Mathematical Biosciences and Engineering

MBE, 20(8): 14959–14977.
DOI: 10.3934/mbe.2023670
Received: 16 March 2023
Revised: 20 June 2023
Accepted: 04 July 2023
Published:12 July 2023

http://www.aimspress.com/journal/MBE

Research article

CDBC: A novel data enhancement method based on improved between-class learning for darknet detection

Binjie Song[1,*], Yufei Chang[2,*], Minxi Liao[3], Yuanhang Wang[1], Jixiang Chen[2] and Nianwang Wang[1,*]

[1] Academy of A&AD, Zhengzhou 450000, China
[2] South China University of Technology, Guangzhou 511400, China
[3] State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China

* **Correspondence:** Email: 201720139968@mail.scut.edu.cn, web0413@163.com.

Abstract: With the development of the Internet, people have paid more attention to privacy protection, and privacy protection technology is widely used. However, it also breeds the darknet, which has become a tool that criminals can exploit, especially in the fields of economic crime and military intelligence. The darknet detection is becoming increasingly important; however, the darknet traffic is seriously unbalanced. The detection is difficult and the accuracy of the detection methods needs to be improved. To overcome these problems, we first propose a novel learning method. The method is the Chebyshev distance based Between-class learning (CDBC), which can learn the spatial distribution of the darknet dataset, and generate "gap data". The gap data can be adopted to optimize the distribution boundaries of the dataset. Second, a novel darknet traffic detection method is proposed. We test the proposed method on the ISCXTor 2016 dataset and the CIC-Darknet 2020 dataset, and the results show that CDBC can help more than 10 existing methods improve accuracy, even up to 99.99%. Compared with other sampling methods, CDBC can also help the classifiers achieve higher recall.

Keywords: between-class learning; darknet, detection; traffic classification

1. Introduction

With the development of the network, users' awareness of privacy protection has been continuously improved, and many users choose to use anonymous communication tools to access the Internet to prevent their privacy from being compromised while surfing [1−3]. Anonymity service such as the second generation onion router (Tor) [4−6], invisible internet project (I2P) [7], Freenet [8,9] and ZeroNet [10] can provide a high degree of anonymity and become an important means of protecting privacy on the Internet. However, these tools can also provide protection for illegal users, which brings difficulties to network supervision. For example, many illegal users use anonymous communication tools to conduct illegal transactions on the darknet. As most people know, darknet [10] is defined as a restricted access network. Common conditions that need to be met are special settings, specific software, authorization or non-standard protocols or port access. Nowadays, there are many types of darknet, and they have gradually become platforms for terrorism and crime [12]. From the perspective of network management, to monitor and even prevent possible illegal activities on the darknet, it is essential to detect the activities of users and is necessary to improve the detection capability. However, in the existing datasets of darknet traffic, the amount and types of darknet traffic are scarce. The detection accuracy is not high enough. To detect a small amount of darknet traffic and its type, we propose CDBC, and based on it, we propose a novel darknet traffic detection method.

The contributions of this paper are summarized as follows.

(1) To solve the problem of the small amount of the darknet traffic. The experiment takes the darknet traffic as a small sample of data, and the CDBC is proposed, which can learn the spatial distribution of the darknet datasets and generate gap data around the small samples to reduce the impact of data imbalance.

(2) To the best of our knowledge, it is the first time that Between-class learning is adopted to solve multi-classification problems, and good results are achieved.

(3) The proposed method enhances the capability of darknet detection, by federating CDBC with over 10 classifiers respectively. Experimental results show that the detection method based on CBDC and random forest achieves an accuracy of 99.99%.

The structure of the paper is arranged as follows. In section II, we mainly introduce darknet detection and Between-class learning. The proposed method is introduced in detail in section III. In section IV, we mainly present the experimental results and analyze them. Finally, the conclusions and prospects of the method we proposed are given.

2. Related work

2.1. Darknet detection

Darknet detection can be regarded as a special encrypted traffic detection problem. This section introduces some research work related to darknet traffic detection.

2.1.1. The methods based on machine learning

In 2016, Draper-Gil et al. [13] proposed an encrypted traffic detection method based on time

series analysis. The proposed method adopts decision tree (DT) and K-nearest neighbor (KNN) to detect VPN traffic according to different types of traffic, and the detection accuracy is 80%. In 2018, Montieri et al. [1] used machine learning methods such as naive Bayes (NB), random forest (RF) to classify the Anon17 darknet dataset according to different anonymity tools (Tor, I2P and JonDonym), and the reached more than 75%. In 2020, Hu et al. [14] collected a real darknet dataset, including Tor, I2P, ZeroNet and Freenet. Moreover, experiments are conducted on the basis of feature selection and multiple classifiers. The detection accuracy for the types of darknet traffic is 96.9%, and the average detection accuracy for the type of application is 91.6%. In 2021, Rawat et al. [15] applied the term frequency-inverse document frequency (TF-IDF) algorithm in the field of text data mining the darknet traffic detection task, and then detected the darknet based on the LightGBM algorithm. The accuracy is more than 98%. In 2022, Abu et al. [16] proposed a method for detecting darknet traffic based on machine learning, and experiments were performed on the CIC-Darknet-2020 dataset [17]. The authors merged VPN and Tor, and finally, the results showed the accuracy of 99.50%. However, the above detection accuracy still needs to be improved, and they did not pay attention to the influence of the dataset distribution.

2.1.2. The methods based on deep learning

Compared with traditional machine learning methods, the methods based on deep learning can automatically learn features of the traffic. Recently, detection methods based on deep learning have made some progress. In 2019, Liu et al. [18] applied recurrent neural networks (RNN) to encrypted traffic detection and proposed the FS-Net method, which is based on an end-to-end classification. By learning effective features and reconstructing the network, the method mines sequence features, and the feature learning ability are enhanced. In 2020, Habibi et al. [19] proposed a method named DeepImage, which first selects features and generates two-dimensional grayscale images and then uses two-dimensional convolutional neural networks (CNN) to detect darknet traffic. The experimental results showed that the accuracy of the method is 86%. In the same year, Lotfollahi et al. [19] proposed a method called Deep Packet. This method is an automated framework for network traffic feature extraction based on one-dimensional CNN and stacked autoencoders (SAE). The detection accuracy of darknet traffic reaches 98%, and the accuracy of darknet application types reaches 93%. In 2020, Wang et al. [20] proposed an end-to-end method named App-Net, which learns the joint features of traffic and applications by combining RNN and CNN. Finally, annotations for flow sequences and specific applications are simultaneously implemented. In 2021, Sarwar et al. [21] proposed a novel darknet detection method, which improved CNN-LSTM and CNN-GRU. The results showed that the accuracy was 96%. Obviously, the accuracy of the methods based on deep learning is not high enough, and they didn't consider the spatial distribution of small samples in the dataset, which affects the detection.

2.2. Between-class learning

The idea of the learning method mainly comes from classification and the recognition of pictures, the sound recognition, etc. [22−24]. Initially, Between-class learning is adopted in sound recognition. It mixes data of two different types in random proportions to generate new data. The new data will be considered adoption data and will be used in the experiment. Tokozume et al. [25]

proposed a new deep sound recognition network (EnvNet-v2) based on Between-class learning. In the experiment, the authors mixed two different sounds, created new sounds and used the synthetic dataset to train the model and output the mix ratio. Gao et al. [26] improved Between-class learning and proposed a novel method for anomaly detection, named EBC learning. This method calculates the Euclidean distance before mixing, and then mixes the data with similar distances. Finally, RF is used for detection. However, this method can only solve binary classification problems.

