present a new method to detect malicious URLs with imbalanced data problem. The main contributions in this paper are as follows:

(1) The class imbalance issue in malicious URL detecting is addressed by CS-XGB method. Experiment results show that CS-XGB can greatly improve the malicious URLs' identification rate.

(2) The study of the relationship between the value of the cost-sensitive factor a of the G-mean has certain reference value for subsequent analysts on how to set the value of a when using the CS-XGB model.

(3) Compared with the SMOTE+XGB, which uses SMOTE to deal with the unbalanced data before establishing the XGB model, CS-XGB not only avoids the excessive time and space cost, but also the performance is good.

The rest of the paper is organized as follows. Section 2 discusses some methods used in our study. A description of the dataset and the feature set is provided in Section 3. Section 4 discusses the experimental results by comparing models. We conclude this paper in Section 5.

II. METHOD

A. EXTREME GRADIENT BOOSTING

XGB is an advanced Gradient Tree Boosting-based software that can efficiently handle large-scale Machine Learning tasks. Merited by its strong predictive performance and fast training speed, it has repeatedly won the top spot in Kaggle competitions [26].

The idea of this algorithm is to keep adding trees, and to keep splitting the features to grow a tree. You actually learn a new function to fit the last predicted residual when each time you add a tree. Letting x_i be the input, y_i be true label and z_i be the 'raw prediction' before the sigmoid function, according to [8], the objective function of the XGB model is:

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, Z_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + c \tag{1}$$

where $l(.,.)$ denotes the loss function, t stands for the tth tree, Ω penalizes the complexity of the model, $\Omega(f_t)$ represents the penalty term of regularization, and c is constant.

The second-order Taylor expansion is:

$$f(x + \Delta x) \approx f(x) + f'(x)\Delta x + 1/2 f''(x)\Delta x^2 \tag{2}$$

Taking equation (2) into equation (1), we can get

$$L^{(t)} \approx \sum_{i=1}^{n} [l(y_i + Z_i^{(t-1)}) + g_i f_t(x_i) + \tfrac{1}{2} h_i (f_t(x_i))^2] + \Omega(f_t) + c \tag{3}$$

where $g_i = \partial L/\partial z_i$, and $h_i = \partial^2 L/\partial z_i^2$. Removing the constant terms, we can obtain the following simplified objective at step t.

$$L^{(t)} \approx \sum_{i=1}^{n} [g_i f_t(x_i) + \tfrac{1}{2} h_i (f_t(x_i))^2] + \Omega(f_t) \tag{4}$$

In this objective function, g_i and h_i are required for fitting the XGB model.

For binary classification problems, the default loss function of XGB is the cross entropy (CE) loss:

$$L = -\sum_{i=1}^{n} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{5}$$

In equation (5), $\hat{y}_i = 1/[1 + \exp(-z_i)]$, that is sigmoid is selected as activation. Therefore, we can get:

