Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 7367107, 12 pages
https://doi.org/10.1155/2021/7367107

Research Article

A Novel Way to Generate Adversarial Network Traffic Samples against Network Traffic Classification

Yongjin Hu, Jin Tian, and Jun Ma

Information Engineering University, Zhengzhou 450001, China

Correspondence should be addressed to Jun Ma; sijunhan@163.com

Received 29 April 2021; Revised 9 July 2021; Accepted 12 August 2021; Published 26 August 2021

Academic Editor: James Ying

Network traffic classification technologies could be used by attackers to implement network monitoring and then launch traffic analysis attacks or website fingerprint attacks. In order to prevent such attacks, a novel way to generate adversarial samples of network traffic from the perspective of the defender is proposed. By adding perturbation to the normal network traffic, a kind of adversarial network traffic is formed, which will cause misclassification when the attackers are implementing network traffic classification with deep convolutional neural networks (CNN) as a classification model. The paper uses the concept of adversarial samples in image recognition for reference to the field of network traffic classification and chooses several different methods to generate adversarial samples of network traffic. The experiment, in which the LeNet-5 CNN is selected as a classification model used by attackers and Vgg16 CNN is selected as the model to test the transferability of the adversarial network traffic generated, shows the effect of the adversarial network traffic samples.

1. Introduction

As a basic technology for enhancing network controllability, network traffic classification technology helps researchers understand traffic distribution, optimize network transmission, and improve network service quality; however, it is often leveraged by attackers for monitoring network traffic against the network targets and classifying the application types (such as mail, multimedia, and websites) the network traffic belong to. Based on the classification results, network traffic interception is implemented and a possible website fingerprint attack may be followed [1]. In particular, the network traffic classification, in which area machine learning and deep learning are applied, provides attackers easier conditions that result in extremely high classification accuracy. A typical scenario for a network traffic classification method based on deep learning that is used by attackers is shown in Figure 1.

Although the application of deep learning in network traffic classification can improve the accuracy of classification and has demonstrated huge potential in areas such as image recognition and natural language processing, adversaries against the deep learning models including the convolutional neural networks (CNN) have raised the interest of scholars on the concept of "Adversarial Sample" that was introduced to the area of computer vision by Szegedy et al. [2].

In the study of image recognition, Szegedy has found that CNN tends to give an error output with high confidence degrees when intentionally adding some undetectable and tiny perturbations to the input samples of the learning models. For deep learning models, these are called "Adversarial Samples" that are crafted by these tiny perturbations to the original dataset. From the perspective of attack, the most direct application of adversarial samples is in the area of computer vision, including face identification and automatic driving. By adding perturbation undetected by eyes to the image, failures in face identification and traffic signs [3] are triggered and damages from misclassification are then caused. In the area of information security, it can also lead to detection avoidance [4] by deceiving the malware detection models based on neural network. However, on the contrary, the adversarial samples, from the perspective of defense, are also of high value. First, it can improve the robustness of deep learning models in responding to possible

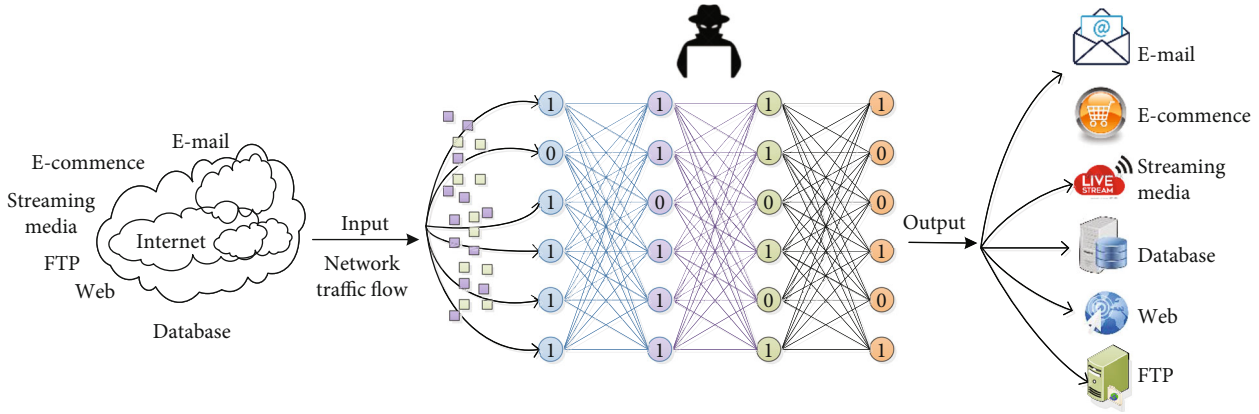

Figure 1: A typical scenario for network traffic classification method based on deep learning.

Table 1: Five types of flow in network traffic classification.

Flow granularity	Points of interest
TCP connections	Heuristics based on the observation of some TCP flags (i.e., SYN, FIN, and RST) or TCP state machines are used to identify the start and the end of each connection.
Flow	A typical flow definition uses the 5-tuple {source (IP), source (port), destination (IP), destination (port), and transport-level protocol}.
Bidirectional flows	Same as above, but includes both directions of traffic, assuming both directions of flows can be observed (especially challenging on backbones where internet routing is often asymmetric).
Services	Typically defined as all traffic generated by an IP-port pair.
Hosts	Some approaches classify a host by the predominant traffic it generates, assuming both directions of traffic (to and from the host) can be observed.

adversarial sample attack by being trained with adversarial samples generated in advance [5]. Second, the adversarial samples can be leveraged to deceive the classification models by attackers using the deep learning network, which results in misclassification and increase of attack cost, thus cancelling the attacks. From the second view above, this paper is designed for defenders to trigger errors in attackers' network classification by crafting adversarial samples for network traffic with the addition of perturbation and thus forming deceptive network traffic against attackers' network traffic classification attacks.

In this paper, the concept of adversarial samples is introduced to defend the network traffic classification attacks initiated by attackers. Adversarial samples of network traffic are generated to deceive network traffic classification models based on deep learning network used by the attackers, resulting in misclassification and attack failure. The contributions of the paper are as follows: firstly, the concept of adversarial samples is introduced into network traffic as a view of active defense, and deceptive effects initiated by different adversarial samples are compared. Secondly, contrary to the fact that attackers initiate attacks with adversarial samples in other areas, the adversarial samples of a network are considered as a defensive way to confuse the attackers' classification models, that can be regarded as "attacks in active defense." Finally, the LeNet-5 CNN is selected as a network traffic classification model used by attackers to be deceived, and Vgg16 CNN is chosen as the model to test the transferability of the adversarial network traffic generated.