3. Proposed methodology

In this section, we will introduce the method of darknet detection in detail, which consists of three aspects, data preprocessing, CDBC and detection. The detection framework is shown in Figure 1.

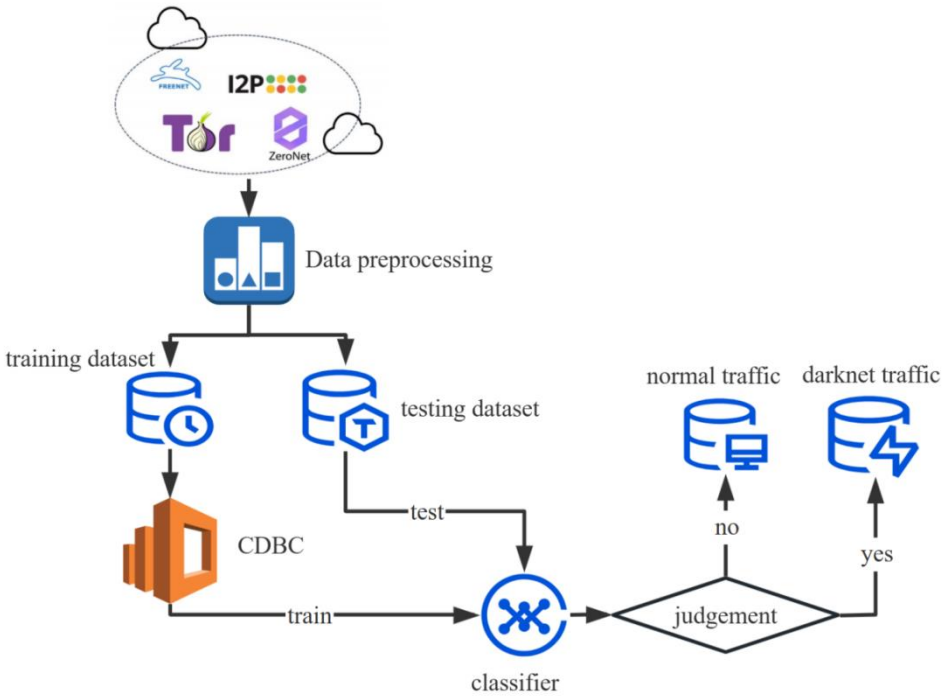


Figure 1. The darknet traffic detection framework.

3.1. Data preprocessing

In data preprocessing, vectorization, normalization and One-hot are adopted to process the original dataset. Simultaneously, dimensionality reduction is performed on high-dimensional data, which can remove redundant features and retain the most relevant features. It can improve detection accuracy and training efficiency. The dataset has non-numeric features, it needs to be vectorized. We remove some features which cannot be processed. Additionally, IP addresses cannot be processed and calculated as numerical values; so we perform frequency processing on IP addresses. The number of occurrences of the IP address is taken as the characteristic value. For non-numeric timestamp data, we replace it with the number of occurrences in a day or in an hour. For "inf" and "Nan" values in the dataset, we take the average of the features. We adopt a normalization method to

normalize the feature values and scale them to [0, 1]. The calculation is as follows:

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

where x represents the original feature, x_{max} and x_{min} represent the maximum and minimum features. The experiment uses One-hot to label the data. For example, there are four labels which can be represented as [[0, 0, 0, 1], [0, 0, 1, 0], [0, 1, 0, 0], [1, 0, 0, 0]]. In our experiments, we have 8 labels at most.

3.2. CDBC

The main idea of CDBC is to generate gap data around unbalanced traffic to enhance the distribution boundaries between different types of traffic. It is important to stress that gap data is not any kind of traffic, but a kind of data between darknet traffic and normal traffic, which is distributed exactly in between them. It is shown in Figure 2. Compared with common methods, CDBC can optimize detection by focusing only on a little traffic. Therefore, the algorithm has obvious advantages.

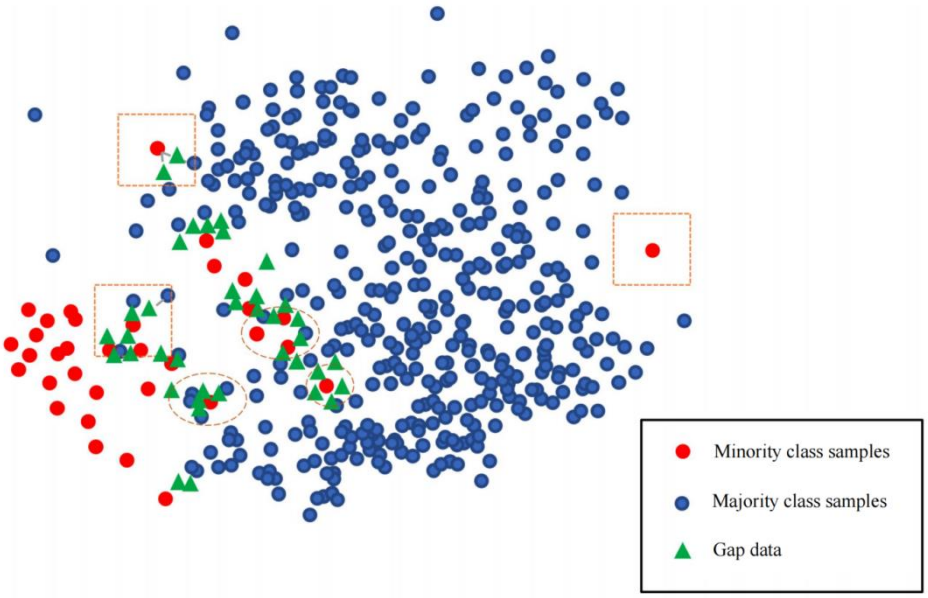


Figure 2. Example of CDBC solving binary classification detection.

As shown in Figure 2, CDBC finds the *k*-nearest neighbors of various types of traffic by calculating the Chebyshev distance, and generates gap data between the neighbors. When the training traffic distribution is unbalanced, CDBC can significantly improve the ability of the classifier to identify small samples.

In the experiment, we adopt Chebyshev distance as a metric in multi-classification problems, because it can highlight the difference between traffic, and the calculation equation is as follows:

$$D_{Chebyshev}(x_i, x_j) = \max_l\left(\left|x_i^l - x_j^l\right|\right) = \lim_{k\to\infty}\left(\sum_{l=1}^{n}\left|x_i^l - x_j^l\right|^k\right)^{1/k} \tag{2}$$

where x^l represents the l-th dimension of the features, $D_{Chebyshev}(x_i, x_j)$represents the maximum distance between the traffic x_i and x_j in the l-th dimension of the features.

CDBC generates gap data by randomly mixing two different types of traffic. The equation is as follows:

$$x_{new} = gap \times x_i + (1 - gap) \times x_j \tag{3}$$

where gap is randomly generated, and $gap \in U(0,1)$.