$$\partial \hat{y}_i / \partial z_i = \hat{y}_i (1 - \hat{y}_i) \tag{6}$$

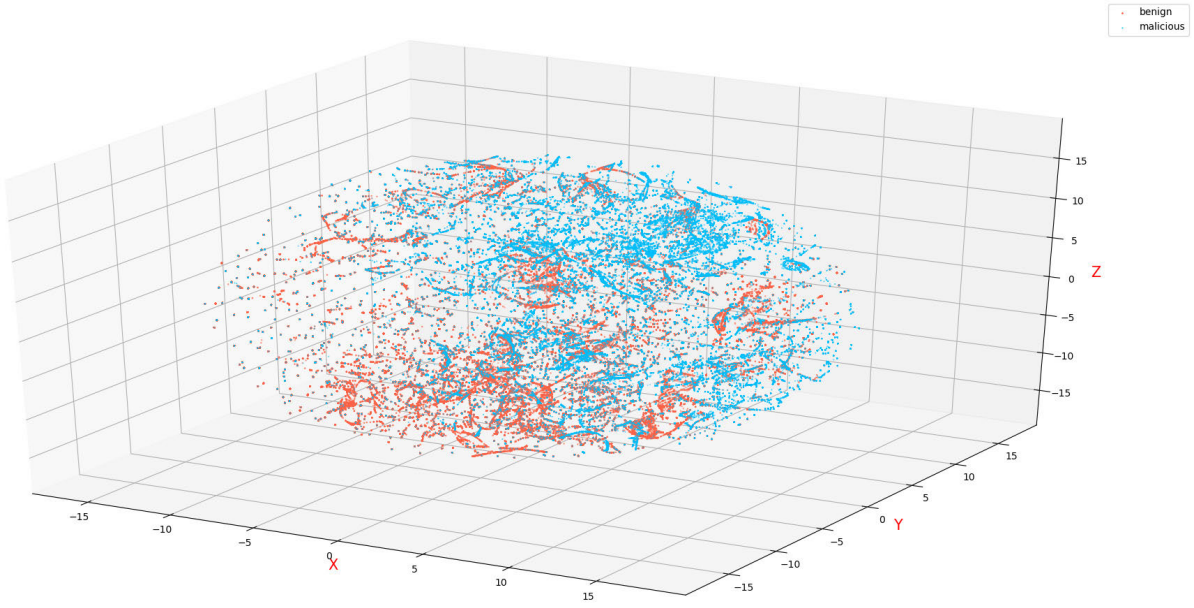

FIGURE 1. Visualization of dataset by t-SNE.

B. COST-SENSITIVE EXTREME GRADIENT BOOSTING

In most of the classification algorithms, there is no difference between the cost of the correctly classified and misclassified examples and they only focus on minimizing the loss function. The core idea of cost-sensitive learning is that the error caused by the positive sample of a misclassified class is given a larger weight in the loss function, so that positive samples are given more attention in the process of learning the model. A cost matrix has given in Table 1 for binary classification.

TABLE 1. Two-class cost matrix.

	Actual malicious (Positive)	Actual benign (Negative)
Predict malicious (Positive)	TP (cost= c_{11})	FP (cost= c_{01})
Predict benign (Negative)	FN (cost= c_{10})	TN (cost= c_{00})

In the cost-sensitive extreme gradient boosting (CS-XGB) model, we only consider the misclassification case. So let $c_{00} = c_{11} = 0$, $c_{10} = a(a > 0)$, and $c_{01} = 1$, the loss function with cost-sensitive factor can be denoted as follows:

$$L_a = -\sum_{i=1}^{n} [a y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (7)$$

where a indicates the cost-sensitive factor. It is not hard to see extra loss will be counted on False Negative (FN) when a is greater than 1; On the contrary, if a is less than 1, extra loss will be counted on False Positive (FP).

The first-order derivative g_i and the second-order derivative h_i are presented as follows:

$$g_i = \partial L_a / \partial z_i = \hat{y}_i(1 - y_i + a y_i) - a y_i \quad (8)$$

$$h_i = \partial^2 L_a / \partial z_i^2 = \hat{y}_i(1 - \hat{y}_i)(1 - y_i + a y_i) \quad (9)$$

TABLE 2. Dataset information.

Dataset	Malicious	Benign	Total
Number	200,000	400,000	600,000

C. SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE

SMOTE is a classic oversampling method proposed by Chawla *et al.* [27] in 2002. The idea of the SMOTE is to analyze minority samples and synthesize new samples based on minority samples and add them to the dataset. SMOTE works by taking a random point between the minority class sample and its k-nearest neighbors as a new synthetic sample.

III. MATERIALS

A. DATASET COLLECTION

URLs were collected from a crawler deployed in our lab. We randomly selected 600,000 URLs as a dataset from the data collection. A large number of malicious URLs appear every day, which is attributed to the mature URL generation algorithm. This study presents an ordinary volume in real applications. The dataset information is showed in Table 2.

B. FEATURES EXTRACTION

By analyzing the dataset we collected, it can be found that malicious URLs often have certain commonalities. Based on these commonalities, relevant features can be extracted and used for machine learning models. We have extracted 28 features. These features are important to identify malicious URLs from benign URLs. The categories of features contain domain name features, WHOIS information, geographic information and suspicious words based features. The mean

and standard deviation of some features in both groups are given in Table 3. In Table 3, it is clear to see the deviations between the malicious URL and benign URL. For example, the number of digits in the domain name, which is in malicious URLs generally more than benign URLs.

To make the features easier to understand, we use the t-Distributed Stochastic Neighbor Embedding (t-SNE) techniques to reduce the dimension of the data and illustrated them in Figure 1. It shows that data points with malicious label are mostly concentrated in the upper right hemisphere, and benign samples are mostly in the lower left hemisphere.

C. METRICS

There are many evaluation criteria for classification models. The classification performance can be clarified in relation to the confusion matrix described in Table 1. To test and evaluate the algorithms we use 10-fold cross-validation in this paper. In this process, the dataset is divided into 10 subsets. Each time, one of the 10 subsets is used as the test set and the other 9 subsets form the training set.

