Hindawi
Security and Communication Networks
Volume 2022, Article ID 3104392, 14 pages
https://doi.org/10.1155/2022/3104392

Research Article

Network Traffic Obfuscation against Traffic Classification

Likun Liu,[1] **Haining Yu**,[1] **Shilin Yu**,[2] **and Xiangzhan Yu**[1]

[1]*School of Cyber Science and Technology, Harbin Institute of Technology, Harbin 150001, China*
[2]*Aerospace Science and Industry Academy of Intelligent Operation and Information Security (Wuhan) Co., Ltd, Wuhan 430040, China*

Correspondence should be addressed to Haining Yu; yuhaining@hit.edu.cn

Received 4 November 2021; Revised 12 May 2022; Accepted 9 August 2022; Published 31 August 2022

Academic Editor: Yanhui Guo

In recent years, tremendous progress has been made in network traffic classification and its use has become ubiquitous in many emerging applications, such as Internet censorship in many countries and ISP traffic engineering. However, the traffic analysis of intermediaries brings the risk of privacy disclosure to users. This paper presents a network traffic obfuscation technology to resist traffic classification. It deceives the machine learning and deep learning models by generating adversarial samples. The adversarial samples generation algorithm includes a white-box attack algorithm based on fuzzy strategy and a black-box attack algorithm based on smote data enhancement. Experiments based on the ISCXTor2016 public data set show that the MIM algorithm has the best performance in white-box attacks, and the obfuscation success rate of DNN and LSTM models is 90%. In the black-box attack, the obfuscation effect of LSTM is the best, while CNN has stronger robustness.

1. Introduction

Advanced traffic analysis technology has brought great challenges to information concealment and privacy protection. Common deep packet inspection (DPI) determines traffic categories and user behaviors by monitoring and analyzing incoming and outgoing data packets from intermediate nodes. Powerful deep learning technology enables network intermediaries to identify target information from huge network traffic. Traffic obfuscation is one of the common techniques to resist traffic analysis. Traditional traffic obfuscation technologies can only protect computers from the attack of traffic analysis to some extent. The machine learning classifier with low performance brings great challenges to traffic obfuscation. Adversarial machine learning technology opens a new door to defend against traffic analysis.

In order to improve the ability of hiding private information, traffic obfuscation technology aims to ensure that the targeted traffic cannot be recognized by the attacker in the observed traffic set by a series of operations such as randomization and mimicry shaping. For example, the Tor browser reencapsulates the anonymous network traffic through meek [1] before sending the message and disguises the anonymous network traffic as the traffic accessing Microsoft Azure or Amazon Web service, to realize traffic obfuscation. Due to the existence of network supervision requirements, the identification and tracking technology for encrypted or obfuscated traffic has also attracted much attention. Network intermediaries identify target traffic through traffic fingerprints. Traffic fingerprint is a feature or a series of feature combinations representing certain traffic, including static fingerprint features and dynamic fingerprint features. Even if traffic obfuscation technology hides some information of the original traffic, network censors, regulators, and network intermediaries can still study traffic identification technology based on small differences. The most common way of network traffic identification is through DPI [2]. DPI is a new and effective real-time packet detection technology, which is used to monitor and analyze incoming and outgoing packets from intermediate nodes. For example, China, Turkey, Iran [3], and other countries widely use DPI for network censorship. With the rapid development of artificial intelligence, more and more machine learning technologies are applied to traffic identification, which effectively improves the efficiency and accuracy of traffic identification. Compared with traditional machine learning methods, the deep learning method does not need to extract flow features manually and achieves better results. Due to the small scale and poor scalability of the data set, the effectiveness of this traffic identification technology still needs to be verified in a real large-scale environment.

While the identification of obfuscated traffic is continuously strengthened, new obfuscation technologies have also been introduced accordingly. These two technologies are offensive and defensive to each other, and each has its advantages and disadvantages in the process of development. The research on traffic obfuscation for secure link channels is of great significance in the field of information encryption and information hiding. Legally using traffic obfuscation is an important mean to protect Internet users' privacy and data security. It can effectively prevent a series of attacks against users' privacy, such as network eavesdropping, traffic analysis, and so on. Traffic obfuscation complicates the communication flow between users and provides security protection for the information content itself by employing information encryption, to improve the difficulty of man-in-the-middle traffic analysis attack.

The main contributions of this paper are as follows.

(i) We propose an improved white-box attack to generate an obfuscation strategy, which is a gradient iterative descent algorithm. In the process of updating the adversarial sample, it is not allowed to reduce the packet size; that is, relative to the packet size in the original sequence, the corresponding packet size in the adversarial sample sequence follows the monotonic nondecreasing rule.

(ii) We propose a black-box attack method by training the alternative model, which uses SMOTE technology for data enhancement. We discuss the transitivity of adversarial samples between different models and then evaluate the effectiveness of the black-box attack algorithm.

The paper is organized as follows: Section 2 presents the related work. Section 3 describes the white-box attack, followed by the black-box attack in Section 4. The experiment setup shows in Section 5. Finally, we summarize the research.

2. Related Work

Network traffic classification groups similar or related traffic data, which is one mainstream technique in the field of network management, security, and man-in-the-middle traffic analysis attacks. A cost-sensitive SVM (CMSVM) [4] is proposed to solve the imbalance problem in network traffic identification. To deal with the limitation of encrypted traffic classification in accuracy, Ren et al. [5] proposed a tree structural recurrent neural network (Tree-RNN), which divides a large classification into small classifications by using the tree structure. Dong et al. [6] applied sampled NetFlow data to traffic identification and proposed a Deep Belief Networks Application Identification (DBNAI) method to improve performance. Traffic classification methods are emerging, but there are few techniques against classification.

The purpose of traffic obfuscation is to hide the characteristics of traffic fingerprints and resist traffic analysis based on deep packet inspection or machine learning. The traditional traffic obfuscation includes randomization obfuscation, mimicry obfuscation, and tunnel obfuscation.

Randomization obfuscation mainly randomizes some visible metadata features and message load of the targeted traffic utilizing encryption and adding random noise, so that the opponent cannot identify the targeted traffic from the traffic set. Obfs1 is the protocol obfuscation layer of TCP. To hide the protocol type in use, it randomizes the message load by using stream cipher after key negotiation. It does not provide authentication or data integrity and does not randomize the data length. Obfs2 [7] is the first obfuscation protocol widely used in onion networks. However, due to the lack of a robust key exchange method, any opponent capable of monitoring the initial handshake of obfs2 can recover the key. To resist such attacks, obfs3 [8] uses a customized Diffie Hellman key exchange protocol to negotiate keys. Compared with obfs2 and obfs3, Brandon's dust protocol [9] has a more random payload. Except for Mac, IV, and randomly filled fields, other fields are encrypted. To complete key exchange without unencrypted handshake, dust protocol adopts out-of-band half handshake. Similar to Kopsell's model [10], peers must receive out-of-band invitations to join the network. Weinberg et al. introduced a proxy framework, StegoTorus [11], which can confuse the protocol identification on the application layer and the transport layer to improve the resilience of Tor to fingerprint attacks. Tan [12] analyzed DHT attacks and eclipse attacks against Tor. To improve Tor's ability to defend against active detection attacks, Winter [13] proposed ScrambleSuit polymorphic network protocol. The ScrambleSuit protocol can be easily integrated into Tor's existing ecosystem, and it is difficult for inspectors to identify ScrambleSuit using regular expressions. Obfs4 [14] tries to provide authentication and data integrity based on ScrambleSuit. In the handshake phase, the data is filled with random length to confuse the initial stream signature. After the handshake is completed, the application layer data is split into "packets" for transmission, and encryption and authentication are completed in NaCl secret box (Poly1305/XSalsa20) [15] "frames".