When CDBC is applied to the binary classification of darknet detection, there are two kinds of labels, where "0" represents the non-darknet traffic label, and "1" represents the darknet traffic label. The label determination method of the generated samples is as follows: One-hot is used to label the gap data. In the experiment, it is represented by the distance of the gap data to the two samples respectively. For example, the labels of minority class samples x_i and majority class samples x_j are represented by One-hot as $[0.,1.]$ and $[1.,0.]$ respectively, and the labels of gap data can be expressed as $[1 - gap, gap]$, the equation is as follows:

$$\left(d_i^{new}, d_j^{new}\right) = \left(\frac{\max_l\left(\left|x_{new}^l - x_i^l\right|\right)}{\max_l\left(\left|x_i^l - x_j^l\right|\right)}, \frac{\max_l\left(\left|x_{new}^l - x_j^l\right|\right)}{\max_l\left(\left|x_i^l - x_j^l\right|\right)}\right) \tag{4}$$

We adopt $d_{i,j}$ to denote the Chebyshev distance, x_{new}^l is represented by Eqs (2) and (3). The equation is simplified as follows:

$$\begin{aligned}\left(d_i^{new}, d_j^{new}\right) &= \left(\frac{\max_l\left(\left|x_{new}^l - x_i^l\right|\right)}{d_{i,j}}, \frac{\max_l\left(\left|x_{new}^l - x_j^l\right|\right)}{d_{i,j}}\right.\\ &= \frac{1}{d_{i,j}} \times \left(\max_l\left(\left|x_i^l \times gap + (1 - gap) \times x_j^l - x_i^l\right|\right),\right.\\ &\quad \left.\max_l\left(\left|x_i^l \times gap + (1 - gap) \times x_j^l - x_j^l\right|\right)\right)\\ &= \frac{1}{d_{i,j}} \times \left(\max_l\left(\left|(1 - gap) \times (x_j^l - x_i^l)\right|\right),\right.\\ &\quad \left.\max_l\left(\left|gap \times (x_i^l - x_j^l)\right|\right)\right)\\ &= \frac{1}{d_{i,j}} \times \left(\max_l\left(\left|(1 - gap) \times d_{i,j}\right|\right), \max_l\left(\left|gap \times d_{i,j}\right|\right)\right)\\ &= (1 - gap, gap)\end{aligned} \tag{5}$$

Finally, the label of x_{new} can be represented as $lab(x_{new}) = (d_j^{new}, d_i^{new}) = (gap, gap - 1)$.

3.2.1. CDBC to solve the binary classification tasks

When CDBC is applied to the classification in the binary classification scenario, its main steps are shown in Algorithm 1.

Algorithm 1

Input: training dataset $D_{train} = D_{major} \cup D_{minor}$, k, $sample\ times$
Output: D_{cdbc}
1. **For** x_i in D_{major} **do**
2. Calculate the *k*-nearest neighbors of x_i in D_{train}
3. **End for**
4. **For** x_i in D_{minor} **do**
5. **For** x_j which is the neighbor of x_i **do**
6. **If** $x_j \in D_{major}$ **then**
7. **While** $sample\ times > 0$ **do**
8. $x_{new} = gap \times x_i + (1 - gap) \times x_j$
9. $lab(x_{new}) = (gap, gap - 1)$
10. $(x_{new}, lab(x_{new})) \rightarrow D_{cdbc}$
11. $sample\ times = sample\ times - 1$
12. **End while**
13. **End if**
14. **End for**
15. **End for**
16. $D_{cdbc} = D_{train} \cup D_{cdbc}$
17. **Return** D_{cdbc}

As shown in Algorithm 1, the input dataset is D_{train} (the original dataset is divided into training dataset and testing dataset according to 7 : 3). The training set D_{train} includes the majority class D_{major} and the minority class D_{minor}. k and $times$ represent the number of selected nearest neighbors and the number of times to generate gap data.

First, we determine the *k*-nearest neighbors of x_i in D_{major}, where x_i belongs to D_{minor}. Chebyshev distance is adopted to find the *k*-nearest neighbors, and it is shown in Eq (2). Then k neighbors are traversed to determine the types of x_j.

Then, based on Eqs (3) and (5), generate gap data, labels, and a new dataset D_{cdbc}. Repeat the steps until the end of condition is reached.

3.2.2. CDBC to solve the Multi-classification tasks

The idea of CDBC for multi-classification is the same as of binary classification. The advantage is that multi-classification is more scalable and conforms to darknet detection. In this section, we mainly introduce CDBC in multi-classification tasks.

Algorithm 2

Input: training dataset $D_{train} = D_{majors} \cup D_{minors}$, k, $sample\ times$
// $D_{majors} = \{ D_{major\_1}, D_{major\_2}, \ldots, D_{major\_m} \}$,
// $D_{minors} = \{ D_{minor\_1},, D_{minor\_2}, \ldots, D_{minor\_n} \}$,

Output: D_{cdbc}

For x_i in D_{minors} **do**
  Calculate the *k*-nearest neighbors of x_i in D_{train}
End for
For $D_{minor} \in D_{minors}$ **do**
  For $x_i \in D_{minor}$ **do**
    For x_j which is the neighbor of x_i **do**
      $D_{other\_types} = D_{majors} \cup (D_{minors} - D_{minor})$
      If $x_j \in D_{other\_types}$ **then**
        While $sample\ times > 0$ **do**
          $x_{new} = gap \times x_i + (1 - gap) \times x_j$
          $lab(x_{new}) = (gap, gap - 1)$
          $(x_{new}, lab(x_{new})) \rightarrow D_{cdbc}$
          $sample\ times = sample\ times - 1$
        End while
      End if
    End for
  End for
End for
$D_{cdbc} = D_{train} \cup D_{cdbc}$
Return D_{cdbc}

As can be seen from Algorithm 2, it is different from Algorithm 1. Firstly, there can be multiple majority and minority classes in the Input, and the division of the majority class and the minority class can be customized. Second, it is worth noting that when the sample and its k-nearest neighbors generate gap data, the label of the gap data is determined by the label of the neighbors and label of the samples.

4. Experimental results and analysis

In this section, the experimental environment, datasets, evaluation metrics are introduced and experiments are conducted to verify the effectiveness of the proposed method for detection.

4.1. Experimental environment

In the process of research, our experimental environment is set as follows: Operating System:Ubuntu 18.04, Processor: Intel i9-10920X CPU@3.50GHZ, Memory: 16GB, GPU: GeForce RTX 1080 Ti,nd Software Environment: conda 4.11.0,Python 3.7.5, sklearn 0.24.2, etc..

4.2. Datasets

The experiments are tested on two datasets, which are ISCXTor 2016 dataset [27] and CIC-Darknet 2020 dataset [17].