2. Related Work

2.1. Network Traffic Classification. Based on the granularity of network traffic, the study in network traffic classification is mainly for the following three levels [6]: packet, flow, and stream. In the three levels mentioned above, the flow level includes five types of flow network traffic according to different granularities [7] as shown in Table 1, which are the most widely used.

In this paper, flow network traffic is used as the original data. By crafting adversarial samples of network traffic, the defenders deceive the attackers who use deep learning methods as their classification models. The classification methods based on deep learning assume that the statistical characteristics (such as flow duration distribution) of the network layers for some types of applications are unique. These methods, including Decision Tree, Naive Bayes, Support Vector Machine, Association Rules Learning, Neural Network, and Genetic Algorithm, are applied in the classification model's construction to classify, with such characteristics as broad scenarios, high classification accuracy, and ability in encrypted data traffic classification.

For studies of traffic classification based on machine learning, the main idea is to construct united statistical

Table 2: Notations of attack model.

Notation	Description	Remark
TF	Network traffic observed by attackers	
X	Feature set of traffic TF	$X = \{x_1, x_2, x_3 \cdots x_m\}$
C	Application type set corresponding to network traffic TF	$C = \{c_1, c_2, c_3 \cdots c_i, \cdots c_n\}$
$F(x)$	Classification function of the classification model	Input value is traffic TF, output value is the probability of the ith application type in application set C

Table 3: Notations of defense model.

Notation	Description	Remarks
P	Perturbation	
TA	The network traffic adversarial samples that are created by defenders with addition of perturbation P into traffic TF	
X'	Feature set of traffic TA	$X' = \left\{x_1', x_2', x_3' \cdots x_m'\right\}$
$F\left(x'\right)$	Classification function of the classification model	Input value is traffic TA, output value is the probability of type i'th of C traffic whereTA belongs to

*Input:*Normal Network Traffic TF
*Output:*Adversarial Samples of Network Traffic TA
BEGIN.
1.Preprocess (TF); //*Pre-process* TF and Extract characteristic X;
2.TranspPcapToIDX (TF); //Transform TF from pcap format to IDX format;
3.Normalized (); //Delaminate each characteristic dimension and normalize into section [0,255];
4.Reshape (TF); //Reshape each characteristic value of multiple types of characteristic as a grey value;
5.Visualization (TF); //Form a 28×28 matrix and visualize the network traffic;
6.Training (TF, mode); //Train CNN models
7.Test (TF); //Test the accuracy of normal network traffic;
8.CraftingPerturbation (method); //use different methods of perturbation crafting to craft perturbation;
9.TA = *GenerateAdvSample ();* //$TA = TF + P$, overlay the perturbation and original traffic to craft adversarial samples of network traffic TA
10.Visualization (TA); //Compare TA and results from Step 5
11.Evaluate (TA); //evaluate adversarial samples of traffic network TA being crafted
12.Return TA; //output adversarial samples of traffic network.
END

Algorithm 1: Adversarial samples of network traffic crafting algorithm.

attributes of traffic as the fingerprint to classify. Ref. [8] applies for the first time machine learning into network traffic and assumes the fact that the bytes in flow can be regarded as pixels in images, and the deep learning method with excellent performance in image recognition can be used for network classification. Ref. [9] integrates feature extraction, feature selection, and classification into an end-to-end framework and calculates the load bytes of different behaviours by first-order CNN to construct fingerprints. Ref. [10] leverages characteristics of anonymized TOR (The Onion Router) network and applies the direction of the length sequence as the input for deep learning networks including SAE (Stacked Auto Encode), CNN, and LSTM (Long Short-Term Memory), to classify the webpage access. Ref. [11] applies for the first time the method of representation learning into the area of malicious network traffic classification, which regards the original traffic data as images, then it conducts classification with CNN that does well in image classification tasks, and finally, it achieves the purpose of classifying the malicious network traffic. These studies have proven the feasibility of deep learning in traffic classification and at the same time, provided targets for adversarial samples of network traffic classification based on deep learning.

For studies in adversarial network traffic classification based on deep learning, Ref. [12] proposes a defense method loading background network traffic and validates the Tor and JAP (Java Anon Proxy) anonymized network. Ref. [13] has validated the effects of encrypted network traffic classification adversary filled by encrypted protocol bytes. Ref. [14] applies different real traffic as noise during website access. Ref. [15] proposes that Walkie-talkie loads a website in

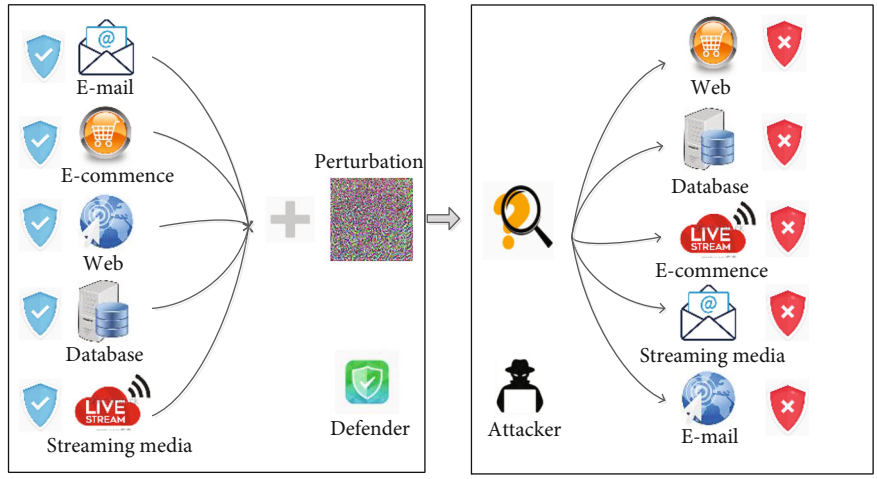

Figure 2: Attack and defense scenario.

Table 4: Experiment's environment and parameters.