In the mimicry obfuscation, the mimicry client is responsible for shaping the traffic to make it imitate other protocols to some extent, and the mimicry server is responsible for recovering the traffic. To resist statistical traffic analysis, Wright et al. [16] shaped one type of traffic into another by using convex optimization technology. This traffic shaping method is more efficient than traditional packet filling but ignores the key element of a secure encryption channel. Wang et al. [17] proposed a novel anti-censorship network browsing framework CensorsSpoofer, which uses IP address spoofing to send data from the agent to the user and imitates the encrypted VoIP session to transmit downstream data. Dyer et al. [18] used a new cryptography primitive called format conversion encryption

(FTE) in the mimicry obfuscation technology, which extends the traditional symmetric encryption and can convert the ciphertext into a specific format. An FTE scheme includes three parts: key generation, encryption, and decryption. An additional recording layer is used to buffer, encode, parse, and decode the FTE message stream. Compared with the standard SSH tunnel, using FTE as the proxy system will not produce any delay overhead, and the bandwidth overhead is only 16%. Mohajeri et al. [19] proposed the SkypeMorph model, which uses Skype video call as the target protocol, making it difficult for opponents to distinguish confusing bridge communication from actual Skype call through statistical characteristics. Because Skype traffic accounts for a high proportion of the Internet and all communications are encrypted, it can provide an ideal encrypted channel for Tor traffic. In the initial setup phase, the SkypeMorph client and bridge use Skype API to log in to Skype and exchange public keys to start Skype video calls between both parties. There are various TCP connections in a video call, and some TCP connections remain in the same state after the end of the conversation. In the traffic shaping stage, the Oracle component controls the size and timing of each packet sent. The component provides a simple traffic shaping method and traffic morphing method, respectively.

Tunnel traffic obfuscation uses normal traffic samples as tunnels to transmit target traffic. Tunnel technology can also be regarded as mimicry technology. Infranet [20] first embedded the real communication into the web session, sent the request using a secret channels, and used the picture steganography to return the data. This way can easily be located to the agent by the examiner disguised as an ordinary user. Foe [21] is an anticensorship tool that uses e-mail as a tunnel. It is based on SMTP protocol and can run on most e-mail servers. In addition, the CensorSpoofer framework [17] also uses tunneling technology, but it is only used to send user requests (such as URLs). To hide the real network traffic of users, Brubaker et al. [22] designed a new set of censorship avoidance systems, called Cloudtransport, which uses cloud storage services such as Amazon S3 to establish tunnels. Cloudtransport uses the same “cloud client” library, protocol, and network server as any other application based on given cloud storage, so simple protocol identification is invalid for Cloudtransport. Cloudtransport only confuses the traffic before passing through the proxy cloud service. When the traffic reaches the bridge, it will no longer hide the user traffic. Meek [1] uses the “domain name prefix” technology to forward messages to the Tor relay bridge. Domain name prefix refers to the use of different domain names at different communication layers. The message sent by the Tor browser will be reencapsulated by a meek client to build a special HTTPS request and sent to the intermediate web service (e.g., CDN) configured with multiple domain names.

With the significant increase of network traffic scale and complexity, compared with the content-based DPI method, machine learning and data-driven traffic identification technology shows better performance. Especially after the emergence of deep learning, it no longer relies on manual extraction of traffic characteristics, which saves a lot of manpower and material resources and has strong scalability. The traffic recognition technology based on deep learning will be the focus and mainstream of future research, such as [23, 24]. The development of adversarial machine learning provides opportunities for traffic obfuscation technology. Especially in the face of traffic identification technology based on deep learning, adversarial machine learning can effectively protect the security and privacy of network traffic. A variety of attack algorithms (FGSM [25], BIM [26], MIM [27], and CW [28]) can generate obfuscation traffic samples under legal modification and limited resource constraints. These carefully designed samples can be used not only as training samples for poisoning attacks, but also as test samples for evasion attacks. In addition, Muhammad et.al [29] utilized the mutual information (MI) for crafting adversarial perturbations and substitute model training to perform the black-box adversarial attack. However, the lack of frequency of mutual information may lead to sample imbalance and abnormal characteristics.

This paper mainly focuses on how to restrict the identification of anonymous traffic by traffic obfuscation. By interfering with the input data, the recognition accuracy of the model is reduced. As shown in Figure 1, the network traffic classification model and traffic obfuscation technology are rivals. The former attacks the secure link channel system and the latter protects the secure link channel systems from eavesdropping, sniffing, traffic analysis, and other attacks, including white-box attacks and black-box attacks.

3. White-Box Attack

3.1. Architecture. The obfuscation traffic generation framework based on a white-box attack is shown in Figure 2, which is mainly divided into four steps: data set division, model training, constructing antitraffic samples, and finally verifying the effectiveness of the attack algorithm.

Step 1: divide the original data set into the training set and test set, in which the training set is used for model training, and the test set is used to construct adversarial traffic samples. Only one fixed random partition is performed. Different deep learning algorithms use the same training data set, and different antiattack algorithms use the same test data set.

Step 2: under the same training data set, DNN, CNN, and LSTM deep learning algorithms are used for experiments, respectively. For the trained neural network model, persistence processing is needed to provide direct access for white-box attackers and verify the success rate of the attack algorithm.

Step 3: construct adversarial traffic samples through a white-box attack algorithm, including FGSM, BIM, MIM, and CW. This step requires the attack algorithm to fully access the deep learning model and obtain the gradient of the model about the given input. For each attack algorithm, a confused traffic set is constructed, respectively. The size of the traffic set is similar to that of the test set, which is provided to the neural network model for prediction, and the attack success rate of the algorithm is estimated by the prediction success rate.

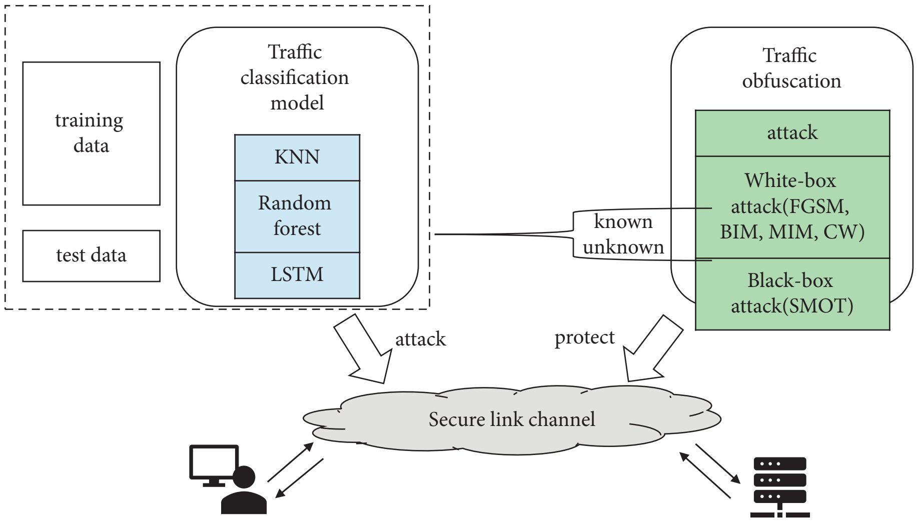


Figure 1: Network traffic obfuscation against traffic classification.

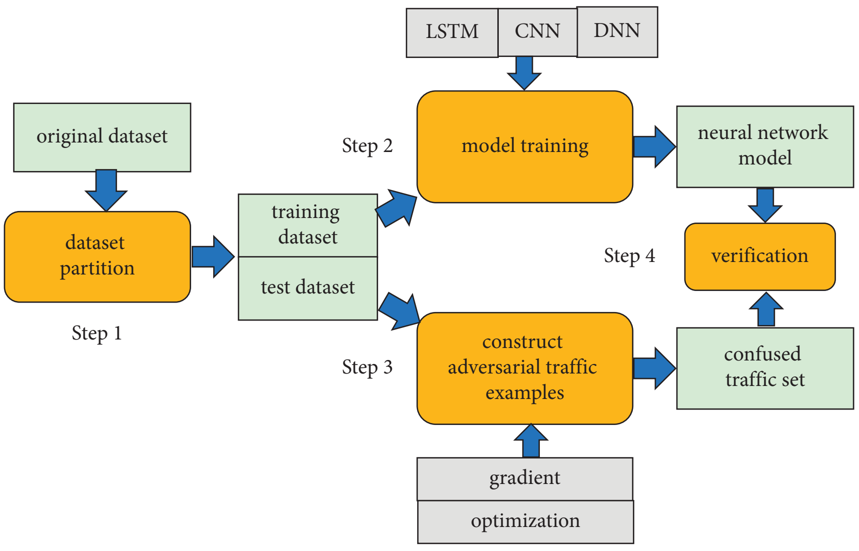


Figure 2: Generating obfuscated traffic based on white-box attack.