4.2.1 ISCXTor 2016 dataset ($D_{ISCXTor\text{-}A}$ and $D_{ISCXTor\text{-}B}$).

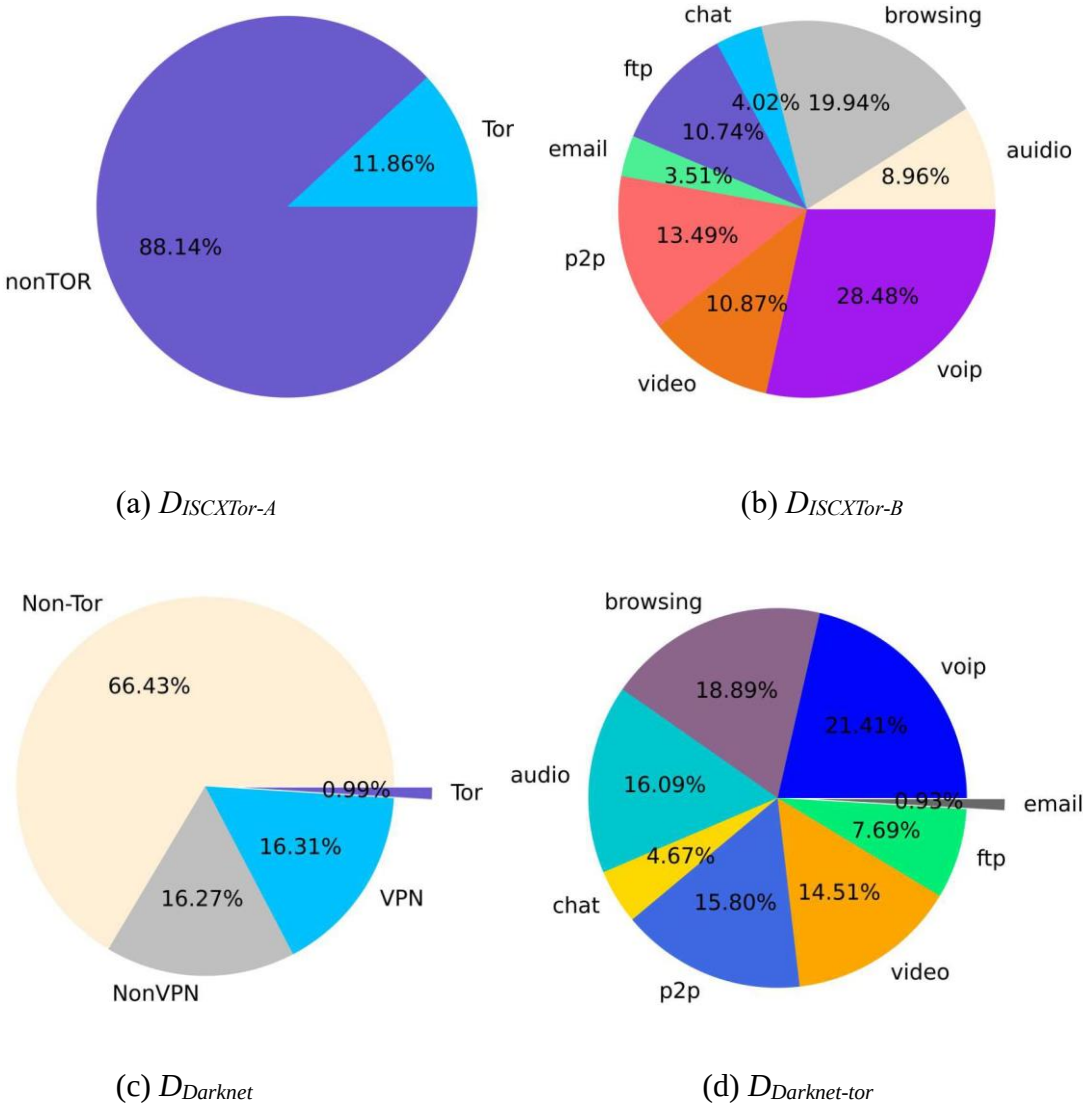
The ISCXTor 2016 dataset is a real traffic dataset recorded by the University of New Brunswick. This dataset includes two scenarios. Scenario A includes Tor traffic and non-Tor traffic. Scenario B includes 8 types Tor traffic. The details of the datasets are shown in Figure 3(a) $D_{ISCXTor\text{-}A}$ and (b) $D_{ISCXTor\text{-}B}$, and the types of Scenario B are shown in Table 1.

Table 1. The types included in $D_{Darknet\text{-}tor}$ and $D_{ISCXTor\text{-}B}$.

Types	**Source**
Browsing	Firefox,Chrome
Email	SMPTS, POP3S, IMAPS
Chat	ICQ, AIM, Skype, Facebook, Hangouts
File Transfer	Skype, FTP over SSH/SSL
P2P	uTorrent, Transmission
Audio	Spotify
VoIP	Facebook, Skype, Hangouts
Video	Vimeo,Youtube

Table 2. The number of the types in the datasets.

Types	**Dataset details**			
	$D_{ISCXTor\text{-}A}$	$D_{ISCXTor\text{-}B}$	$D_{Darknet}$	$D_{Darknet\text{-}tor}$
total	67834	8044	141530	1392
tor	8044	\	1392	\
non-tor	59790	\	93356	\
vpn	\	\	22919	\
non-vpn	\	\	23863	\
video	\	874	\	202
voip	\	2291	\	298
audio	\	721	\	224
browsing	\	1604	\	263
chat	\	323	\	65
file-transfer	\	864	\	107
mail	\	282	\	13



(a) $D_{ISCXTor\text{-}A}$ (b) $D_{ISCXTor\text{-}B}$

(c) $D_{Darknet}$ (d) $D_{Darknet\text{-}tor}$

Figure 3. $D_{ISCXTor}$ and $D_{Darknet}$ datasets distribution.

4.2.2 CIC-Darknet 2020 dataset ($D_{Darknet}$ and $D_{Darknet\text{-}tor}$)

The CIC-Darknet 2020 dataset is a public dataset of darknet traffic provided by the Canadian Institute for Cybersecurity. There are two layers in the dataset, the first layer ($D_{Darknet}$) contains four types: Tor, Non-Tor, VPN and NonVPN, and the second layer ($D_{Darknet\text{-}tor}$) contains 8 types which are shown in Table 1.

The $D_{Darknet}$ dataset contains more than 140,000 records, whose distribution is shown in Figure 3(c) $D_{Darknet}$ and (d) $D_{Darknet\text{-}tor}$. Tor traffic accounts for less than 1%, which is extremely unbalanced. The specific and detailed numbers in the datasets are shown in Table 2.

4.3. Evaluation metrics

The experiments include binary classification and multi-classification tasks. The binary classification distinguishes darknet traffic from non-darknet traffic. The multi-classification task is to classify the traffic more finely, to facilitate the processing and analysis of traffic types. In binary

classification, accuracy (*ACC*), precision, recall, false positive rate (*FPR*) and F1-score (F_1) are adopted to evaluate the detection. In multi-classification, macro-average is adopted. The calculations are shown as follow:

ACC indicates the proportion of correct predictions in all samples and is calculated as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

Precision indicates the proportion of samples for which the prediction is "1" that are indeed "1". The calculation is as follows:

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

Recall indicates the percentage of samples that are actually labelled as "1", which are actually identified. The calculation is as follows:

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

FPR represents the proportion of positive samples that are wrongly predicted to the total positive samples, which is calculated as follows:

$$FPR = \frac{FP}{TN + FP} \tag{9}$$

F_1 is a composite indicator and the core idea is to close the gap while increasing Precision and Recall as much as possible. The calculation is as follows:

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{10}$$

Macro Precision is an evaluation parameter for multi-classification problems and calculates the average value of Precision. The calculation is as follows:

$$macro\ Precision = \frac{1}{n}\sum_{i=1}^{n} Precision_i \tag{11}$$

Macro Recall is similar to *Macro Precision*. It is used to evaluate multi-classification problems. The formula for calculating the mean of the *Recalls* is as follows:

$$macro\ Recall = \frac{1}{n}\sum_{i=1}^{n} Recall_i \tag{12}$$

On the same principle, *Macro* F_1 is also used as a composite indicator for evaluating multi-classification problems:

$$macro\ F_1 = \frac{2 \times macro\ Precision \times macro\ Recall}{macro\ Precision + macro\ Recall} \tag{13}$$

where true positive (TP) is the number of correctly identified darknet traffic, true negative (TN) represents the number of normal traffic that is correctly identified, false positive (FP) represents the number of normal traffic that is incorrectly identified as darknet traffic, false negative (FN) indicates that darknet traffic is incorrectly identified as normal traffic.