Environment	Parameters
OS	Win10,64bit
Processor	CPU: Intel Core i5-7200 U; GPU: NVIDIA GeForce 940MX
Memory	8 GB, LPDDR3, 2133 MHz
Pycharm	Community Edition 2019.3.1
Tensorflow	Ver. 2.1.0
Library support	Cleverhans 3.0.1 [32]

simplex mode to confuse the burst feature. The abovementioned studies have mainly achieved the goal of modifying the communication characteristics of the traffic and have proposed methods that mostly focus on how to avoid being detected, which are of limited ability to disguise and deceive, and with an insufficient adversary. At present, the studies close to our work are those on network traffic disguise and confusion in the area of privacy protection, in which TOR releases obfsproxy, an obfuscated proxy software [16] that makes the encrypted traffic of SSL (Secure Socket Layer) or TLS (Transport Layer Security) look like unencrypted HTTP or instance communication traffic. Ref. [17] releases TOR's transmission layer plug-in, SkypeMorph, to fill the communication traffic between a TOR client and a network bridge to Skype video communication traffic for statistical analysis of adversarial traffic. Ref. [18] proposes the method of analysis of traffic classification rules in a black box, which can infer traffic analysis identification rules through tests and thus modify the communication packet to avoid being detected. However, all these studies neither apply the concept of adversarial samples into those on adversarial traffic analysis nor discuss it as a method of defense for defenders, which, however are the focus of this paper.

2.2. Adversarial Samples. The key of adversarial samples is to craft adversarial perturbation. In the area of computer vision, it is essential for perturbation to meet the requirement of being invisible to human eyes after addition of original images and be able to confuse original classification models. In this paper, the deception for traffic classification models still have to meet certain requirements (e.g., bandwidth), though it is not necessary for the perturbation being crafted to meet the requirement of "being invisible to human eyes."

Now, the majority of studies are focused on crafting the adversarial perturbation to misclassify an image. Szegedy et al. [2] discovered the weakness of the deep neural network in the area of image classification, proposed the concept of adversarial samples, and described the perturbation crafting as an optimized issue for the first time. Goodfellow et al. [19] proposed an optimal method of max norm constrained perturbation, which is called the Fast Gradient Sign Method (FGSM), to improve the computational efficiency and proved that high dimension linearity is the primary reason to make adversarial samples better. Kurakin et al. [20] proposed a basic iteration method that leverages FGSM in iteratively crafting perturbation. Moosavi-Dezfooli et al. [21] discovered adversarial perturbation irrelevant to particular images in image classification models, that is, the existence of universal perturbations, which can lead the classification models to misclassify any image with the addition of this perturbation. Athalye et al. [22] have discovered that the deep network classifier could also be deceived by objects in the real world printed by 3D printers. DeepFool [23] further improved the effectiveness of adversarial perturbation. Metzen et al. [24] introduced Universal Adversarial Perturbation (UAP) for semantic segmentation tasks and extended the iterative FGSM attack of [21] and changed the labels for prediction of each pixel. Mopuri et al. sought data-free universal perturbation without any sample data distribution. They proposed a new algorithm without target data to craft universal adversarial perturbation called FFF [25]. Their later work, GDUAP [26], has improved the attack effect to cause misclassification for different structures and parameter classification models and validated the validity of the method in tasks across computer visions. Furthermore, attacks in other areas are studied besides those on classification and recognition tasks in computer visions, and there is presently no research on attacks against network traffic classification.

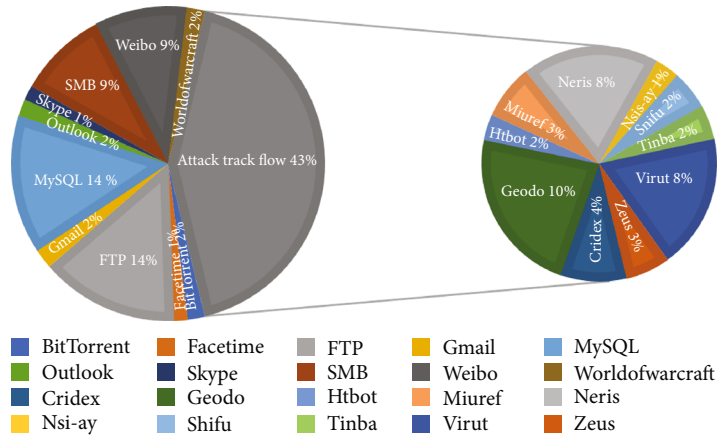

Figure 3: Statistical chart of USTC-TFC2016 dataset distribution.

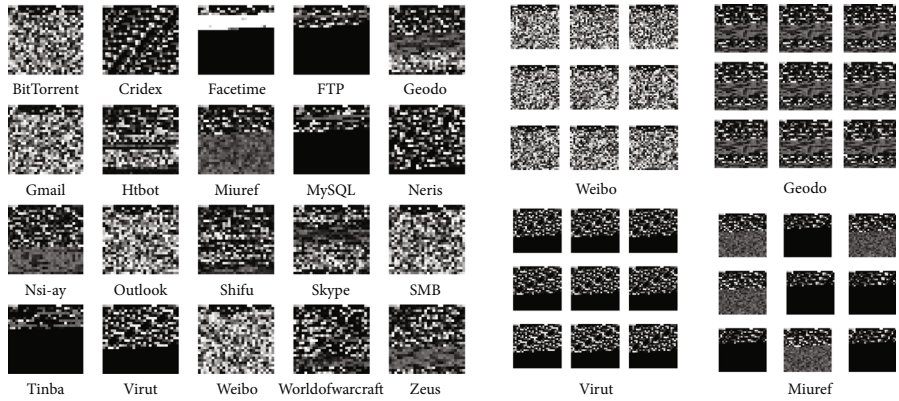

Figure 4: Visualization of 20 types of network traffic and consistency in the same type.

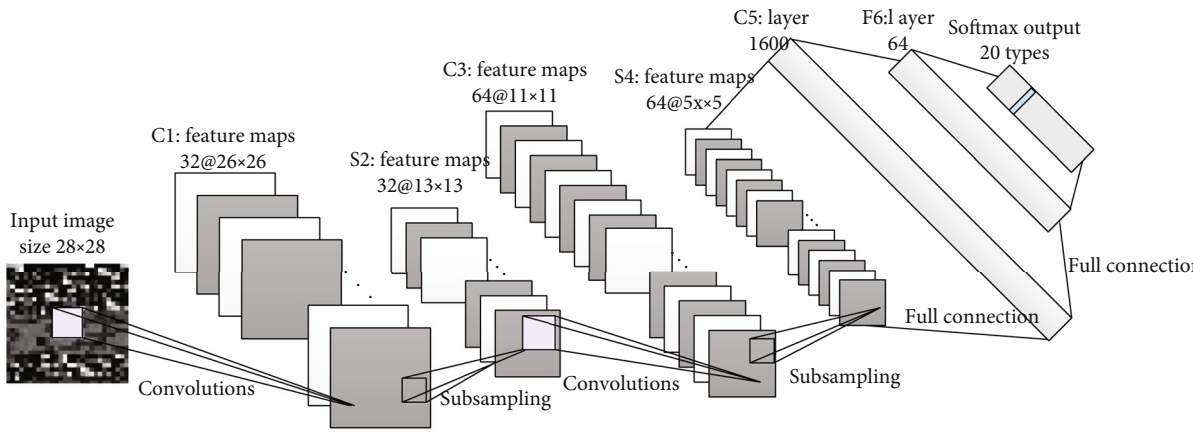

Figure 5: Network structure and parameters of LeNet-5 CNN.