Step 4: evaluate the effectiveness of the algorithm from several dimensions, including the effectiveness of different algorithms for the same model and the reliability of the same model in the face of different algorithms.

3.2. White-Box Attack-Based Gradient. In the white-box attack, the attacker can fully access the model, parameters, and architecture and organize the adversarial attack by making noise through the adversarial attack algorithm. There are usually gradient methods and decision-making methods in white-box antiattack algorithms. The gradient-based attack is the most commonly used method in the literature. It uses the detailed information of the target model about the gradient of a given input to iteratively generate adversarial samples, which requires the attacker to fully understand and access the target model. Decision-based attack is a simpler and more flexible method, which only needs to query the Softmax layer of the target model and iteratively calculate the noise by using the rejected process. FGSM, BIM, and MIM are typical gradient-based adversarial attack algorithms.

3.2.1. Adversarial Samples Based on FGSM. When the attacker obtains the complete neural network model and the original test samples, he can cheat the model to output the wrong category by updating the original sample input in the gradient direction of the objective function. The objective function describes the error between the original output and the target output of the model.

The neural network model can be formally described as

$$\begin{aligned} F &= \text{softmax} \cdot F_n \cdot F_{n-1} \cdot \cdots \cdot F_1, \\ F_i(x) &= \sigma(\theta_i \cdot x) + \hat{\theta}_i. \end{aligned} \tag{1}$$

The output vector $\boldsymbol{y}$ of the neural network is probability, and the last label of the classifier is

$$C(x) = \arg\max_i F(x)_i. \tag{2}$$

FGSM (fast gradient sign method) is a one-step gradient method, which generates adversarial samples only through one update. Assuming that the attacker can fully access the neural network, including architecture and parameters, the way to generate adversarial samples using FGSM is

$$x' = x - \epsilon \cdot \operatorname{sign}(\nabla_x J(x, y)). \tag{3}$$

The function $\boldsymbol{J}$ is the loss function, which is used to measure the change degree of classification results. The commonly used loss function is the cross-entropy function. A sign is a symbolic function used to obtain the gradient direction.

The generalized formula of FGSM is

$$x' = x - \epsilon \cdot \frac{\nabla_x J(x, y)}{\nabla_x J(x, y)_2}. \tag{4}$$

For each negative sample in the test set, use (3) to update the negative sample input to maximize the deviation predicted by the deep learning model, that is, the probability that the model will finally classify the input as positive as possible. There is no requirement for the positive and negative direction of length change, only the maximum disturbance threshold is limited to 20, and the length interval is [0, 200].

3.2.2. Adversarial Samples Based on BIM. BIM (basic iterative method) method is an iterative version of FGSM, that is, iterative multiple fast gradient sign method. Although FGSM is fast, in many cases, a single update is not enough to fail the model prediction. Therefore, it is considered to update the samples iteratively to improve the attack success rate.

The iterative formula of BIM is

$$\begin{cases} x_0' = x \\ x_{k+1}' = x_k' - \alpha \cdot \operatorname{sign}(\nabla_x J(x_k', y)) \end{cases}. \tag{5}$$

3.2.3. Adversarial Samples Based on MIM. MIM (momentum iterative method) integrates the momentum term into the iterative process of attack, makes the update of direction more stable, gets rid of the bad local maximum, and produces more transitive adversarial samples. Similarly, in the first-order optimization algorithm, the momentum method adds the momentum term to the gradient descent method and accumulates the previous "momentum" to replace the real gradient, to accelerate in the correct gradient direction. MIM and momentum method have similar forms, and the specific parameters are not the same.

The iterative formula of MIM is

$$\begin{cases} v_{k+1} = \mu \cdot v_k + \dfrac{\nabla_x J(x_k', y)}{\nabla_x J(x_k', y)_1} \\ \\ x_{k+1}{}' = x_k' - \alpha \cdot \operatorname{sign}(v_{k+1}) \end{cases}. \tag{6}$$

Compared with BIM, MIM can achieve a higher attack success rate. Because BIM has been moving greedily along the given gradient direction in the iterative process, the adversarial sample is easy to fall into the local maximum and the sample overfitting. The momentum term introduced in MIM is helpful to cross some narrow valleys or humps, to skip the bad local minimum or maximum.

3.2.4. Adversarial Samples Based on CW. Different from FGSM, BIM, MIM, and other methods, CW not only requires model classification error but also wants to be as similar as possible between the adversarial sample and the original sample; that is, the false rate is lower. The method formalizes the original optimization problem of finding adversarial samples as follows:

$$\begin{gathered} \text{minimize}\, D(\mathrm{x}, x'), \\ \text{such that}\, C(x') = \mathrm{t}, x' \in [\mathrm{clip}_{\min}, \mathrm{clip}_{\max}], \end{gathered} \tag{7}$$

where function $D(x, x')$ is a distance measurement function, which is used to describe the similarity between the adversarial sample and the original sample. Because the classifier function $C(x')$ is highly nonlinear, the original problem is difficult to solve. An objective function f needs to be found so that $C(x') = t$, if and only if $f(x') = 0, C(x') = t$. Thus, the original optimization problem is transformed into a new optimization problem, which is formally described as follows:

$$\begin{gathered} \text{minimize}\, D(\mathrm{x}, x') + c \cdot f(x'), \\ \text{such that}\, x' \in [\mathrm{clip}_{\min}, \mathrm{clip}_{\max}]. \end{gathered} \tag{8}$$

Because L_0 distance metric is nondifferentiable, L_∞ distance measurement is not completely differentiable, so gradient descent method is not suitable to optimize parameters. Therefore, this chapter uses Euclidean distance to measure the distance between the adversarial sample and the original sample and uses the gradient descent method to solve the new optimization problem, in which the objective function f uses the loss function loss_2 defined by CarliniWagnerL2 in the CleverHans attack library.

3.2.5. Obfuscation Traffic Generation Strategy. The representation of the obfuscated flow *adv* is obtained by adding the original flow representation *ori* and the noise representation *del*, namely: *adv* = *ori* + *del*. When representing traffic with a sequence of top *n* packets' lengths, there are usually two ways to add noise into the original traffic, by appending extra bytes to packets of the intended traffic flow and inserting extra packets at specific intervals into the original data stream.

For example, for a certain original flow representation *ori* = [150, 300, 350, 280, 500, 350, 150, 250, 500, 400], the noise vector is [10, -20, 5, 10, -30, 30, 20, -30, -5, 50], and the target representation is [160, 280, 355, 270, 470, 380, 170, 220, 495, 450]. To add noise to the original traffic, firstly insert a packet with length of 280 between the 1st and 2nd packets. Then insert a packet with length of 270 between the

2nd and 3rd packets. Lastly insert four data packets with lengths of 170, 220, 495, and 450 after the 4th packet. Now the flow representation becomes [150, 280, 300, 270, 350, 280, 170, 220, 495, 450]. Then append extra bytes with the lengths of 10, 55, 120, 100 to the 1st, 3rd, 5th, and 6th packets to get the noise representation. There is a total amount of data with size of 2170 added into the original flow.

The strategy of traffic obfuscation by adding extra packets and bytes always generates a large amount of noise data and brings high costs to network transmission and processing, thereby destroying existing protocols or reducing network service quality. How to effectively reduce the network overhead caused by the noise is one of the core problems in the traffic obfuscation.

In the process of updating the adversarial sample, it is not allowed to reduce the packet size; that is, relative to the packet size in the original sequence, the corresponding packet size in the adversarial sample sequence follows the monotonic nondecreasing rule.

This paper proposes the following two improved methods to realize the obfuscation traffic generation strategy:

(1) After a fixed number of iterations, the adversarial sample is corrected.

(2) In each iteration, delete the illegal disturbance direction, set a larger EPS value for a single iteration, and set a smaller EPS value for truncation of the final output.