Table 3. The results of the binary classification.

Classifier	Dataset	CDBC (%)			without CDBC (%)		
		ACC	**FPR**	**F_1**	**ACC**	**FPR**	**F_1**
RF	$D_{ISCXTor\text{-}A}$	**99.99**	**0.02**	**99.94**	99.93	0.05	99.71
	$D_{Darknet}$	**99.88**	**0.06**	**99.65**	99.82	0.10	99.46
XGBoost	$D_{ISCXTor\text{-}A}$	**99.98**	**0.02**	**99.97**	99.97	**0.02**	99.96
	$D_{Darknet}$	**99.95**	**0.02**	**99.79**	99.92	0.03	99.77
GBDT	$D_{ISCXTor\text{-}A}$	**99.95**	**0.03**	**99.79**	99.94	**0.03**	99.77
	$D_{Darknet}$	**99.75**	**0.06**	**99.58**	99.69	0.11	99.46
Bagging	$D_{ISCXTor\text{-}A}$	**99.99**	**0.02**	**99.94**	99.93	0.06	99.69
	$D_{Darknet}$	99.81	0.11	99.68	**99.87**	**0.07**	**99.77**
AdaBoost	$D_{ISCXTor\text{-}A}$	**99.99**	**0.01**	**99.99**	99.97	0.02	99.96
	$D_{Darknet}$	**99.92**	**0.03**	**99.77**	**99.92**	**0.03**	**99.77**
LR	$D_{ISCXTor\text{-}A}$	**95.09**	**1.48**	**76.98**	95.05	1.54	76.87
	$D_{Darknet}$	90.09	6.62	85.03	**91.42**	**6.29**	**85.51**
SVM	$D_{ISCXTor\text{-}A}$	**99.03**	**0.08**	**94.99**	98.84	0.09	94.87
	$D_{Darknet}$	**92.39**	**1.78**	**85.13**	91.25	2.49	82.71
NB	$D_{ISCXTor\text{-}A}$	64.68	39.79	57.32	**66.38**	**37.87**	**58.66**
	$D_{Darknet}$	57.88	45.13	52.28	**58.23**	**44.89**	**53.10**
DT	$D_{ISCXTor\text{-}A}$	**99.99**	**0.01**	**99.99**	**99.99**	**0.01**	99.96
	$D_{Darknet}$	**99.94**	**0.03**	**99.79**	99.92	**0.03**	99.77
KNN	$D_{ISCXTor\text{-}A}$	**99.20**	**0.40**	**96.33**	99.13	0.42	96.30
	$D_{Darknet}$	**97.39**	**1.23**	**92.23**	93.55	1.73	87.61
K-Means	$D_{ISCXTor\text{-}A}$	84.76	4.79	**50.54**	**88.18**	**0.01**	46.86
	$D_{Darknet}$	**58.23**	**44.89**	**37.60**	44.17	55.11	13.60

4.4. Experiment and analysis

Two groups of environments are set up in this experiment. The first group adopts CDBC, the second group does not adopt CDBC (without CDBC), and 11 methods are tested respectively, we set $k = 1$ and collect 1 *time* in total.

4.4.1 Test for binary-classification task

To explore the effect of CDBC on the darknet detection, a comparative experiment is conducted on the $D_{ISCXTor\text{-}A}$ and $D_{Darknet}$. Darknet traffic detection can be regarded as a binary classification task. The comparison results are shown in Table 3.

As can be seen from Table 3, the detection performance of the classifiers is better with CDBC. On the $D_{ISCXTor\text{-}A}$, the results of 10 methods (except NB) are improved. Especially with the ensemble methods, the accuracy is even close to 100%. On the $D_{Darknet}$, most of the metrics are improved in the CDBC environment. The experimental results show that in the binary classification task, the detection performance is better than that without CDBC.

4.4.2 Test for multi-classification task

In this section, the experiments are tested on multi-classification tasks, and the environment settings are as in the previous section. Considering the binary classification results, the ensemble learning methods are better than the single classifiers, so only 5 ensemble methods are selected in the multi-classification task. The comparison results are shown in Table 4.

In multi-classification tasks, the performance of all CDBC based methods is improved on $D_{ISCXTor\text{-}B}$ and $D_{Darknet\text{-}tor}$. Generally, CDBC can effectively form the "boundary" between samples and heterogeneous small samples, which helps improve the classification ability of the classifiers.

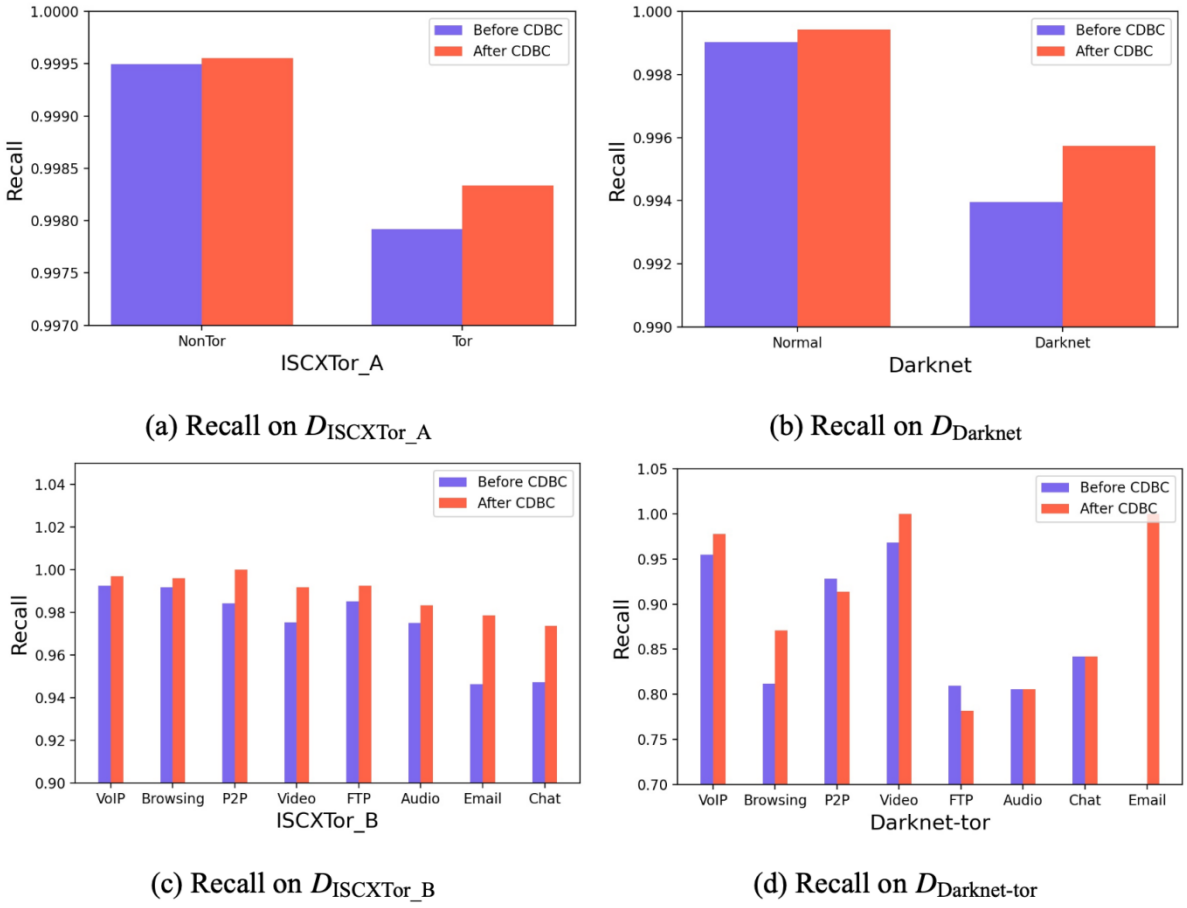
Table 4. The results of the multi-classification.