3. Security Models

3.1. Attack Model. This paper assumes that attackers can observe the flow-level network traffic between host nodes and extract features such as packet size and internal packet arrive time. By training on classification models with these network traffic, attackers could infer the application types to thus conduct classification. From [27, 28], we build an attack model, which is specifically described as follows: attackers attempt to classify the network traffic TF being observed into type i of application set C, in which:$C = \{c_1, c_2, c_3 \cdots c_i, \cdots c_n\}$. The feature set of network traffic TF is X, $X = \{x_1, x_2, x_3 \cdots x_m\}$. $F(x)$ is the classification function of the model, and the output value is the probability of type i in C. Related notations are shown in Table 2.

Table 5: Network parameters of LeNet-5 CNN.

Name		Parameters
Input layer		28×28
C1 convolution layer	Convolution core	$32 \times (3 \times 3)$
	Output	$32 \times (26 \times 26)$
S2 pooling layer	Sampling window	2×2
	Output	$32 \times (13 \times 13)$
C3 convolution layer	Convolution core	$64 \times (3 \times 3)$
	Output	$64 \times (11 \times 11)$
S4 max pooling layer	Sampling window	2×2
	Output	$64 \times (5 \times 5)$
Full connection layer		1600×1
Full connection layer		64×1
Output layer		20×1

3.2. Defense Model. Defenders, according to network traffic TF, generate network traffic adversarial samples TA by adding perturbation P. This paper will generate different adversarial samples of network traffic TA by different methods of crafting perturbation, from which the feature set $X' = \{x'_1, x'_2, x'_3 \cdots x'_m\}$ is extracted, which will make the output of the attackers' classification function $F(x')$ different from the original output $F(x)$. That is to say, the attackers will misclassify the traffic into the type i'th rather than type i th. Related notations are shown in Table 3.

3.3. Methods of Generating Perturbation. Ref. [29] summarizes the perturbation crafting into full-pixel perturbation and partial-pixel perturbation, on the basis of which there are three secondary types including target/nontarget, black box/-white box, and visible/invisible. In collaboration with characteristics of network traffic classification, the methods of crafting perturbation introduced in this paper are just like those of the full-pixel perturbation in image classification; that is, adversarial samples are crafted under the context that the parameters and internal structure of the classifier used (such as LeNet-5) by attackers are known. These adversarial samples are required to lead the attackers' classifier to misclassify into not only target label but also nontarget label. Based on the abovementioned, the four perturbation crafting methods introduced in this paper are as follows:

(1) L-BFGS

L-BFGS is introduced by Szegedy [2] when he proposed the concept of adversarial samples. L-BFGS generates adversarial samples based on optimization, and is described as follows:

$$\min c \times \left\| x - x' \right\|_2 + \text{loss}_{F,t}\left(x'\right), s.t. x' \in [0,1]^n. \quad (1)$$

(2) FGSM

Table 6: Network structure and parameters of Vgg-16 CNN.

Name		Parameters
Input layer		28×28
Convolution layer	Convolution core	$32 \times (3 \times 3)$
	Output	$32 \times (26 \times 26)$
Batch normalization layer	Output	$32 \times (26 \times 26)$
Convolution layer	Convolution core	$32 \times (3 \times 3)$
	Output	$32 \times (24 \times 24)$
Batch normalization layer	Output	$32 \times (24 \times 24)$
Max pooling layer	Sampling window	2×2
	Output	$32 \times (12 \times 12)$
Convolution layer	Convolution core	$64 \times (3 \times 3)$
	Output	$64 \times (10 \times 10)$
Batch normalization layer	Output	$64 \times (10 \times 10)$
Convolution layer	Convolution core	$64 \times (3 \times 3)$
	Output	$64 \times (8 \times 8)$
Batch normalization layer	Output	$64 \times (8 \times 8)$
Max pooling layer	Sampling window	2×2
	Output	$64 \times (5 \times 5)$
Convolution layer	Convolution core	$128 \times (3 \times 3)$
	Output	$128 \times (3 \times 3)$
Batch normalization layer	Output	$128 \times (3 \times 3)$
Flatten layer	Output	1152×1
Full connection layer dense 1	Output	64×1
Full connection layer dense 2	Output	20×1

As one of the basic methods in crafting adversarial samples, FGSM, proposed by Goodfellow et al. [19], induces a network to misclassify the image generated by adding increments into the direction of a gradient based on the principle of gradient descent. FGSM calculates perturbation by using the following:

$$P = \varepsilon \text{ sign } (\nabla \mathfrak{F}(\theta, \mathrm{x}, \mathrm{y})). \quad (2)$$

(3) JSMA

JSMA is a typical white box and targeted attack algorithm constrained by l_0 norm proposed by Papernot et al. in 2016, which is aimed at computing a direct mapping from the input to the output to achieve an explicit adversarial goal. JSMA algorithm mainly includes three processes: calculating forward derivative of a deep neural network, calculating adversarial saliency maps, and modifying samples by adding perturbation [30].