For unbalanced classification problems, the area under the curve (AUC) and G-mean are usually chosen as the key performance indexes. Sensitivity, which also called the true positive rate, measures the proportion of positives that are correctly identified [28]. The purpose of the models is to improve malicious URLs detection, we also evaluated sensitivity. The AUC measures the area under this curve, where the higher AUC the better [29]. The measure of sensitivity calculates the relative accuracy of malicious URLs class.

$$sensitivity = TP/(TP + FN) \tag{10}$$

Also, G-mean is a popular evaluation index in general as it integrates the recalls of all categories [30]. The higher G-mean value, the better the comprehensive performance of a classifier. The G-mean is calculated using the following equation:

$$G - mean = [(TP/(TP+FN)) \times (TN/(TN+FP))]^{1/2} \tag{11}$$

IV. EXPERIMENTS

We perform all experiments on an Intel(R) Core(TM) i5-8265U @1.60 GHz processor and 8GB of RAM in a 64-bit platform. All algorithms are implemented in Python version 3.7. And all experimental results are obtained by 10-fold cross-validation. Figure 2 shows the flowchart of the experimental design. In order to verify the validity and effectiveness of the proposed CS-XGB model, 18 datasets with different degrees of imbalance are constructed. We let ρ denote the imbalanced class ratio as the ratio of the number of malicious samples over the total number of samples, i.e., $\rho = P/(N + P)$, where N is the number of benign group and P is the number of malicious group. Table 4 shows the information and ρ of these datasets selected in this paper.

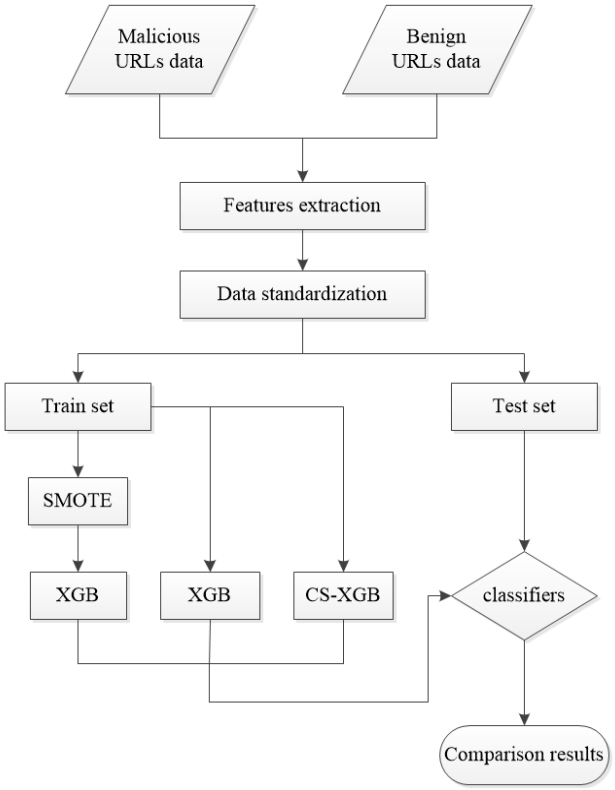

FIGURE 2. The flowchart of the experimental design.

A. THE CONNECTION BETWEEN COST-SENSITIVE FACTOR a AND G-MEAN

In this section, the main discussion is the impact of cost-sensitive factor a on the results of the CS-XGB model. To provide observation of the performance improvement particularly by the cost-sensitive factor a, we used the same parameter set for different a in the process of training the model. Conducting experiments on the 18 datasets list in Table 5, the result is shown in Figure 3.

In Figure 3, some clear regulations are as follows:

① For all datasets with different imbalanced class ratios ρ, the general trend of G-mean is that as a increases, it first goes up, then stabilizes, and then falls again. At the same time, the smaller the imbalanced class ratio ρ, the more serious imbalance, the longer the intermediate stationary phase.

② For datasets of the same ρ but of different volumes, the trend of G-mean is almost consistent.

③ For all datasets with different ρ except C6, G-mean reaches its highest point for the first time near $a = 1/\rho$.

B. COMPARISON OF EXPERIMENTAL RESULTS

In this section, we compare the results of CS-XGB and SMOTE+XGB with XGB. The results are shown in Table 5. For all datasets in Table 4, CS-XGB has high scores in AUC, sensitivity and G-mean. It seems that the ability to detect malicious URLs of our model is effective. It is not difficult to find using SMOTE oversampling can reduce the imbalance

TABLE 3. The mean and standard deviation of some features.