Classifier	Dataset	CDBC (%)			without CDBC (%)		
		ACC	**macroP**	**macroF$_1$**	**ACC**	**macroP**	**macroF$_1$**
RF	$D_{ISCXTor\text{-}B}$	**99.39**	**99.32**	**99.11**	98.38	97.84	97.65
	$D_{Darknet\text{-}tor}$	**89.85**	**90.70**	**89.84**	87.80	87.85	80.04
XGBoost	$D_{ISCXTor\text{-}B}$	**99.13**	**99.15**	**98.82**	**99.13**	98.86	98.72
	$D_{Darknet\text{-}tor}$	**92.34**	**92.56**	**90.92**	89.95	90.55	86.88
GBDT	$D_{ISCXTor\text{-}B}$	**99.30**	**99.29**	**99.15**	99.21	98.92	98.95
	$D_{Darknet\text{-}tor}$	**91.39**	**92.54**	**90.42**	90.19	91.59	87.88
Bagging	$D_{ISCXTor\text{-}B}$	**99.09**	**99.12**	**98.66**	98.30	97.65	97.50
	$D_{Darknet\text{-}tor}$	**88.28**	**90.76**	**87.80**	82.50	70.35	69.83
AdaBoost	$D_{ISCXTor\text{-}B}$	**99.34**	**98.99**	**98.87**	99.34	98.98	98.86
	$D_{Darknet\text{-}tor}$	**92.34**	**92.51**	**91.01**	89.95	90.55	86.88

4.4.3. The impact of CDBC on Recall

Taking the RF as an example, when the k and times are reasonably selected, Figure 4 shows the Recall on the four datasets.

As can be seen from Figure 4, after data enhancement of small samples by CDBC, the Recall of small samples is significantly improved. On $D_{ISCXTor\text{-}A}$, $D_{ISCXTor\text{-}B}$ and $D_{Darknet}$, Recall is higher than that without CDBC. Because the distribution boundary between darknet and non-darknet traffic is strengthened after using CDBC, the Recall for the categories is improved. On the $D_{Darknet\text{-}tor}$, the Recall of Email improves from 0.2 to 1.0, but the Recall of P2P and FTP also decreases slightly. Based on the above results, CDBC is helpful for the Recall of small samples, and CDBC can effectively assist in improving the detection.



(a) Recall on $D_{\text{ISCXTor\_A}}$ (b) Recall on D_{Darknet}

(c) Recall on $D_{\text{ISCXTor\_B}}$ (d) Recall on $D_{\text{Darknet-tor}}$

Figure 4. The impact of CDBC on Recall.

4.4.4. Comparing CDBC and other sampling methods

In this section, CDBC is compared with SMOTE_D 28 and Gaussian_SMOTE 29. The results are shown in Table 5.

As can be seen from Table 5, CDBC performs better on the $D_{ISCXTor\_B}$ and $D_{Darknet\text{-}tor}$, when comparing SMOTE_D and Gaussian_SMOTE. Although the accuracy of Bagging is not high enough, other classifiers perform better in the CDBC environment. Because the distribution of the two datasets is unbalanced, the gap data generated by CDBC can enhance the classification boundary, which can improve the classification ability of classifiers.

Table 5. Compared with other sampling methods.

Classifier	Dataset	CDBC(%)		GS_SOMTE(%)		SMOTE_D(%)	
		ACC	Recall	ACC	Recall	ACC	Recall
RF	$D_{ISCXTor\text{-}B}$	**99.34**	**98.90**	98.88	98.41	99.21	98.64
	$D_{Darknet\text{-}tor}$	**89.85**	**89.00**	87.55	83.42	88.75	84.71
XGBoost	$D_{ISCXTor\text{-}B}$	**99.34**	**98.49**	98.67	98.29	99.21	98.12
	$D_{Darknet\text{-}tor}$	**92.34**	**89.34**	90.91	86.29	91.15	88.59
GBDT	$D_{ISCXTor\text{-}B}$	**99.42**	**99.01**	96.77	93.83	98.34	97.79
	$D_{Darknet\text{-}tor}$	**91.39**	**88.39**	89.23	84.62	90.67	88.24
Bagging	$D_{ISCXTor\text{-}B}$	98.30	98.20	98.80	98.09	**99.21**	98.04
	$D_{Darknet\text{-}tor}$	88.28	85.03	87.08	77.85	**88.51**	**86.37**
AdaBoost	$D_{ISCXTor\text{-}B}$	**99.33**	98.75	98.67	98.29	99.30	**98.93**
	$D_{Darknet\text{-}tor}$	**92.34**	**89.56**	90.91	86.29	91.15	88.59

4.4.5. The effects of hyperparameters *k* and *times* on CDBC

This section conducts experiments on the hyperparameter settings of CDBC. The hyperparameters include the *k* nearest neighbors of the minority sample, and the number of *times* to generate gap data samples.

The experiment is carried out on the $D_{ISCXTor\_B}$, and the value of *k* ranges from 5 to 100 with an interval of 5, and $times = 2$. The results are shown in Figure 5.

As shown in Figure 5, when the value of *k* increases, the Accuracy, F_1, etc. become lower. Only AdaBoost and XGBoost have little effect on the value of *k*. However, the effect of other methods oscillates with the increase of *k*, and the effect generally declines. Especially when GBDT is used as a classifier, the increase of *k* has obvious influence on detection. We analyze the reason for this because *k* represents the number of neighbors. After the neighbors increase, the samples that are not on the edge are regarded as neighbors. The generated gap data cannot enhance the spatial distribution edges well.

We set *times* to range from 1 to 5 with an interval of 1. Figure 6 shows the impact of *times* on detection.