(4) C&W Method

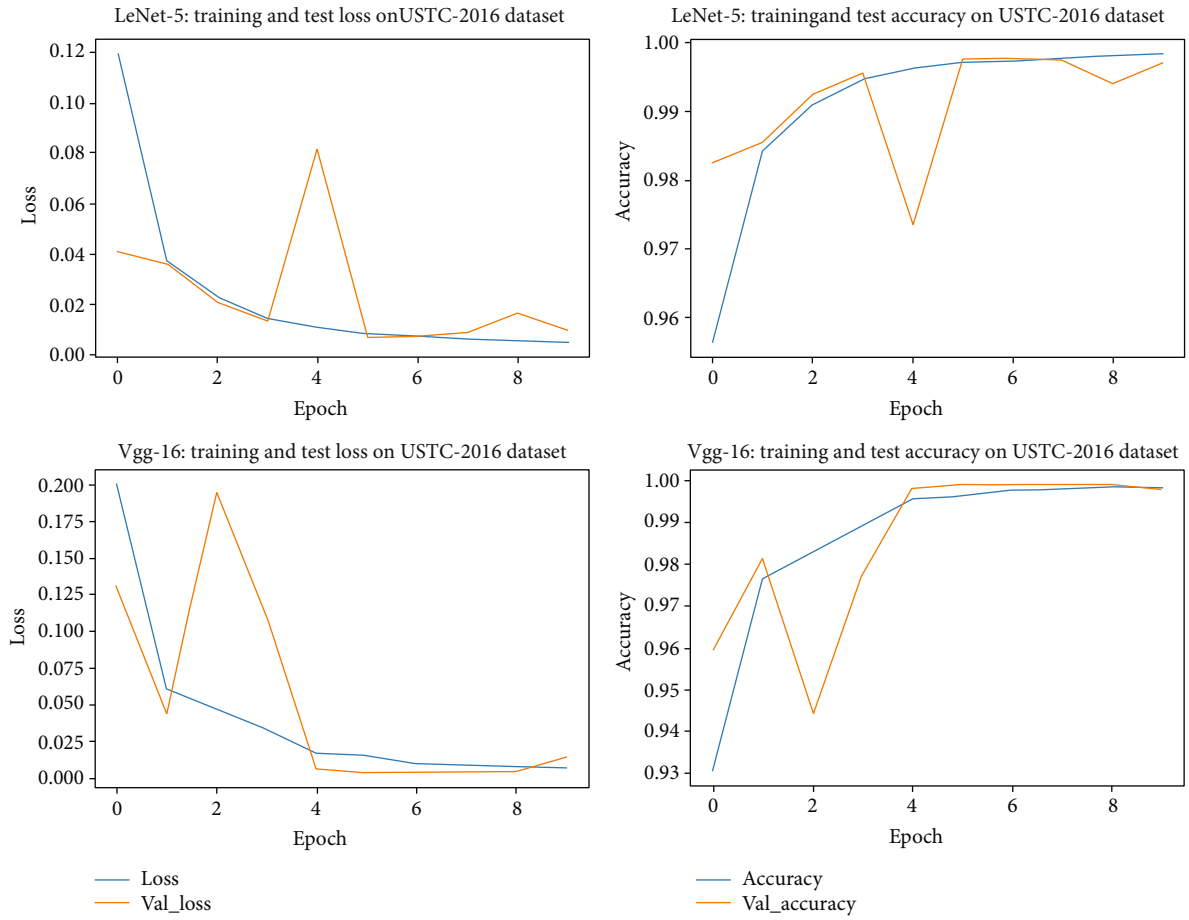

Figure 6: Classification train and test on the UST-TFC2016 dataset.

C&W is proposed by Carlini and Wagner [31] based on FGSM, L-BFGS, and JSMA, which improves greatly in norm l_0, l_2, l_∞. The method with norm l_2 as an example is shown in equation (3). C&W can produce strong adversarial samples, enhance its adversarial transferability, and achieve the ability of black box attacks.

where

$$\min \left\| \frac{1}{2}\left((\tanh(w)+1)-x\right)\right\|_2 + c\cdot f\left(\frac{1}{2}(\tanh(w)+1)\right),$$
$$f\left(x'\right) = \max\left(\max\left\{Z\left(x'\right)_i : i \neq t\right\} - Z\left(x'\right)_i - k\right). \tag{3}$$

3.4. Adversarial Samples of Network Traffic Crafting Algorithm. Based on the abovementioned analysis and perturbation crafting algorithm, this paper has designed the adversarial samples of a network traffic crafting algorithm. The details are as follows:

In Algorithm 1, the real traffic needs to be preprocessed and normalized first. And then, each characteristic value of multiple types of characteristics is reshaped as a grey value in 0-255 and the network traffic is visualized. Next, the CNN model selected by attackers is constructed and trained. The function *Training* (*TF*, *mode*) enables it to classify the traffic data visualized and test the accuracy of classification. At the same time, different methods of crafting perturbation are used to generate perturbation, which will be overlayed with original traffic to be adversarial samples of network traffic *TA*. Finally, by comparing *TA* and *TF*, adversarial samples of traffic network will be evaluated.

4. Experiments

This paper constructs an attack and defense scenario shown in Figure 2, in which attackers are assumed to be able to observe the flow-level network traffic of different applications between host nodes and use the classification models based on deep learning for further attacks. Defenders use the adversarial samples of network traffic crafting method proposed in this paper to add a different perturbation to lead attackers to misclassify during network traffic classification and thus achieve the purpose of defense.

Environment and parameters required by the experiment are shown in Table 4.

4.1. Dataset. The USTC-TFC2016 dataset [11] used in this paper is as the flow traffic observed by attackers which is commonly used by network traffic classification. This dataset includes ten types of malware traffic captured from the real

Table 7: Classification test on the UST-TFC2016 dataset.

Application type	Accuracy		Precision		F1_score	
	LeNet-5	Vgg-16	LeNet-5	Vgg-16	LeNet-5	Vgg-16
BitTorrent	1.00	1.00	1.00	1.00	1.00	1.00
Cridex	1.00	1.00	1.00	1.00	1.00	1.00
Facetime	1.00	1.00	1.00	1.00	1.00	1.00
FTP	1.00	1.00	1.00	1.00	1.00	1.00
Geodo	1.00	1.00	1.00	1.00	1.00	1.00
Gmail	0.99	0.99	0.99	1.00	0.99	0.99
Htbot	1.00	1.00	1.00	1.00	1.00	1.00
Miuref	1.00	1.00	1.00	1.00	1.00	1.00
MySQL	1.00	1.00	1.00	1.00	1.00	1.00
Neris	0.99	1.00	0.99	1.00	0.99	1.00
Nsis-ay	0.99	1.00	1.00	0.99	0.99	1.00
Outlook	0.98	1.00	1.00	0.99	0.99	0.99
Shifu	1.00	1.00	1.00	0.99	1.00	1.00
Skype	1.00	0.98	1.00	1.00	1.00	0.99
SMB	1.00	1.00	1.00	1.00	1.00	1.00
Tinba	1.00	1.00	1.00	1.00	1.00	1.00
Virut	0.99	1.00	0.99	1.00	0.99	1.00
Weibo	1.00	1.00	1.00	1.00	1.00	1.00
WOW	1.00	1.00	1.00	0.99	1.00	1.00
Zeus	1.00	1.00	1.00	1.00	1.00	1.00

network environment by CTU researchers from 2011 to 2015 and ten types of normal traffic data simulated by professional tools. To reflect more kinds of traffic as possible, ten kinds of traffic contain eight classes of common applications. The size of the USTC-TFC2016 dataset is 3.71 GB, and the format is pcap.