In the first method, when the fixed number of iterations is set to 1, it is necessary to compare the adversarial sample with the original sample at the last step of each iteration process, calculate the disturbance vector, reset all negative values in the disturbance vector to 0, and then add this disturbance to the original sample to complete the update. When the fixed number of iterations is set too small, the adversarial samples may always update in the wrong direction. Therefore, it is necessary to set a large value to correct the sequence.

The optimization process of the first method on BIM and MIM algorithms is shown in Algorithms 1 and 2.

In the second method, the illegal disturbance direction is deleted in each iteration, which is the same as setting the number of fixed iterations to 1 in method 1. In order to reduce the negative impact of the search direction, consider using a larger search step in the update process, that is, setting a larger EPS value. Finally, the adversarial samples are limited to the legal range by a small EPS value.

The optimization process of the second method on BIM and MIM algorithms is shown in Algorithms 3 and 4.

4. Black-Box Attack

4.1. Architecture. The obfuscated traffic generation framework based on black-box attack is shown in Figure 3, which is mainly divided into five steps: data set division, data enhancement, training the original model and alternative model, constructing adversarial traffic samples, and finally verifying the effectiveness of the attack algorithm.

Step 1: divide the original data set into three parts: two training sets and one test set. One training set is used to train the original model and the other is used to train the alternative model. The test set is used to construct antitraffic samples.

Step 2: since the scale of a training set 2 is smaller than that of training set 1 if the training set 2 is directly used to train the alternative model, the alternative model will overfit the data set and lack generalization. Therefore, the adversarial traffic sample constructed by it has no transitivity and is no longer applicable to other models. Therefore, it is necessary to generate new training samples according to a training set 2 to obtain additional data. This process is called data enhancement.

Step 3: training set 1 is used as the training set of the original model and training set 3 is used as the training set of the alternative model. Different machine learning algorithms are used to train the original model (including KNN, random forest, DNN, CNN, and LSTM) and the alternative model (including DNN, CNN, and LSTM) and persist the model.

Step 4: use MIM and CW adversarial attack algorithms to construct adversarial traffic samples on the alternative model. For each algorithm and model, a corresponding confusing traffic set is generated, and the size of the traffic set is similar to that of the test set.

Step 5: verify the effectiveness of the black-box attack, explore the transitivity of adversarial traffic samples between different models, and use the confused traffic samples in the confused traffic set to make the success rate of misclassification of the original model as its evaluation index. There are a total of 30 combinations of original models, alternative models, and adversarial attack algorithms.

4.2. Black-Box Attack-Based SMOTE. Black-box attackers have no knowledge of the training data set, learning model, and other relevant information of the original model, assuming that they can only master no more than the amount of data used by the original model. Therefore, black-box attackers need to establish a powerful alternative model to make adversarial samples. Because the adversarial samples are transitive, they will still be misclassified by the original model. Because the amount of data mastered by the black-box attacker is insufficient, it is necessary to expand the original data set through a series of data enhancement means to make the limited data produce more equivalent data. In the selection of alternative models, we need to take into account the performance of the model itself and the effectiveness of the attack algorithm applied to the model.

4.2.1. Transitivity of Adversarial Samples. The transitivity of adversarial samples is the basis of a black-box attack. This feature means that the adversarial samples generated by one model are still valid for another model, which is used to deal with the same task. For many samples, their gradient directions on different models are orthogonal to each other.

```
    Input: classifier $f$; loss function $J$; original sample $x$; authentic label $y$; disturbance size $\alpha$; fixed number of iterations $n$; number of
    iterations $T$.
    output: adversarial sample. $x^{'}$
(1)   $\epsilon = \alpha/T$;
(2)   $x_0' = x$;
(3)   for $k = 0$ to $T - 1$ do
(4)     get gradient $\nabla_x J(x_k', y)$ of func $f$ with respect to $x_k'$;
(5)     //update the adversarial sample $x_{k+1}^{'}$
(6)     $x_{k+1}^{'} = x_k' - \alpha \cdot \text{sign}(\nabla_x J(x_k', y))$;
(7)     if $k\%n = 0$ then
(8)       //modify $x_{k+1}^{'}$
(9)     $\gamma = x_{k+1}^{'} - x$;
(10)    set all elements greater than 0 in the vector $\gamma$ to
(11) 0;
(12)      $x_{k+1}^{'} = x_{k+1}^{'} - \gamma$;
(13)    end if
      end for.
      return $x_{k+1}^{'}$;
```

Algorithm 1: Improved BIM using method 1.

```
    Input: classifier $f$; loss function $J$; original sample $x$; authentic label $y$; disturbance size $\alpha$; fixed number of iterations $n$; number of
    iterations $T$; attenuation factor $\mu$.
    output: adversarial sample. $x^{'}$
(1)   $\epsilon = \alpha/T$;
(2)   $g_0 = 0, x_0' = x$;
(3)   for $k = 0$ to $T - 1$ do
(4)     get gradient $\nabla_x J(x_k', y)$ of func $f$ with respect to $x_k'$;
(5)     //update momentum $v_{k+1}$
(6)     $v_{k+1} = \mu \cdot v_k + \nabla_x J(x_k', y)/\nabla_x J(x_k', y)_1$;
(7)     //update adversarial sample $x_{k+1}^{'}$
(8)     $x_{k+1}^{'} = x_k' - \alpha \cdot \text{sign}(v_{k+1})$;
(9)     if $k\%n = 0$ then
(10)      //modify $x_{k+1}^{'}$
(11)  $\gamma = x_{k+1}^{'} - x$;
(12)  set all elements greater than 0 in the vector $\gamma$ to
(13) 0;
(14)    $x_{k+1}^{'} = x_{k+1}^{'} - \gamma$;
(15)  end if
      end for.
      return. $x_{k+1}^{'}$;
```

Algorithm 2: Improved MIM using method 1.

When using the gradient-based adversarial attack method, they will search in the same antiattack direction. Because different models may still have similar decision boundaries, which are very consistent, and the boundary diameter along the gradient direction is smaller than that in the random direction, moving along the gradient direction of the variable will significantly change the value of the loss function [30], thus changing the output values of two different models at the same time.

To evaluate the transitivity of adversarial samples, two models are usually required, one is the original model, and the other is the alternative model. The proportion that can misclassify the alternative model in all adversarial samples generated by the original model is calculated as the measurement standard. The ratio between the number of transitive adversarial samples and the total number of adversarial samples is also called the matching rate. The higher matching rate means better transitivity.

4.2.2. Data Enhancement. In the black-box attack, the amount of data mastered by the attacker is much smaller than the training set size of the original model. In order to make the decision boundary of the alternative model approximate to the original model, the attacker needs to explore the input domain and generate a comprehensive data

```
     Input: classifier f; loss function J; original sample x; authentic label y; disturbance size $\epsilon_1, \epsilon_2$ ; fixed number of iterations n; number of iterations T.
     output: adversarial sample. $x^{'}$
(1)  $\epsilon = \epsilon_1$;
(2)  $x'_0 = x$;
(3)  for $k = 0$ to $T - 1$ do
(4)      get gradient $\nabla_x J(x'_k, y)$ of func f with respect to $x'_k$;
(5)      //update adversarial sample $x^{'}_{k+1}$
(6)      $x^{'}_{k+1} = x'_k - \alpha \cdot \text{sign}(\nabla_x J(x'_k, y))$;
(7)      if $k\%n = 0$ then
(8)          //modify $x^{'}_{k+1}$
(9)          $\gamma = x^{'}_{k+1} - x$;
(10)         set all elements>0 in the vector $\gamma$ to 0;
(11)         $x^{'}_{k+1} = x^{'}_{k+1} - \gamma$;
(12)     end if
(13) end for
(14) //according to $\epsilon_2$, truncate $x^{'}_{k+1}$
(15) set all elements greater than 0 in the vector $x^{'}_{k+1}$ to $\epsilon_2$;
     return $x^{'}_{k+1}$;
```

Algorithm 3: Improved BIM using method 2.

```
     Input: classifier f; loss function J; original sample x; authentic label y; disturbance size $\alpha$; fixed number of iterations n; number of iterations T; attenuation factor $\mu$.
     output: adversarial sample. $x^{'}$
(1)  $\epsilon = \epsilon_1$;
(2)  $g_0 = 0, x'_0 = x$;
(3)  for $k = 0$ to $T - 1$ do
(4)      get gradient $\nabla_x J(x'_k, y)$ of func f with respect to $x'_k$;
(5)      //update momentum $v_{k+1}$
(6)      $v_{k+1} = \mu \cdot v_k + \nabla_x J(x'_k, y)/\nabla_x J(x'_k, y)_1$;
(7)      //update adversarial sample $x^{'}_{k+1}$
(8)      $x^{'}_{k+1} = x'_k - \alpha \cdot \text{sign}(v_{k+1})$;
(9)      if $k\%n = 0$ then
(10)         modify $x^{'}_{k+1}$ according to x;
(11)     end if
(12) end for
(13) //according to $\epsilon_2$, truncate $x^{'}_{k+1}$;
(14) set all elements greater than 0 in the vector $x^{'}_{k+1}$ to $\epsilon_2$;
     return $x^{'}_{k+1}$;
```

Algorithm 4: Improved MIM using method 2.