Feature name	Description	Type	Benign Mean±SD	Malicious Mean±SD	Total Mean±SD
dm_digit_mix_letter	The domain name is a mixture of numbers and letters	binary	0.0526±0.2232	0.3291±0.4699	0.1448±0.3519
dm_digit_cnt	Number of digits in the domain name	numeric	0.1153±0.5495	3.5398±4.6040	1.2568±3.1421
dm_digit_pnt	Percentage of numbers in domain names	numeric	0.0152±0.0750	0.3516±0.3702	0.1273±0.2731
dm_vowel_cnt	Number of vowels in the domain name	numeric	3.6609±1.9198	1.4268±1.8479	2.9162±2.1690
subdm_len	Subdomain length	numeric	0.5892±3.0411	1.1435±4.6745	0.7739±3.6766
dm_regt_tm_flag	The domain name registration time is less than 2 years	binary	0.4014±0.4902	0.7484±0.4339	0.5170±0.4997

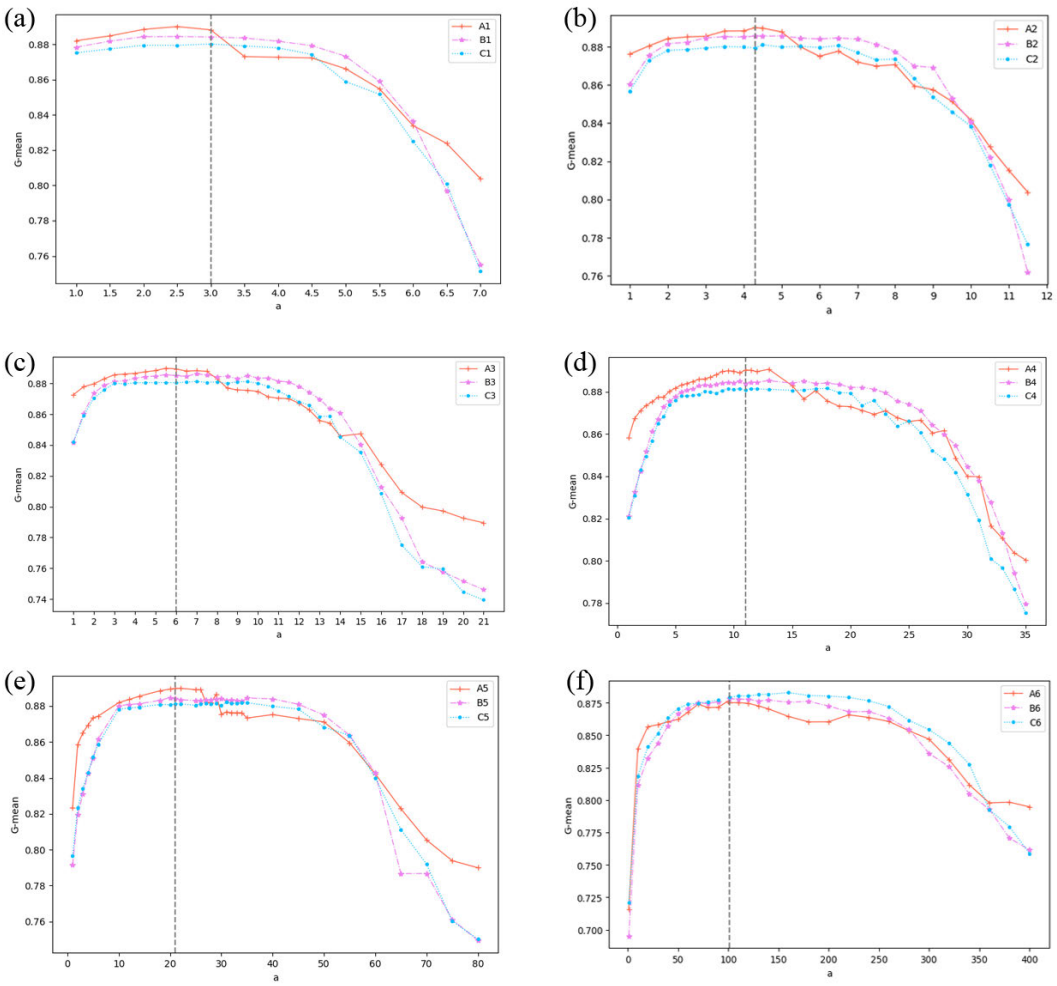

FIGURE 3. The G-mean due to the change of *a* in the CS-XGB model.

of the data. Although the performance of SMOTE+XGB is not as good as CS-XGB, it is better than only XGB. The more serious imbalance of the dataset, the lower values of AUC, sensitivity and G-mean of the XGB model, which is in line with the real situation. However, CS-XGB can narrow the gap caused by different levels of imbalance. For example, in all the data, the sensitivity of XGB ranges from 0.4845 to 0.7969, while the sensitivity of CS-XGB ranges from 0.8055 to 0.8547. In order to see the comparison results of sensitivity in each model more intuitively, we plot these data into a column chart, as shown in Figure 4.