4.2. Data Preprocessing. In this part, with the toolkit USTC-TK2016, raw traffic data (pcap format) is converted to CNN's input data (idx format). The whole process includes traffic split, traffic clear, image generation, and IDX conversion [11]. After preprocessing, 20 types of different applications of network traffic are formed, including 10 types of 243761 normal traffic flows and 10 types of 179252 malicious traffic flows, in which 90% (379812 flows) are used as training dataset and 10% (42201 flows) as test dataset. The statistical chart of dataset distribution is showed in Figure 3.

Each of the 20 types of network traffic can be visualized to grey image with 784 ($28 * 28$) bytes. The visualization results are shown in Figure 4. In Figure 4, the left group shows the visualization result of all 20 types of traffic and the right group shows the consistency in the same traffic type. It is obvious that these images visualized from network traffic have obvious discrimination degree, and each type of traffic has high consistency.

4.3. Attacker Classification Model. It is assumed that the attacker uses LeNet-5 CNN as his classification model, which is widely used in classification of network traffic applications [33]. Ref. [34] has improved the LeNet-5 CNN model with its network structure, and the network structure and parameters of LeNet-5 CNN are shown in Figure 5 and parameters in Table 5.

To validate the transferability of the adversarial samples of network traffic generated, the Vgg-16 CNN model is selected as the classification model to test adversarial samples crafted for LeNet-5. The parameters of the network structure of Vgg-16 are shown in Table 6.

4.4. Classification Test. Without the defense of adversarial samples of network traffic, the effect of classification of LeNet-5 and Vgg-16 used by the attacker is perfect. The classification test is shown in Figure 6 and Table 7 with three evaluation metrics: accuracy, precision, and F1 score.

4.5. Perturbation Crafting. With Algorithm 1 and four methods of perturbation crafting, adversarial samples of network traffic *TD* of the defender model are crafted for LeNet-5 CNN. Taking with Geodo type as the example, perturbations crafted by different methods are shown in Figure 7. In Figure 7, the column "Perturbation" shows the difference of perturbations generated by four methods, in which the brightness of the perturbation pixel corresponds to the value of the heat map. The value "1" and value "-1" of the heat map mean the strongest "positive" and "negative" perturbations after standardization. For example, perturbations generated by JASM are stronger than the perturbations generated by C&W.

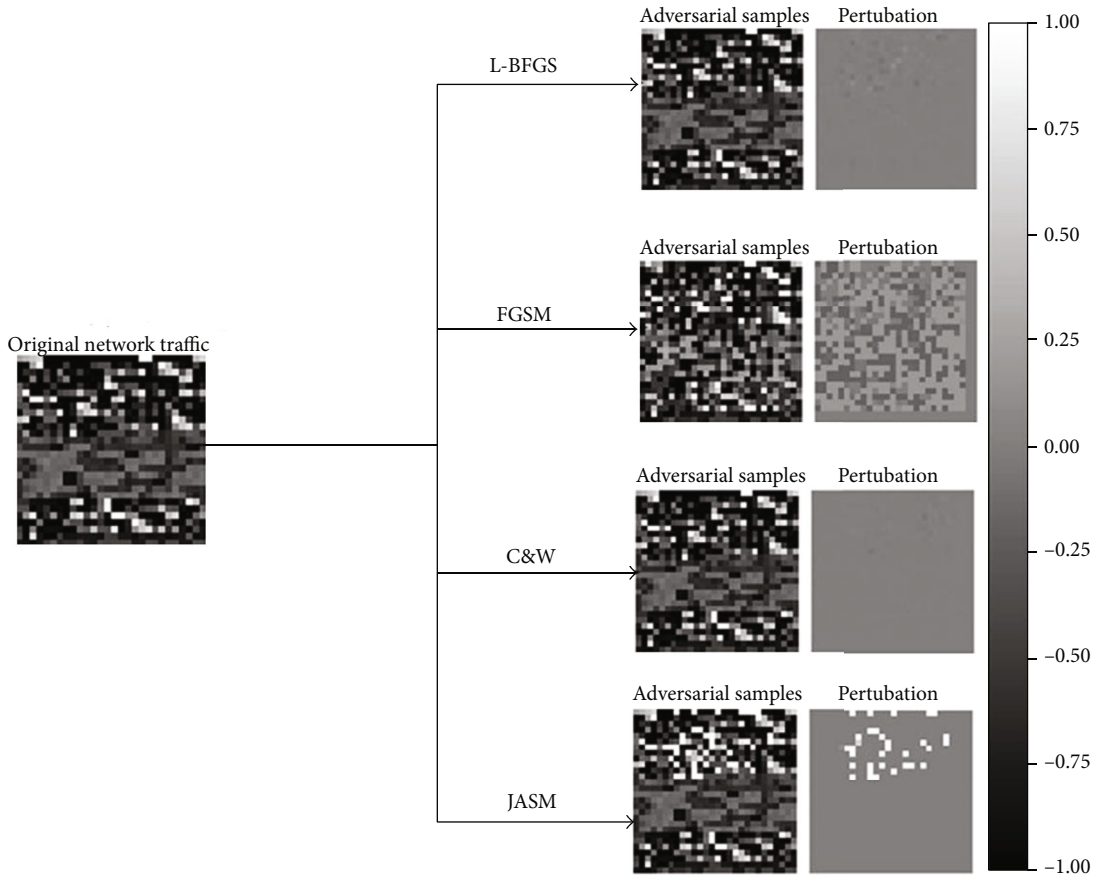

Figure 7: Comparison of adversarial samples of network traffic crafted by different methods.