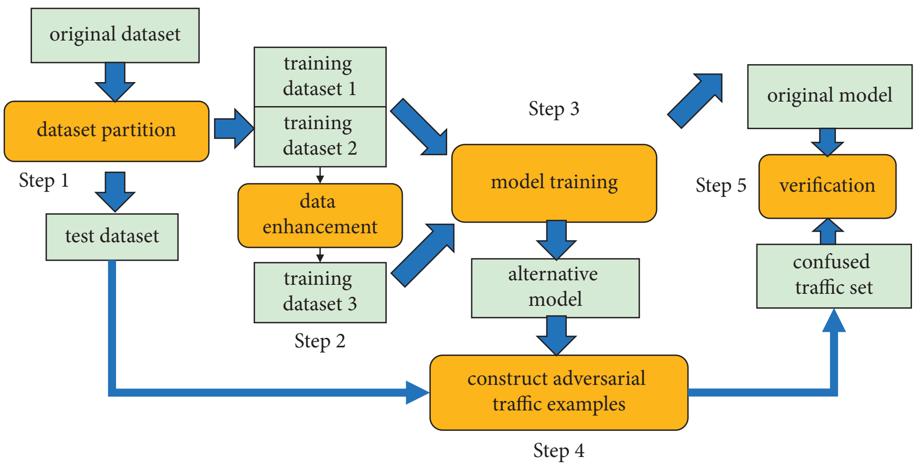


Figure 3: Generating obfuscated traffic based on black-box attack.

set according to a small part of the initially collected data. This process is called data enhancement.

In the field of image classification, there are many data enhancement strategies based on spatial geometric transformation, such as rotation, clipping, rotation, scaling deformation, affine transformation, and so on. The data enhancement strategies of noise class and fuzzy class can also generate new training samples. The former randomly superimposes some noise on the basis of the original picture, and the latter realizes pixel smoothing by reducing the difference of pixel values. SMOTE [31] and SamplePairing are diverse data enhancement methods, that is, using multiple samples to generate new samples. SMOTE is based on difference; it can synthesize new samples for small sample classes and solve the problem of sample imbalance while enhancing data. In this paper, SMOTE method is used as the data enhancement method of black-box attack.

SMOTE sample is a linear combination of two similar samples from a few classes, which is defined as follows:

$$s = x + u \cdot \left(x^R - x\right), \tag{9}$$

where $0 \leq u \leq 1$, x^R is a vector randomly selected from the k nearest neighbors of X, and K is set to 5 by default. The specific process of synthesizing new sample points is shown in Figure 4, in which red sample points are x, blue is the five real sample points closest to x, and green is other real sample points.

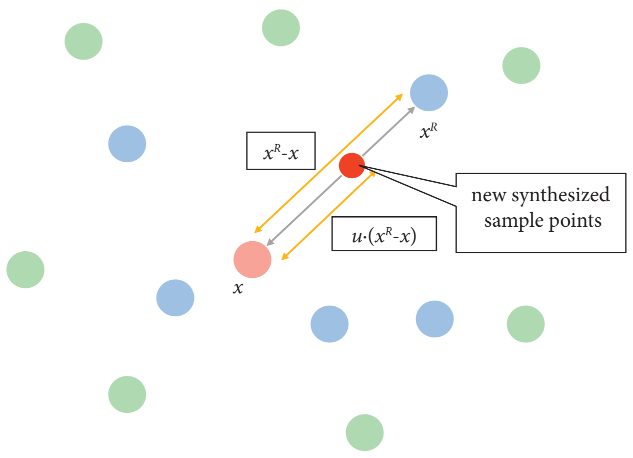


Figure 4: SMOTE algorithm.

4.2.3. Alternative Model. In the black-box attack, the alternative model is a machine learning model that is really used to construct antitraffic samples. It “replaces” the original model that the attacker cannot obtain to perform the adversarial attack and finally verifies the effectiveness of these adversarial traffic samples on the original model. This chapter constructs five original models based on five algorithms: KNN, random forest, DNN, CNN, and LSTM. Since KNN and random forest algorithms have no gradient, only three algorithms such as DNN, CNN, and LSTM are selected as alternative models.

There are 15 different combinations of original models and alternative models. When the architecture/algorithm of the original model and the alternative model are different, the attacker only has a small amount of input data information; when the architecture/algorithm of the original model and the alternative model are the same, the attacker can master more model information. These combinations can be divided into two parts according to the degree of information the attacker has, as shown in Table 1.

5. Experiment

5.1. Data Set. This paper adopts the public data set TornonTor (ISCXTor2016) of Canadian Institute of network security [32]. The data size is 22 Gb, including 7 types of traffic, as follows:

(1) *Browsing*: HTTPS traffic generated by browsers (chrome, Firefox).

(2) *e-mail*: The sender sends mail through SMTP/s, and the receiver receives the traffic generated by mail using POP3/SSL and IMAP/SSL, respectively.

(3) *Chat*: Instant messaging software (Facebook, hangouts, Skype, aim, ICQ).

(4) *Streaming*: Skype, FTP over SSH (SFTP), and FTP over SSL (FTPS).

(5) *VoIP*: voice calls from Facebook, hangouts, and Skype.

(6) *P2P*: uTorrent and transmission download different torrent files from the public repository

Through the statistics of the original pcap packet content, the number of streams and the total number of packets of each type of traffic can be obtained. In this paper, only one-way flow is considered, and each flow is intercepted with 10s as the time threshold. The statistical results are shown in Table 2.

5.2. White-Box Attack Experiment. The attack success rate is the ratio that the confused traffic set is recognized as positive samples by the classification model. The EPS parameter range of the three gradient-based white-box attack algorithms is 1–25. A larger EPS parameter range, i.e., 1–35, is tested separately for CNN model. In the improved white-box attack method, the EPS of method 1 is 25, the larger EPS of method 2 is 25, and the smaller EPS is 21.

5.2.1. Gradient Iterative Attack Parameters. In Experiment 1, DNN, CNN, and LSTM binary classification models were trained using the training set, and three gradient iterative attack algorithms of FGSM, BIM, and MIM were implemented. The gradient iterative attack algorithm is applied to the test set to count the attack success rate under different EPS values. Figure 5 shows the experimental results of FGSM algorithm. FGSM is a one-step gradient attack algorithm. Figure 6 shows the experimental results of BIM and MIM gradient iterative attack algorithms.

In Figure 5, the attack success rate of CNN is the lowest. With the increase of EPS, the attack success rate increases

Table 1: Original model and alternative model.

Prior knowledge	Combination of original model and alternative model
Less training data	DNN-DNN, CNN-CNN, LSTM-LSTM
Less training data + original model	DNN-KNN, DNN-random Forest, DNN-CNN, DNN-LSTM CNN–KNN, CNN-random Forest、CNN-DNN, CNN-LSTM LSTM-KNN, LSTM-random Forest, LSTM-DNN, LSTM-CNN

Table 2: Original model and alternative model.