To further evaluate the performance of CS-XGB, we introduce another imbalanced dataset from Kaggle: creditcard (https://www.kaggle.com/mlg-ulb/creditcardfraud).

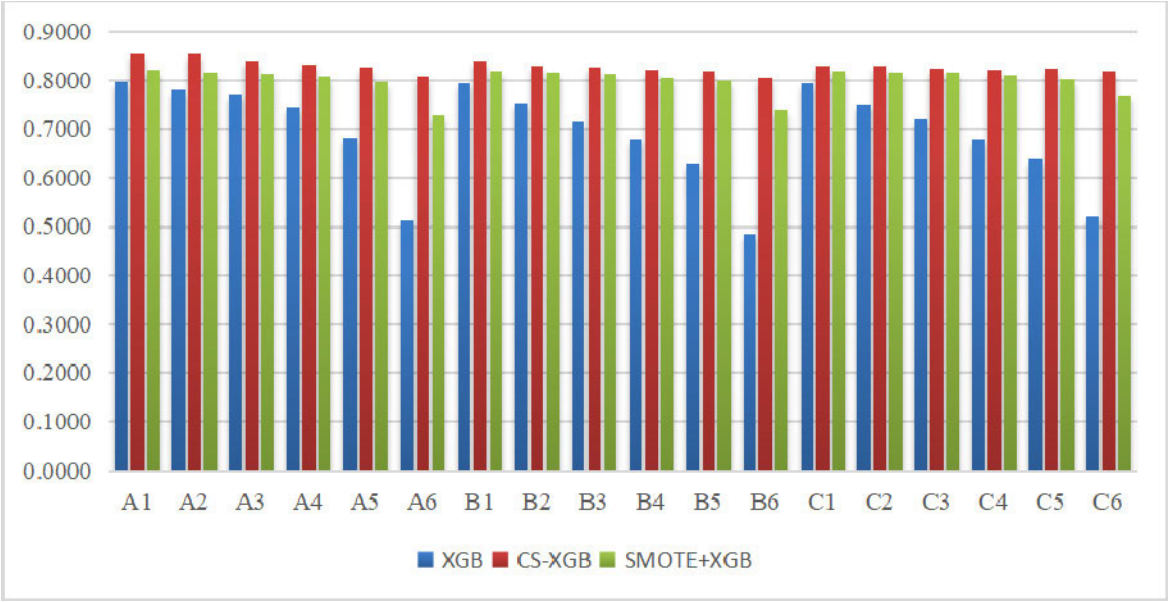

FIGURE 4. Sensitivity results of comparison.

The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The imbalanced class ratio ρ of creditcard dataset is approximately 1/579. Slightly different from the experiments on all 18 URLs' datasets, we divide the creditcard dataset into 9:1 as training and a testing set, i.e., 90% for training and 10% for testing, on the grounds that the dataset includes a small number of fraud instances.

The classification result is presented in Table 6, with AUC, G-mean, and sensitivity. Our model CS-XGB seems to get the best results that got high scores in three metrics (99.05% in AUC, 96.70% in G-mean, 93.88% in sensitivity).

C. DISCUSSION

Experiments based on eighteen imbalanced URL classification datasets are conducted with competitive performances illustrated. Experimental results in Table 5 demonstrate that the CS-XGB model is effective for detecting malicious URLs. Among all the results, the least increase in sensitivity is from 0.7932 to 0.8289, and the most is from 0.4845 to 0.8055. In terms of AUC and G-mean, which are considered the most crucial metrics for imbalanced classification, CS-XGB outperforms XGB and SMOTE+XGB by a large margin. As the imbalance ratio goes up, the improvements on G-mean and sensitivity for CS-XGB become more significant. And while sensitivity is improved, AUC is also slightly improved. Although the AUC, sensitivity and G-mean values of the SMOTE+XGB model are higher than those of the XGB model, they are lower than those of the CS-XGB model. We use a total of 600,000 real URL samples. Although large samples make the model relatively more stable, the disadvantage is that the model training time will be relatively increased, and the up-sampling process will greatly increase the training time.