4.6. Comparison and Analysis. The comparison of the experiment consists of two parts: (1) The defender carries out untargeted defense to the attacker, which means the purpose of defender is make the attacker misclassify the application class to another but no particular class. For example, the attacker misclassifies the Outlook network traffic to any other class such as Gmail and FTP. (2) The defender carries out targeted defense to the attacker, which means the purpose of the defender is to make the attacker misclassify the application class to a particular class. For example, the attacker misclassifies the Outlook network traffic to MySQL. In each part, after the test of the effect of the adversarial samples for LetNet-5 CNN, the transferability is validated, which means the defender uses adversarial samples generated for LeNet-5 CNN to deceive Vgg-16 CNN. To evaluate the effect of deceiving in untargeted defense, deceiving rate (DR) is used as shown in equation (4). And in targeted defense, matching rate (MR) [35] is defined as that which describes the percentage of the adversarial examples generated for the source model that is misclassified as the target label by the target model.

$$\mathrm{DR} = 1 - \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i}. \tag{4}$$

To evaluate the quality of adversarial samples generated, l_0 norm, l_2 norm, and Structural Similarity Index (SSIM) are used. The comparison of the experiment is shown in Table 8 and Table 9:

From the comparison in Tables 8 and 9, we can validate the effect of adversarial samples of network traffic generated by four different methods. In the untargeted defense group, the adversarial samples crafted by C&W perform best on deceiving LeNet-5 CNN with low change to original but with disadvantages of slow to crafting perturbation and low transferability to other CNN models, which could be used in an application field that could provide high computation ability and demand for high deceiving rate. FGSM could craft perturbation quickly and transfer the deception to other CNNs. However, the change is to the original image of perturbation by FGSM which is much more than other methods. In the targeted defense group, C&W is also the best to perform the ability of deceiving LeNet-5 but has no effect of transferability to Vgg-16, and neither are other methods. Contrary to L-BFGS performing better in this part than in the untargeted part, FGSM performs worse in contrast to the performance in the untargeted part. About the transferability of the adversarial samples of network traffic, only JASM performs a little bit of transferability.

Table 8: Untargeted defense on LeNet-5 and transferability on Vgg-16.

Methods of crafting perturbation	Traffic application class	Deceiving rate on LeNet-5	L2 norm	L0 norm	SSIM	Time consuming (second)	Transferability deceiving rate on Vgg-16
L-BFGS	Geodo	6.40%	8.31%	73.28%	74.86%	13	31.25%
	Neris	53.60%	2.58%	65.20%	91.04%	13	62.31%
	Virut	46.00%	2.71%	68.96%	93.72%	13	60.87%
	Cridex	0.20%	5.00%	81.00%	86.53%	15	100.00%
	Average	26.55%	4.65%	72.11%	86.54%	14	63.61%
FGSM	Geodo	100.00%	10.83%	51.24%	38.28%	2	38.28%
	Neris	90.20%	8.79%	60.45%	54.64%	2	98.00%
	Virut	95.40%	7.74%	70.86%	77.97%	2	58.91%
	Cridex	90.80%	8.26%	76.53%	76.33%	2	44.05%
	Average	94.10%	8.91%	64.77%	61.81%	2	59.81%
C&W	Geodo	100.00%	1.00%	50.00%	99.59%	151	0.00%
	Neris	100.00%	1.00%	42.00%	99.99%	89	0.40%
	Virut	100.00%	1.00%	58.00%	99.98%	110	1.20%
	Cridex	100.00%	1.00%	75.00%	99.88%	134	0.20%
	Average	100.00%	1.00%	56.25%	99.86%	121	0.45%
JASM	Geodo	99.80%	11.38%	4.52%	63.30%	135	61.12%
	Neris	96.40%	10.12%	5.36%	65.20%	137	62.24%
	Virut	86.20%	8.91%	5.86%	71.04%	135	36.19%
	Cridex	99.80%	7.89%	5.26%	72.98%	136	31.66%
	Average	95.55%	9.56%	5.25%	68.10%	136	47.80%

Table 9: Targeted defense (MySql as the targeted class) on LeNet-5 and transferability on Vgg-16.

Methods of crafting perturbation	Traffic application class	Matching rate on LeNet-5	L2 norm	L0 norm	SSIM	Time consuming (second)	Transferability deceiving rate on Vgg-16
L-BFGS	Geodo	100.00%	1.07%	66.19%	98.65%	68	0.00%
	Neris	100.00%	1.24%	72.76%	97.71%	69	0.00%
	Virut	94.60%	1.15%	75.26%	99.21%	68	0.00%
	Cridex	100.00%	1.11%	78.59%	99.66%	66	0.00%
	Average	98.65%	1.14%	73.20%	98.80%	68	0.00%
FGSM	Geodo	10.60%	10.54%	46.00%	36.26%	2	0.00%
	Neris	0.20%	8.00%	72.00%	78.37%	2	0.00%
	Virut	1.80%	8.33%	67.22%	76.67%	2	0.00%
	Cridex	2.20%	8.27%	76.27%	80.75%	2	0.00%
	Average	3.70%	8.79%	67.37%	68.01%	2	0.00%
C&W	Geodo	100.00%	1.00%	51.00%	99.71%	174	0.00%
	Neris	100.00%	1.00%	55.00%	99.86%	135	0.00%
	Virut	100.00%	1.00%	66.00%	99.89%	139	0.00%
	Cridex	100.00%	1.00%	76.00%	99.73%	167	0.00%
	Average	100.00%	1.00%	62.00%	99.80%	154	0.00%
JASM	Geodo	93.60%	7.43%	2.62%	75.24%	136	0.00%
	Neris	61.40%	11.10%	5.24%	59.74%	135	0.33%
	Virut	28.80%	10.27%	6.06%	64.65%	135	0.69%
	Cridex	69.80%	8.62%	5.65%	70.47%	135	0.00%
	Average	63.40%	9.36%	4.87%	67.53%	135	0.26%

5. Conclusion and Further Work

This paper first describes the current research status in the area of network traffic classification. Then, from the perspective of defenders and based on researches related, it introduces the concept of adversarial samples to network traffic and raises a novel way to generate adversarial samples of network traffic. After the models of attack and defense are described, experiments are conducted with four methods of crafting perturbation. In the experiments, LeNet-5 CNN is considered as the classification model used by the attacker to be deceived. By adding perturbation generated by different methods to grey images transformed from normal network traffic, the adversarial samples of network traffic are formed, respectively, to confuse the target model. The experiments not only compared the effect of adversarial samples generated on LeNet-5 CNN but also validated the transferability of adversarial samples of network traffic on Vgg-16 CNN.