Data set	Number of flows	Number of packets	File size
ori_browsing	68239	833358	557 MB
ori_e-mail	1225	55498	325 MB
ori_chat	967	20109	7.83 MB
ori_streaming	9702	318825	1.97 GB
ori_filetransfer	2814	353272	2.78 GB
ori_voip	7873	312191	229 MB
ori_p2p	72070	1040642	4.01 GB
tor_browsing	1797	89527	482 MB
tor_e-mail	208	24818	349 MB
tor_chat	273	14186	10.9 MB
tor_streaming	1725	193356	2.14 GB
tor_filetransfer	598	72026	3.05 GB
tor_voip	1559	189749	625 MB
tor_p2p	1484	188488	4.68 GB

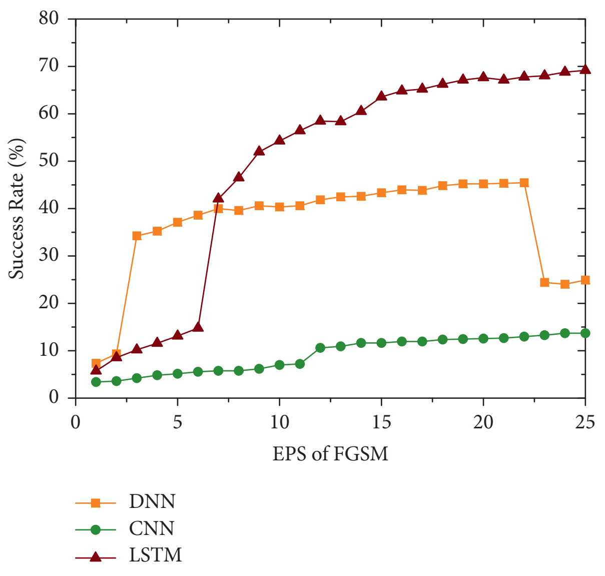


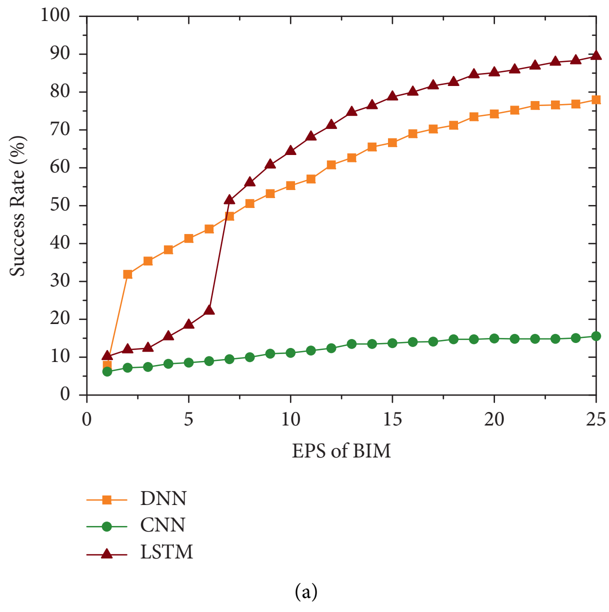
Figure 5: One-step gradient attack algorithm.

very slowly. When EPS increases from 1 to 5, the curves of the DNN and LSTM show an upward trend, and the accuracy of DNN is higher than that of LSTM; when EPS increases from 5 to 10, the accuracy of LSTM exceeds that of DNN. When EPS changes to 33, the attack success rate of FGSM decreases rapidly, from about 45% to 25%. FGSM algorithm is applied to DNN, CNN, and LSTM traffic classification models, and the highest attack success rates are 45.45%, 13.7%, and 69.17%, respectively.

In Figure 6, the experimental results of BIM algorithm are shown in (a), and the experimental results of MIM algorithm are shown in (b). Compared with the experimental results of FGSM algorithm, DNN, CNN, and LSTM show similar change trends. However, for DNN curve, there is no sharp decline in attack success rate within the range of 1–25 limited by EPS value. With the increase of EPS, the corresponding attack success rate in the three curves shows an upward trend. In addition, for MIM algorithm, the attack success rate against DNN and LSTM finally achieves a similar result. BIM algorithm is applied to DNN, CNN, and LSTM traffic classification models, and the highest attack success rates are 77.96%, 15.55%, and 89.43%, respectively. MIM algorithm is applied to three traffic classification models: DNN, CNN, and LSTM. The highest attack success rates are 89.66%, 17.1%, and 91.46%, respectively.

The results of Experiment 1 show that the gradient iterative attack algorithm has a higher attack success rate than the one-step gradient attack algorithm, and the MIM algorithm has better performance than the BIM algorithm. The gradient-based adversarial attack algorithm is more effective for DNN and LSTM, but not for CNN.

For the white-box attack of CNN, because Figures 5 and6 cannot describe the change trend of CNN curve in detail, expand the EPS parameter range to 1–35 for CNN, and use three gradient-based adversarial attack algorithms to experiment, respectively. The experimental results are shown in Figure 7; three different color curves are used to depict the change of attack success rate of FGSM, BIM, and MIM gradient methods applied to CNN. When EPS changes to 28, the attack success rate of MIM algorithm is significantly improved, from 18.43% to 38.11%, and then the curve is still stable. With the increase of EPS, FGSM curve and BIM curve rise slowly. The experimental results show that it is necessary to set a larger EPS parameter value to improve CNN



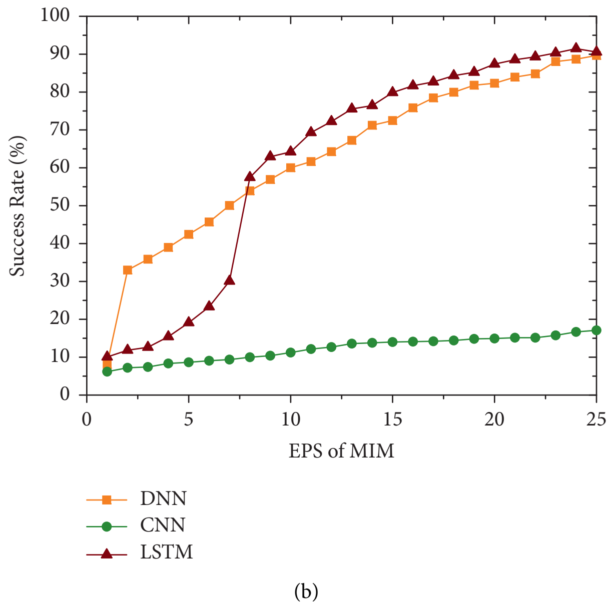


Figure 6: Gradient iterative attack algorithm. (a) Basic iterative method. (b) Momentum iteration method.

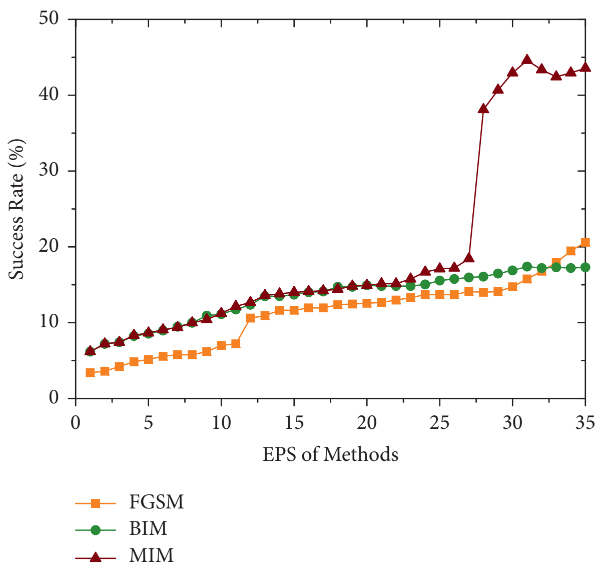


Figure 7: White-box attack against CNN.