To further demonstrate the effectiveness of our proposed CS-XGB model, as shown in Table 6, we compare with

TABLE 4. The information of 18 datasets.

Dataset	Total numbers	Benign (N)	Malicious (P)	N:P	ρ
A1	150000	100000	50000	2：1	1/3
A2	130000	100000	30000	10：3	3/13
A3	120000	100000	20000	5：1	1/6
A4	110000	100000	10000	10：1	1/11
A5	105000	100000	5000	20：1	1/21
A6	101000	100000	1000	100：1	1/101
B1	300000	200000	100000	2：1	1/3
B2	260000	200000	60000	10：3	3/13
B3	240000	200000	40000	5：1	1/6
B4	220000	200000	20000	10：1	1/11
B5	210000	200000	10000	20：1	1/21
B6	202000	200000	2000	100：1	1/101
C1	600000	400000	200000	2：1	1/3
C2	520000	400000	120000	10：3	3/13
C3	480000	400000	80000	5：1	1/6
C4	440000	400000	40000	10：1	1/11
C5	420000	400000	20000	20：1	1/21
C6	404000	400000	4000	100：1	1/101

TABLE 5. Performance evaluation of each model.

dataset	XGB			CS-XGB			SMOTE+XGB		
	AUC	G-mean	Sensitivity	AUC	G-mean	Sensitivity	AUC	G-mean	Sensitivity
A1	0.9529	0.8821	0.7969	**0.9534**	**0.8914**	**0.8547**	0.9532	0.8882	0.8213
A2	0.9532	0.8762	0.7813	**0.9537**	**0.8912**	**0.8536**	0.9533	0.8862	0.8160
A3	0.9531	0.8727	0.7732	**0.9533**	**0.8917**	**0.8392**	0.9531	0.8854	0.8124
A4	0.9536	0.8583	0.7445	**0.9539**	**0.8908**	**0.8314**	0.9525	0.8840	0.8078
A5	0.9498	0.8234	0.6818	**0.9517**	**0.8901**	**0.8258**	0.9505	0.8796	0.7970
A6	0.9236	0.7156	0.5140	**0.9414**	**0.8781**	**0.8070**	0.9336	0.8448	0.7290
B1	0.9413	0.8785	0.7940	**0.9435**	**0.8849**	**0.8403**	0.9427	0.8843	0.8193
B2	0.9397	0.8606	0.7531	**0.9427**	**0.8856**	**0.8300**	0.9425	0.8832	0.8166
B3	0.9381	0.8417	0.7157	**0.9431**	**0.8858**	**0.8262**	0.9417	0.8831	0.8144
B4	0.9345	0.8211	0.6780	**0.9435**	**0.8850**	**0.8224**	0.9414	0.8798	0.8043
B5	0.9283	0.7916	0.6288	**0.9420**	**0.8847**	**0.8197**	0.9395	0.8785	0.7995
B6	0.9161	0.6952	0.4845	**0.9361**	**0.8781**	**0.8055**	0.9278	0.8494	0.7405
C1	0.9391	0.8752	0.7932	**0.9406**	**0.8805**	**0.8289**	0.9404	0.8795	0.8187
C2	0.9378	0.8569	0.7514	**0.9407**	**0.8810**	**0.8291**	0.9397	0.8795	0.8155
C3	0.9367	0.8419	0.7207	**0.9404**	**0.8811**	**0.8250**	0.9402	0.8798	0.8154
C4	0.9302	0.8206	0.6802	**0.9412**	**0.8815**	**0.8223**	0.9393	0.8788	0.8101
C5	0.9273	0.7965	0.6386	**0.9408**	**0.8818**	**0.8232**	0.9385	0.8761	0.8035
C6	0.9205	0.7209	0.5210	**0.9397**	**0.8803**	**0.8175**	0.9339	0.8615	0.7705

TABLE 6. Performance evaluation on the creditcard dataset.

Model	AUC	G-mean	Sensitivity
XGB	0.9783	0.6388	0.4082
CS-XGB	**0.9905**	**0.9670**	**0.9388**
SMOTE+XGB	0.9843	0.9554	0.9184

several methods using the creditcard dataset. Compared with XGB and SMOTE+XGB, our model has demonstrated its superiority, especially in improving the sensitivity and G-mean.