There are three limitations and related future work about this work. First, the main goal of this paper is to show the effect of adversarial samples of network traffic, so only the basic methods of crafting perturbation are used and compared. The effect of other methods should also be considered. Secondly, it is assumed that the classification model used by the attacker in the experiment is LeNet-5. However, in the real attack and defense, other CNNs may be selected as the classification model too. So, the effect on other CNNs will be validated next. Lastly, our work in this paper only performs the transformation from the network traffic to grey images, but how to change the image to network traffic and how to keep the integrity of the original network traffic during transforming need to be studied carefully in further work.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by the Foundation of Science and Technology on Information Assurance Laboratory (No. KJ-15-108).

References

[1] A. Back, U. Möller, and A. Stiglic, "Traffic analysis attacks and trade-offs in anonymity providing systems," in *Information Hiding*, I. S. Moskowitz, Ed., vol. 2137 of Lecture Notes in Computer Science, , Springer, 2001.

[2] C. Szegedy, W. Zaremba, I. Sutskever et al., "Intriguing properties of neural networks," 2013, http://arxiv.org/abs/1312.6199.

[3] C. Sitawarin, A. N. Bhagoji, A. Mosenia, P. Mittal, and M. Chiang, "Rogue signs: deceiving traffic sign recognition with malicious ads and logos," 2018, http://arxiv.org/1801.02780.

[4] K. Yang, J. Liu, C. Zhang, and Y. Fang, "Adversarial examples against the deep learning based network intrusion detection system," in *MILCOM 2018—2018 IEEE Military Communications Conference (MILCOM)*, pp. 559–564, Los Angeles, CA, USA, 2018.

[5] W. W. Hu and Y. Tan, "Generating adversarial malware examples for black-box attacks based on GAN," 2017, http://arxiv.org/1702.05983.

[6] G. Xiong, J. Meng, Z. Cao, Y. Wang, L. Guo, and B. X. Fang, "Research progress and prospects of network traffic classification," *Journal of Integration Technology*, vol. 1, no. 1, pp. 32–42, 2012.

[7] A. Dainotti, A. Pescape, and K. Claffy, "Issues and future directions in traffic classification," *IEEE Network: The Magazine of Computer Communications*, vol. 26, no. 1, pp. 35–40, 2012.

[8] Z. Wang, "The applications of deep learning on traffic identification," https://goo.gl/WouIM6.

[9] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 43–48, Beijing, China, 2017.

[10] V. Rimmer, D. Preuveneers, M. Juarez, T. Van Goethem, and W. Joosen, "Automated website fingerprinting through deep learning," 2017, http://arxiv.org/1708.06376.

[11] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *2017 International Conference on Information Networking (ICOIN)*, pp. 712–717, Da Nang, Vietnam, 2017.

[12] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website fingerprinting in onion routing based anonymization networks," in *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society*, pp. 103–114, New York: ACM Press, 2011.

[13] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton, "Peek-a-boo, I still see you: why efficient traffic analysis countermeasures fail," in *2012 IEEE Symposium on Security and Privacy*, pp. 332–346, San Francisco, CA, USA, 2012.

[14] W. Cui, J. Yu, Y. Gong, and E. Chan-Tin, "Realistic cover traffic to mitigate website fingerprinting attacks," in *2018 IEEE 38th International Conference on Distributed Computing Systems*, pp. 1579–1584, Vienna, Austria, 2018.

[15] T. Wang and I. Goldberg, "Walkie-talkie: an efficient defense against passive website fingerprinting attacks," in *Proceedings of the 26th USENIX Security Symposium*, pp. 1375–1390, Vancouver, BC, 2017.

[16] R. Dingledine, *Obfsproxy: The Next Step in the Censorship Arms Race*, TOR Project official blog, 2012, https://www.torproject.org/projects/ob-fsproxy.

[17] H. M. Moghaddam, B. Li, M. Derakhshani, and I. Goldberg, "SkypeMorph: Protocol Obfuscation for TOR Bridges," in *Proceedings of the 2012 ACM conference on Computer and communications security — CCS '12*, pp. 97–108, Raleigh, NC, USA, 2012.

[18] F. Li, A. M. Kakhki, D. Choffnes, P. Gill, and A. Mislove, "Classifiers unclassified: an efficient approach to revealing IP traffic classification rules," in *Proceedings of the 2016 Internet Measurement Conference*, pp. 239–245, New York, 2016.

[19] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, http://arxiv.org/abs/1412.6572.

[20] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2016, http://arxiv.org/abs/1607.02533.

[21] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press*, pp. 1765–1773, Honolulu, Hawaii,USA, 2017.

[22] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," 2017, http://arxiv.org/abs/1707.07397.

[23] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2574–2582, Las Vegas, Nevada,USA, 2016.

[24] J. H. Metzen, M. C. Kumar, T. Brox, and V. Fischer, "Universal adversarial perturbations against semantic image segmentation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2755–2764, Venice, Italy, 2017.

[25] K. R. Mopuri, U. Garg, and R. V. Bahu, "Fast feature fool: a data independent approach to universal adversarial perturbations," http://arxiv.org/abs/1707.05572,2017.

[26] K. R. Mopuri, A. Ganeshan, and R. V. Babu, "Generalizable data-free objective for crafting universal adversarial perturbations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 10, pp. 2452–2465, 2019.

[27] G. Verma, E. Ciftcioglu, R. Sheatsley, K. Chan, and L. Scott, "Network traffic obfuscation: an adversarial machine learning approach," in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, pp. 1–6, Los Angeles, CA, USA, 2018.

[28] J. B. Xiong, J. Ren, L. Chen et al., "Enhancing privacy and availability for data clustering in intelligent electrical service of IoT," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1530–1540, 2019.

[29] W. W. Pan, X. Y. Wang, M. L. Song, and C. Chen, "Survey on generating adversarial examples," *Journal of Software*, vol. 31, no. 1, pp. 67–81, 2020.

[30] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 372–387, Saarbruecken, Germany, 2015.

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

[32] N. Papernot, I. Goodfellow, R. Sheatsley, R. Feinman, and P. McDaniel, "Cleverhans v1.0.0: an adversarial machine learning library," 2018, http://arxiv.org/abs/1610.00768.

[33] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[34] W. A. Yong, Z. Huiyi, F. E. Hao, Y. E. Miao, and K. E. Wenlong, "Network traffic classification method basing on CNN," *Journal on Communications*, vol. 39, no. 1, pp. 14–23, 2018.

[35] D. Su, H. Zhang, H. Chen, J. Yi, P. Y. Chen, and Y. Gao, "Is robustness the cost of accuracy? — A comprehensive study on the robustness of 18 deep image classification models," 2018, http://arxiv.org/abs/1808.01688.