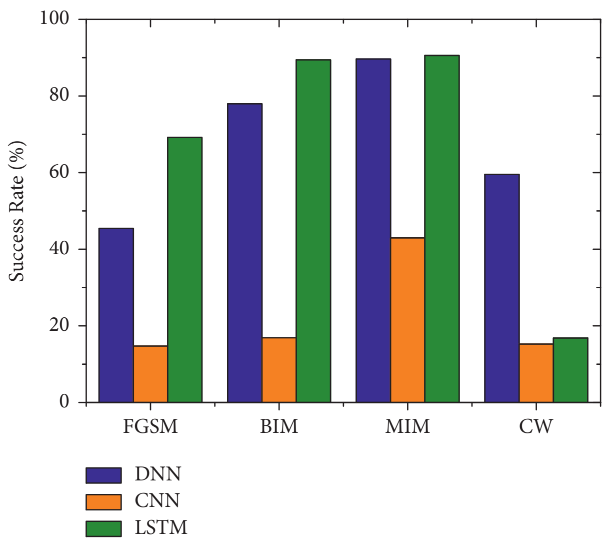


Figure 8: Comparison of four white-box attack algorithms.

performance. The experimental results show that the performance of MIM is obviously better than FGSM and BIM algorithms.

5.2.2. Comparison of White-Box Attack Algorithms. In Experiment 2, four white-box attack algorithms FGSM, BIM, MIM, and CW are completely applied to the deep learning classification model, and the attack success rates of the above four white-box attack algorithms are compared. The experimental results are presented in the form of histogram, as shown in Figure 8.

In Figure 8, for the three deep learning classification models of DNN, CNN, and LSTM, the attack success rates of FGSM algorithm are 45.45%, 14.73%, and 69.17%, respectively, the attack success rates of BIM algorithm are 77.96%, 16.89%, and 89.43%, respectively, the attack success rates of MIM algorithm are 89.66%, 42.95%, and 90.57%, respectively, and the attack success rates of CW algorithm are 59.53%, 15.24%, and 16.82% respectively. Experimental results show that MIM algorithm is better than FGSM, BIM, and CW algorithms. In addition, for LSTM, CW algorithm shows worse attack effect. For the other three algorithms, LSTM is easier to attack successfully. Compared with DNN and LSTM, CNN is more robust against white-box attacks.

5.2.3. Improved White-Box Attack Performance. Experiment 3 tests the performance of the improved white-box attack algorithm. In order to ensure that the corresponding packet size in the adversarial sample sequence follows the monotonic nondecreasing rule, two improved methods are proposed in Section 3.2.5. Method 1 is to modify the adversarial sample after a fixed number of iterations. The fixed number of iterations selected in this experiment is 100 and the total number of iterations is 1000. Method 2 is to delete the illegal disturbance (negative change compared to the original value) in each iteration and set a larger EPS for a single iteration and a smaller EPS for the final output. The larger EPS selected in this experiment is 25 and the smaller EPS is 21. Compared with other algorithms, the MIM algorithm shows a better attack success rate. Therefore, in experiment 3, the MIM algorithm is selected as the basic method to test the performance of the improved white-box attack algorithm, and DNN and LSTM are selected as the attack objects, respectively. The experimental results are shown in Figure 9.

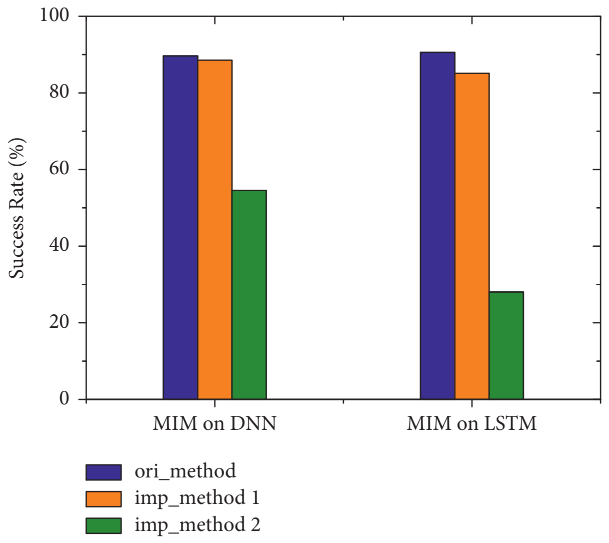


Figure 9: Improved white-box attack performance.

5.3. Black-Box Attack Experiment. The experiment is divided into two steps. The first step is to build an alternative model and test its performance. The second step is to test the success rate of black-box attacks under different model combinations.

5.3.1. Alternative Model Performance. The experiment first needs to build the original model and alternative model. Figure 10 shows the prediction accuracy of the alternative model and compares it with the experimental results of the original model. Figure 11 shows the attack success rate against the alternative model and compares it with the experimental results of the original model.

In Figure 10, for the three deep learning algorithms of DNN, CNN, and LSTM, the accuracy of the original model is 94.77%, 90.02%, and 97.68%, respectively, and the accuracy of the alternative model is 92.74%, 86.34%, and 91.53%, respectively. The experimental results show that because the training set used by the alternative model is smaller than the original model, even if SMOTE data enhancement technology is applied, the recognition accuracy is still lower and the change rate is smaller than the original model.

In Figure 11, the experimental results show that the original model has a lower attack success rate than the alternative model in the face of the same antiattack method; that is, the larger the scale of the model training data set, the stronger the adversarial attack ability of the model.

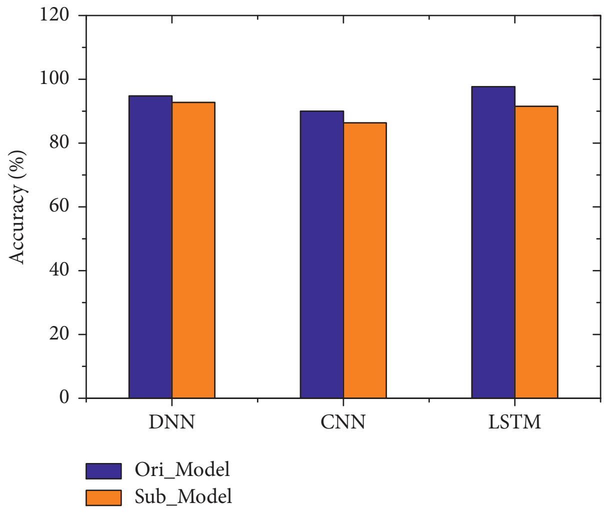


Figure 10: Comparison of accuracy between original model and alternative model.

5.3.2. Black-Box Attack under Different Combinations. For the combination of the original model and alternative model listed in Table 1, MIM and CW are used to carry out black-box attacks, respectively, to verify the transferability of adversarial samples. The experimental results are shown in Table 3.

In Table 3, for KNN and random forest original models, the obfuscation traffic set generated by MIM on the LSTM alternative model is the most effective, and the success rates of black-box attacks are 34.46% and 66.49%, respectively. For the original models of DNN, CNN, and LSTM, the obfuscation traffic set generated by the original model of MIM under the same architecture is the most effective, and the success rates of black-box attack are 47.2%, 42.12%, and 73.63%, respectively. The original LSTM is most vulnerable to black-box attacks. The attack success rate of the confused traffic set generated by MIM on DNN, CNN, and LSTM alternative models is 67.39%, 65.1%, and 73.63%, respectively. Compared with MIM, the adversarial traffic samples generated by CW are almost nontransitive.

5.3.3. Comparison with Advanced Models. Muhammad et al. [29] proposed an advanced mutual information (MI) model and adopts DNN and SVM models. Our SMOTE technology for data enhancement is compared with it in binary class, and the results are shown in Table 4. Obviously, the success rate of smote attack is more than 20% higher than that of MI

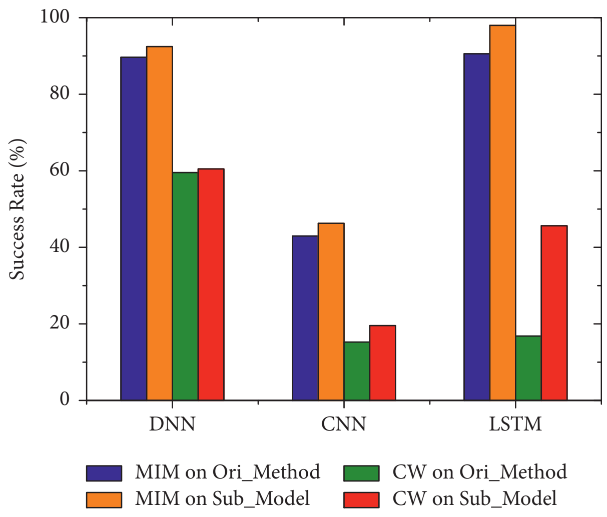


Figure 11: Comparison of attack success rate between original model and alternative model.