As shown in Figure 6, the ordinate represents the prediction results of different classifiers with the increase of *times*. When $times = 1$, the value of the ordinate represents the average value of *k* from 5 to 100 (with an interval of 5). It can be seen that when the *times* increases, the test results of various classifiers decrease, among which the GBDT is the most obvious. Analyzing the reason, as the *times* increase, the number of gap data increases. When there is a lot of gap data, the classifiers overfit. Therefore, generating too much gap data cannot enforce the boundary and even leads to lower detection accuracy. *k* and *times* need to be set appropriately. The principle is to generate a small amount of gap data, which can achieve good results and reduce training overhead.

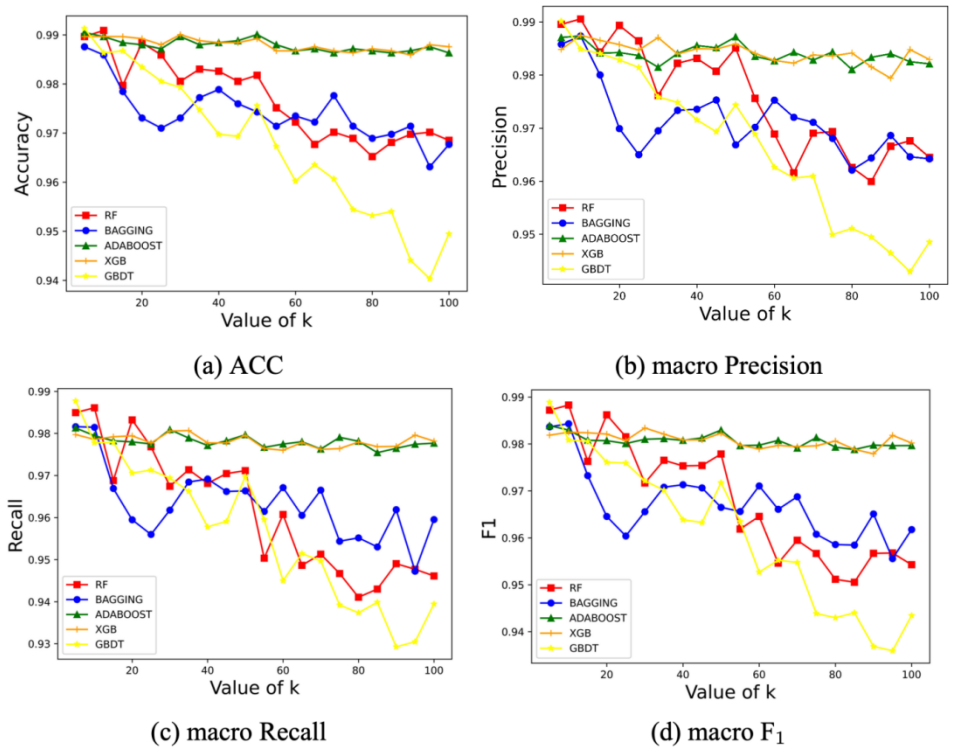


Figure 5. The impact of *k* on each evaluation index.

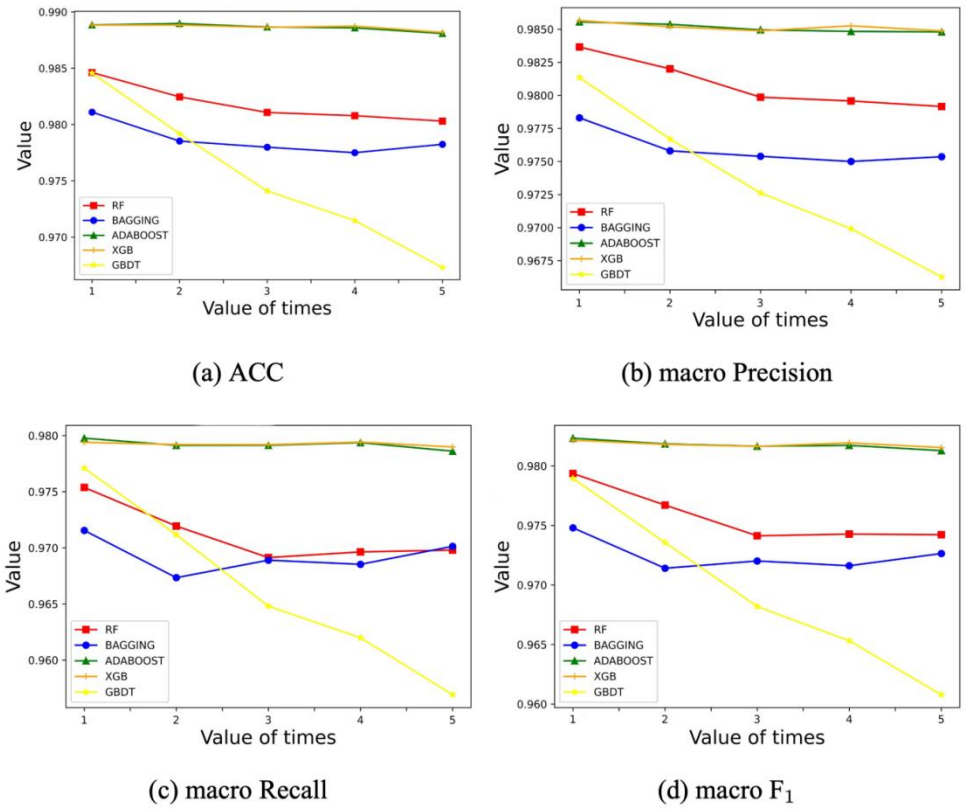


Figure 6. The impact of times on each evaluation index.

5. Conclusions

This paper first proposes a Chebyshev distance based Between-class learning algorithm, called CDBC. The method generates "gap data" by calculating the distances between heterogeneous traffic. Gap data can enhance the boundary between small and other samples and optimize the classification performance of the classifier. Second, the detection architecture of darknet based on CDBC is introduced, and we discuss data preprocessing, training and darknet detection. Thirdly, CDBC is used on two datasets, and the experiments test 11 kinds of classifiers in CDBC and without CDBC environments. The experimental results show that when CDBC is applied to the detection, the accuracy of the classifiers can be improved and the best result is 99.99%. The CDBC based Adaboost method is the best. In addition, CDBC is also used to compare with existing sampling methods, and the results show that CDBC is better than others. We also analyze the hyperparameters and conclude that the detection accuracy of the classifiers is significantly improved when the gap data is sampled in a small amount. The proposed method can overcome the difficulties caused by the small number of samples, and can solve the problem of low detection accuracy. We provide a solution for cyberspace security researchers. Moreover, the sampling method (CDBC) can also be extended to the other fields.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The work of this research article was awarded at the International Symposium on Intelligent Robots and Systems (ISoIRS 2022). So it got the opportunity to be recommended. We are very grateful to ISoIRS 2022 for the recognition and recommendation.