V. CONCLUSION

In this paper, a new methodology CS-XGB was put forward to identify malicious URLs despite the presence of serious category imbalance. The research demonstrated the feasibility of CS-XGB and the effectiveness of the combination of XGB and SMOTE oversampling. The CS-XGB method has a high score overall AUC, G-mean and sensitivity. What is more, the performance of CS-XGB model was better than XGB and SMOTE+XGB according to the different evaluation criteria. And the impact of cost-sensitive factor a on the results of the CS-XGB model has reference value for subsequent analysts. On balance, the identification method of malicious URLs proposed in this paper is novel, simple and promising. We are continuously collecting malicious URLs, and the amount of this data will become larger and larger, which is a challenge for future modeling. At the same time, enhanced privacy preservation and security can be played more attention [31]. The future research direction is to explore the improvements of classifiers in terms of training time and the rate of malicious URLs' identification. We will also try to explore other techniques like explainable AI, reinforcement learning, domain transfer learning, multi model learning in the future. Other data augmentation techniques like GANS and variational auto encoders or python library like NLPAug can be taken into consideration.

REFERENCES

[1] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "PhishNet: Predictive blacklisting to detect phishing attacks," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, Mar. 2010, pp. 1–5.

[2] R. B. Basnet and A. H. Sung, "Learning to detect phishing Web pages," *J. Internet Serv. Inf. Secur.*, vol. 4, no. 3, pp. 21–39, 2014.

[3] M. Kuyama, Y. Kakizaki, and R. Sasaki, "Method for detecting a malicious domain by using whois and dns features," in *Proc. 3rd Int. Conf. Forensics*, Kuala Lumpur, Malaysia, 2016, pp. 74–80.

[4] D. R. Patil and J. B. Patil, "Malicious URLs detection using decision tree classifiers and majority voting technique," *Cybern. Inf. Technol.*, vol. 18, no. 1, pp. 11–29, Mar. 2018.

[5] T. Li, G. Kou, and Y. Peng, "Improving malicious URLs detection Via feature engineering: Linear and nonlinear space transformation methods," *Inf. Syst.*, vol. 91, Jul. 2020, Art. no. 101494.

[6] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Evaluating deep learning approaches to characterize and classify malicious URL's," *J. Intell. Fuzzy Syst.*, vol. 34, no. 3, pp. 1333–1343, Mar. 2018.