Table 3: Black-box attack success rate under different combinations.

Combination	KNN (%)	Random forest (%)	DNN (%)	CNN (%)	LSTM (%)
DNN + MIM	15.98	33.9	**47.2**	14.73	67.39
DNN + CW	1.5	11.69	7.1	3.5	6.88
CNN + MIM	33.83	58.18	17.68	**42.12**	65.1
CNN + CW	1.37	11.56	2.37	1.75	6.62
LSTM + MIM	**34.46**	**66.49**	18.31	16.27	**73.63**
LSTM + CW	3.62	11.95	1.62	0.21	2.17

Table 4: Black-box attack success rate in binary class between MI and SMOTE.

ML	MI (%)	SMOTE (%)
DNN	22.58	47.2
SVM	22.62	43.4

attack in DNN and SVM. In the black-box attack, the sample size is small, and the MI method is prone to sample imbalance. The SMOTE method based on difference makes up for the defect of MI.

6. Conclusions

In order to resist man-in-the-middle traffic analysis attack, this paper focuses on network traffic obfuscation. We implement and test adversarial attack methods based on gradient, including FGSM, BIM, MIM, and optimization adversarial attack methods (including CW). For the gradient iterative method, an improved attack method adapted to the real strategy is proposed. The adversarial samples generated by this method meet the monotonic nondecreasing rule of packet size. Based on the white-box attack algorithm, we design and test a black-box attack method by training the alternative model. This method needs to find a reliable alternative model, and the decision boundary of the alternative model is similar to the original model, so the adversarial samples generated by the alternative model still have a certain effect on the original model. A series of black-box attack experiments are carried out for different original model and alternative model combinations, in which the alternative model uses SMOTE technology for data enhancement. Facing the same attack method, the original model has a lower attack success rate than the alternative model.

Data Availability

The data and source code used to support the findings of this study can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Grant no. 62172123 and no. 61732022), Natural Science Foundation of Heilongjiang Province of China (Grant no. YQ2021F007), National Key Research and Development Program of China (Grant no. 2018YFB1800702), and the Basic Research Program (no. JCKY2019604B002).

References

[1] D. Fifield, C. Lan, R. Hynes, P. Wegmann, and V. Paxson, "Blocking-resistant communication through domain fronting," *Privacy Enhancing Technologies*, vol. 1, no. 2, pp. 46–64, 2015.

[2] A. Ghosh and A. Senthilrajan, "Research on packet inspection techniques," *International Journal Of Scientific & Technology Research*, vol. 8, pp. 2068–2073, 2019.

[3] L. Dixon, T. Ristenpart, and T. Shrimpton, "Network traffic obfuscation and automated internet censorship," *IEEE Security & Privacy*, vol. 14, no. 6, pp. 43–53, 2016.

[4] S. Dong, "Multi class SVM algorithm with active learning for network traffic classification," *Expert Systems with Applications*, vol. 176, no. Aug, p. 114885, 2021.

[5] X. Ren, H. Gu, and W. Wei, "Tree-RNN: tree structural recurrent neural network for network traffic classification," *Expert Systems with Applications*, vol. 167, Article ID 114363, 2021.

[6] S. Dong and Y. Xia, "Network traffic identification in packet sampling environment," *Digital Communications and Networks*, vol. 41, 2022.

[7] K. George and M. Nick, "obfs2(The twobfuscator)," https://gitweb.torproject.org/pluggable-transports/obfsproxy.git/tree/doc/obfs2/obfs2-protocol-spec.txt.

[8] K. George and M. Nick, "obfs3(The Threebfuscator)," https://gitweb.torproject.org/pluggable-transports/obfsproxy.git/tree/doc/obfs3/obfs3-protocol-spec.txt.

[9] B. Wiley, "Dust: A blocking-resistant internet transport protocol," Technical report, 2011, http://blanu.net/Dust.pdf.

[10] S. Köpsell and U. Hillig, "How to achieve blocking resistance for existing systems enabling anonymous web surfing," *2004 ACM Wksp. On Privacy in the Electronic Society Ser. WPES '04*, ACM, vol. 7, pp. 47–58, 2004.

[11] Z. Weinberg, J. Wang, V. Yegneswaran, L. Briesemeister, S. Cheung, and F. Wang, "Stegotorus: a camouflage proxy for the tor anonymity system," *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, vol. 53, pp. 109–120, 2012.

[12] Q. Tan, Y. Gao, J. Shi, X. Wang, B. Fang, and Z. Tian, "Toward a comprehensive insight into the eclipse attacks of Tor hidden services," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1584–1593, 2019.

[13] P. Winter, T. Pulls, and J. Fuss, "Scramblesuit: a polymorphic network protocol to circumvent censorship," *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*, ACM, vol. 12, pp. 213–224, 2013.

[14] A. Yawning and W. Philipp, "obfs4 (The obfourscator)," https://gitweb.torproject.org/pluggable-transports/obfs4.git/tree/doc/obfs4-spec.txt.

[15] D. J. Bernstein, T. Lange, and P. Schwabe, "The Security Impact of a New Cryptographic Library," *Conf. Cyptology & Inform*, pp. 159–176, SecurityLatin, America, Oct 2012.

[16] C. V. Wright, S. E. Coull, and F. Monrose, "Traffic morphing: an efficient defense against statistical traffic analysis," in *Proceedings of the Network and Distributed Security Symposium*, pp. 237–250, 2009.

[17] Q. Wang, X. Gong, G. Nguyen, A. Houmansadr, and N. Borisov, "Censorspoofer: asymmetric communication using ip spoofing for censorship-resistant web browsing," *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, vol. 43, pp. 121–132, 2012.

[18] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton, "Protocol misidentification made easy with format-transforming encryption," *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, pp. 61–72, ACM, 2013.

[19] H. Mohajeri Moghaddam, B. Li, M. Derakhshani, and I. Goldberg, "Skypemorph: protocol obfuscation for tor bridges," *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, ACM, vol. 63, pp. 97–108, 2012.

[20] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. R. Karger, "Infranet: circumventing web censorship and surveillance," *Proc. USENIX Security Symp*, vol. 33, pp. 247–262, 2002.

[21] Google, "foe-project," https://code.google.com/archive/p/foe-project/e.

[22] C. Brubaker, A. Houmansadr, and V. S. CloudTransport, "Using cloud storage for censorship-resistant networking," *Proceedings of the 14th Privacy Enhancing Technologies Symposium (PETS)*, 2014.

[23] J. Ning, G. Gui, and Y. Wang, "Malware Traffic Classification Using Domain Adaptation and Ladder Network for Secure," *IEEE Internet of Things Journal*, vol. 12, 2021.

[24] Z. He, "Edge Device Identification Based on Federated Learning and Network Traffic Feature Engineering," *IEEE Trans. Cogn. Commun. Netw*, vol. 67, 2021.

[25] I. J. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and Harnessing Adversarial Examples*, 2014.

[26] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial Examples in the Physical World," *Artificial Intelligence Safety and Security*, pp. 99–112, Chapman and Hall/CRC, 2018.

[27] Y. Dong, "Boosting adversarial attacks with momentum," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9185–9193, Salt Lake City, UT, USA, June 2018.

[28] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy*, pp. 39–57, San Jose, CA, USA, May 2017.

[29] M. Usama, A. Qayyum, J. Qadir, and A. Fuqaha, "Black-box adversarial machine learning attack on network traffic classification," *Proc. Int. Wireless Commun. Mobile Comput. Conf.* vol. 1, no. 1, pp. 84–89, Jun 2019.

[30] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," *Proc. Int. Conf. Learn. Represent. (ICLR)*, vol. 41, pp. 1–14, 2017.

[31] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[32] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," *Proc. 3rd Int. Conf. Inf. Syst. Security Privacy*, vol. 18, pp. 253–262, 2017.