Conflict of interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

References

1. A. Montieri, D. Ciuonzo, G. Aceto, A. Pescapé, Anonymity services tor, i2p, jondonym: classifying in the dark (web), *IEEE Trans. Dependable Secure Comput.*, **17** (2018), 662−675. https://doi.org/10.1109/TDSC.2018.2804394
2. Y. Gao, J. Lin, J. Xie, Z. Ning, A real-time defect detection method for digital signal processing of industrial inspection applications, *IEEE Trans. Ind. Inf.*, **17** (2021), 3450−3459. https://doi.org/10.1109/TII.2020.3013277

3. W. Wang, N. Kumar, J. Chen, Z. Gong, X. Kong, W. Wei, et al., Realizing the potential of the internet of things for smart tourism with 5G and AI, *IEEE Network*, **34** (2020), 295−301. https://doi.org/10.1109/MNET.011.2000250
4. R. Dingledine, N. Mathewson, P. Syverson, Tor: The second-generation onion router, in *13th USENIX Security Symposium*, **2004** (2004), 303−320. https://doi.org/10.1016/0016-0032(45)90142-6
5. A. Cuzzocrea, F. Martinelli, F. Mercaldo, G. Vercelli, Tor traffic analysis and detection via machine learning techniques, in *2017 IEEE International Conference on Big Data*, **2017** (2017), 4474−4480. https://doi.org/10.1109/BigData.2017.8258487
6. R. Jansen, M. Juarez, R. Galvez, T. Elahi, C. Diaz, Inside job: Applying traffic analysis to measure tor from within, *Network Distributed Syst. Security*, **2018** (2018). http://dx.doi.org/10.14722/ndss.2018.23261
7. H. Yin, Y. He, I2P anonymous traffic detection and identification, in *2019 5th International Conference on Advanced Computing & Communication Systems*, **2019** (2019), 157−162. https://doi.org/10.1109/ICACCS.2019.8728517
8. I. Clarke, O. Sandberg, B. Wiley, Freenet: A distributed anonymous information storage and retrieval system, *Des. Privacy Enhancing Technol.*, **2001** (2001), 46−66. https://doi.org/10.1007/3-540-44702-4_4
9. S. Lee, S. H. Shin, B. H. Roh, Classification of freenet traffic flow based on machine learning, *J. Commun.*, **13** (2018), 654−660. https://doi.org /10.12720/jcm.13.11.654-660
10. S. Wang, Y. Gao, J. Shi, X. Wang, C. Zhao, Z. Yin, Look deep into the new deep network: A measurement study on the ZeroNet, in *Computational Science-ICCS 2020*, (2020), 595−608. https://doi.org/10.1007/978-3-030-50371-0_44
11. M. Wang, X. Wang, J. Shi, Q. Tan, Y. Gao, M. Chen, et al., Who are in the darknet measurement and analysis of darknet person attributes, in *2018 IEEE Third International Conference on Data Science in Cyberspace*, **2018** (2018), 948−955. https://doi.org/10.1109/DSC.2018.00151
12. C. Fachkha, M. Debbabi, Darknet as a source of cyber intelligence: Survey, taxonomy, and characterization, *IEEE Commun. Surv. Tutorials*, **18** (2015), 1197−1227. https://doi.org/10.1109/COMST.2015.2497690
13. G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, A. A. Ghorbani, Characterization of encrypted and VPN traffic using time-related features, in *Proceedings of the 2nd International Conference on Information Systems Security and Privacy*, **1** (2016), 407−414. https://doi.org/10.5220/0005740704070414
14. Y. Hu, F. Zou, L. Li, P. Yi, Traffic classification of user behaviors in tor, i2p, zeronet, freenet, in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications*, (2020), 418–424. https://doi.org/10.1109/TrustCom50675.2020.00064
15. R. Rawat, V. Mahor, S. Chirgaiya, R. N. Shaw, A. Ghosh, Analysis of darknet traffic for criminal activities detection using TF-IDF and light gradient boosted machine learning algorithm, *Innovations Electr. Electron. Eng.*, **2021** (2021), 671−681. https://doi.org/10.1007/978-981-16-0749-3_53
16. Q. A. Al-Haija, M. Krichen, W. A. Elhaija, Machine-learning-based darknet traffic detection system for IoT applications, *Electronics*, **11** (2022), 556. https://doi.org/10.3390/electronics11040556

17. A. H. Lashkari, G. Kaur, A. Rahali, DIDarknet: A contemporary approach to detect and characterize the darknet traffic using deep image learning, in *2020 the 10th International Conference on Communication and Network Security*, (2020), 1−13. https://doi.org/10.1145/3442520.3442521
18. C. Liu, L. He, G. Xiong, Z. Cao, Z. Li, FS-Net: A flow sequence network for encrypted traffic classification, in *IEEE INFOCOM 2019-IEEE Conference On Computer Communications*, (2019), 1171−1179. https://doi.org/10.1109/INFOCOM.2019.8737507
19. M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade, M. Saberian, Deep packet: A novel approach for encrypted traffic classification using deep learning, *Soft Comput.*, **24** (2020), 1999−2012. https://doi.org/10.1007/s00500-019-04030-2
20. X. Wang, S. Chen, J. Su, App-Net: A hybrid neural network for encrypted mobile traffic classification, in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops*, (2020), 424−429. https://doi.org/10.1109/INFOCOMWKSHPS50562.2020.9162891
21. M. B. Sarwar, M. K. Hanif, R. Talib, M. Younas, M. U. Sarwar, DarkDetect: Darknet traffic detection and categorization using modified convolution-long short-term memory, *IEEE Access*, **9** (2021), 113705−113713. https://doi.org/10.1109/ACCESS.2021.3105000
22. W. Cai, L. Xie, W. Yang, Y. Li, Y. Gao, T. Wang, DFTNet: Dual-path feature transfer network for weakly supervised medical image segmentation, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **2022** (2022), 1−12. https://doi.org/10.1109/TCBB.2022.3198284
23. X. Xie, Y. Li, Y. Gao, C. Wu, P. Gao, B. Song, et al., Weakly supervised object localization with soft guidance and channel erasing for auto labelling in autonomous driving systems, *ISA Trans.*, **132** (2023), 39−51. https://doi.org/10.1016/j.isatra.2022.08.003
24. W. Wang, J. Chen, J. Wang, J. Chen, J. Liu, Z. Gong, Trust-enhanced collaborative filtering for personalized point of interests recommendation, *IEEE Trans. Industrial Inf.*, **16** (2020), 6124−6132. https://doi.org/10.1109/TII.2019.2958696
25. Y. Tokozume, Y. Ushiku, T. Harada, Between-class learning for image classification, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, **2018** (2018), 5486−5494, arXiv.1711.10284
26. Y. Gao, J. Chen, H. Miao, B. Song, Y. Lu, W. Pan, Self-learning spatial distribution-based intrusion detection for industrial cyber-physical systems, *IEEE Trans. Comput. Social Syst.*, **9** (2022), 1693−1702. https://doi.org/10.1109/TCSS.2021.3135586
27. A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, A. A. Ghorbani, Characterization of tor traffic using time based features, in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy*, **2017** (2017), 253−262. https://doi.org/10.5220/0006105602530262
28. F. R. Torres, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, SMOTE-D a deterministic version of SMOTE, in *Mexican Conference on Pattern Recognition*, **9703** (2016), 177−188. https://doi.org/10.1007/978-3-319-39393-3_18
29. H. Lee, J. Kim, S. Kim, Gaussian-based SMOTE algorithm for solving skewed class distributions, *Int. J. Fuzzy Logic Intell. Syst.*, **17** (2017), 229−234. https://doi.org/10.5391/IJFIS.2017.17.4.229

AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)