[7] S. Afzal, M. Asim, A. R. Javed, M. O. Beg, and T. Baker, "URLdeep-Detect: A deep learning approach for detecting malicious URLs using semantic vector models," *J. Netw. Syst. Manage.*, vol. 29, no. 3, pp. 1–27, Mar. 2021, doi: 10.1007/s10922-021-09587-8.
[8] I. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf.*, San Francisco, CA, USA, 2016, pp. 785–794.
[9] J. Fan, W. Yue, L. Wu, F. Zhang, H. Cai, X. Wang, X. Lu, and Y. Xiang, "Evaluation of SVM, ELM and four tree-based ensemble models for predicting daily reference evapotranspiration using limited meteorological data in different climates of China," *Agricult. Forest Meteorol.*, vol. 263, pp. 225–241, Dec. 2018.
[10] J. Nobre and R. F. Neves, "Combining principal component analysis, discrete wavelet transform and XGBoost to trade in the financial markets," *Expert Syst. Appl.*, vol. 125, pp. 181–194, Jul. 2019.
[11] R. Wang, S. Lu, and Q. Li, "Multi-criteria comprehensive study on predictive algorithm of hourly heating energy consumption for residential buildings," *Sustain. Cities Soc.*, vol. 49, Aug. 2019, Art. no. 101623.
[12] L. Wang, X. Wang, A. Chen, X. Jin, and H. Che, "Prediction of type 2 diabetes risk and its effect evaluation based on the XGBoost model," *Healthcare*, vol. 8, no. 3, p. 247, Jul. 2020.
[13] S. M. H. Mahmud, W. Chen, H. Jahan, Y. Liu, N. I. Sujan, and S. Ahmed, "IDTi-CSsmoteB: Identification of drug–target interaction based on drug chemical structure and protein sequence using XGBoost with over-sampling technique SMOTE," *IEEE Access*, vol. 7, pp. 48699–48714, 2019.
[14] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 463–484, Jul. 2012.
[15] T. Shen, K. Hao, C. Gou, and F.-Y. Wang, "Mass image synthesis in mammogram with contextual information based on GANs," *Comput. Methods Programs Biomed.*, vol. 202, Apr. 2021, Art. no. 106019.
[16] S. K. J. Rizvi, M. A. Azad, and M. M. Fraz, "Spectrum of advancements and developments in multidisciplinary domains for generative adversarial networks (GANs)," in *Proc. Arch. Comput. Methods Eng.*, Apr. 2021, pp. 11–19, doi: 10.1007/s11831-021-09543-4.
[17] A. Shaikh and S. Patil, "A survey on privacy enhanced role based data aggregation via differential privacy," in *Proc. Int. Conf. Advance Commun. Comput. Technol., Amrutvahini College Eng.*, Sangamner, India, Feb. 2018, pp. 285–290.
[18] X. Tan, S. Su, Z. Huang, X. Guo, Z. Zuo, X. Sun, and L. Li, "Wireless sensor networks intrusion detection based on SMOTE and the random forest algorithm," *Sensors*, vol. 19, no. 1, p. 203, Jan. 2019.
[19] H. Dong, D. He, and F. Wang, "SMOTE-XGBoost using tree Parzen estimator optimization for copper flotation method classification," *Powder Technol.*, vol. 375, pp. 174–181, Sep. 2020.
[20] J. P. Bradford, C. Kunz, R. Kohavi, C. Brunk, and C. E. Brodley, "Pruning decision trees with misclassification costs," in *Proc. 10th Eur. Conf. Mach. Learn.*, Berlin, Germany, 1998, pp. 131–136.
[21] C. Elkan, "The foundations of cost-sensitive learning," in *Proc. 17th Int. Joint Con. Artif. Intell.*, Seattle, WA, USA, 2001, pp. 973–978.
[22] J. Langford and A. Beygelzimer, "Sensitive error correcting output codes," *Lect. Notes Comput. Sci.*, vol. 2459, no. 3, pp. 158–172, 2005.
[23] Y. Sahin, S. Bulkan, and E. Duman, "A cost-sensitive decision tree approach for fraud detection," *Expert Syst. Appl.*, vol. 40, no. 15, pp. 5916–5923, Nov. 2013.
[24] S. B. Jabeur, A. Sadaaoui, A. Sghaier, and R. Aloui, "Machine learning models and cost-sensitive decision trees for bond rating prediction," *J. Oper. Res. Soc.*, vol. 71, no. 8, pp. 1161–1179, Aug. 2020.
[25] D. Fuqua and T. Razzaghi, "A cost-sensitive convolution neural network learning for control chart pattern recognition," *Expert Syst. Appl.*, vol. 150, Jul. 2020, Art. no. 113275.
[26] D. Nielsen, "Tree boosting with XGBoost—Why does XGBoost win 'Every' machine learning competition?" M.S. thesis, Norwegian Univ. Sci. Technol., Trondheim, Norway, Dec. 2016.
[27] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
[28] A. Srivastava, S. Jain, R. Miranda, S. Patil, S. Pandya, and K. Kotecha, "Deep learning based respiratory sound analysis for detection of chronic obstructive pulmonary disease," *PeerJ Comput. Sci.*, vol. 7, p. e369, Feb. 2021, doi: 10.7717/peerj-cs.369.
[29] T. Fawcett, "An introduction to Roc analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
[30] M. Kubat, R. C. Holte, and S. Matwin, "Machine learning for the detection of oil spills in satellite radar images," *Mach. Learn.*, vol. 30, nos. 2–3, pp. 195–215, 1998.
[31] S. Pati and S. Josh, "Improved privacy preservation of personal health records via tokenizationl," *Int. J. Pure Appl. Math.*, vol. 118, no. 18, pp. 3035–3045, 2018.

SHEN HE was born in Beijing, China, in 1980. He received the Ph.D. degree in computer application from the University of Science and Technology of China, in 2007. He is currently the Director of the Department of Security Technology, China Mobile Research Institute. His main research interests include network and information security, mobile communication security and trusted computing, blockchain technology, and quantum communications.

BANGLING LI was born in Baoding, Hebei, China, in 1990. She received the M.S. degree in statistics from the Hebei University of Technology, in 2017. She is currently working as a Security Researcher with the China Mobile Research Institute. Her main research interests include terminal security, network and information security, and artificial intelligence.

HUAXI PENG received the Ph.D. degree in computer science from the Institute of Software Chinese Academy of Sciences, in 2007. He is currently a Senior Engineer with the Department of Security Technology, China Mobile Research Institute. His main research interests include network and information security, mobile communication security, and the IoT security.

JUN XIN received the master's degree in computer science from the Beijing University of Posts and Telecommunications, in 2014. She has been working with the China Mobile Research Institute. Her research interests include terminal security, and network and information security.

ERPENG ZHANG received the master's degree in computer science from Harbin Engineering University, in 2006. He joined the China Mobile Research Institute, in 2011. His research interests include terminal security, and network and information security.