Hindawi
Security and Communication Networks
Volume 2021, Article ID 5395705, 10 pages
https://doi.org/10.1155/2021/5395705

Research Article

FNet: A Two-Stream Model for Detecting Adversarial Attacks against 5G-Based Deep Learning Services

Guangquan Xu,[1,2] Guofeng Feng,[1] Litao Jiao,[2] Meiqi Feng,[1] Xi Zheng,[3] and Jian Liu[1]

[1]*College of Intelligence and Computing, Tianjin University, Tianjin 300350, China*
[2]*School of Big Data, Qingdao Huanghai University, Qingdao 266555, China*
[3]*Department of Computing, Macquarie University, North Ryde 2113, Australia*

Correspondence should be addressed to Jian Liu; jianliu@tju.edu.cn

Received 26 June 2021; Accepted 17 August 2021; Published 7 September 2021

Academic Editor: Benjamin Aziz

With the extensive application of artificial intelligence technology in 5G and Beyond Fifth Generation (B5G) networks, it has become a common trend for artificial intelligence to integrate into modern communication networks. Deep learning is a subset of machine learning and has recently led to significant improvements in many fields. In particular, many 5G-based services use deep learning technology to provide better services. Although deep learning is powerful, it is still vulnerable when faced with 5G-based deep learning services. Because of the nonlinearity of deep learning algorithms, slight perturbation input by the attacker will result in big changes in the output. Although many researchers have proposed methods against adversarial attacks, these methods are not always effective against powerful attacks such as CW. In this paper, we propose a new two-stream network which includes RGB stream and spatial rich model (SRM) noise stream to discover the difference between adversarial examples and clean examples. The RGB stream uses raw data to capture subtle differences in adversarial samples. The SRM noise stream uses the SRM filters to get noise features. We regard the noise features as additional evidence for adversarial detection. Then, we adopt bilinear pooling to fuse the RGB features and the SRM features. Finally, the final features are input into the decision network to decide whether the image is adversarial or not. Experimental results show that our proposed method can accurately detect adversarial examples. Even with powerful attacks, we can still achieve a detection rate of 91.3%. Moreover, our method has good transferability to generalize to other adversaries.

1. Introduction

Deep learning has recently led to significant improvements in many fields, such as computer vision [1–3], speech recognition [4, 5], and natural language processing [6, 7]. With the continuous development of 5G communication and artificial intelligence technology, the two have developed from mutual independence to deep integration. The artificial intelligence promotes the intelligent development of the communication network itself, and the industry widely believes that 5G and artificial intelligence are general-purpose technologies (GPTs) [8]. Many have explored the application of artificial intelligence in 5G communication in the form of investigation or empirical research [9, 10]. Under this trend, communication artificial intelligence has developed rapidly and vigorously. For example, Chinese operators provide services to users and partners by AI center [11]. The AI center is based on artificial intelligence algorithms, encapsulates scene-oriented services, and provides applications and services in customer sales and customer service.

Although deep learning is powerful, there are security risks in deep learning services provided by neural network models. For example, in customer sales services deployed on 5G platforms, the main application of artificial intelligence technology is image classification. This service effectively recognizes the image data through the image classification model. Although the image classification model provides great convenience for users, the inherent fragile of deep learning is a weakness of the service. Recent studies [12, 13]

have shown that an attacker can create adversarial samples by adding small disturbances to the original data. The disturbances are very small, and they are almost invisible to the human eye. If the attacker inputs adversarial samples into the recognition model, the model will not correctly recognize the samples.

The deep learning model deployed on the 5G platform provides intelligent image recognition services. Hackers attack the deep learning model, causing errors in the deployed image recognition service. Specifically, the attacker adds a small disturbance to the original input to cause a huge change in the output of model. Most adversarial attacks currently existing are aimed at image classification. To ensure the security of 5G-based deep learning services, we mainly research the detection method of adversarial samples to protect against image classification models. Although methods [14–17] are proposed to detect adversarial examples, these methods always fail when faced with a powerful attack like the CW. To solve the adversarial attack on the image classification service based on the 5G platform, we mainly research the detection method of adversarial samples. In our method, we regard adversarial disturbances as special noise features that could provide additional evidence for adversarial sample detection. Since adversarial disturbance and image steganography both modify the picture directly, destroying the correlation between the original image pixels, we can apply the method of steganalysis to the field of adversarial sample detection. In fact, Goodfellow et al. also proposed that adversarial attacks can be regarded as accidental steganography [18].

In this paper, we use rich features extracted from the spatial rich model (SRM) [19] to help the deep learning model for detecting adversarial examples. The SRM is a traditional approach to extract noise features in steganalysis [20]. The emergence of steganography promoted the development of steganalysis. Steganography is to add secret information to the original carrier, making the information hidden in the carrier difficult to detect [21]. The steganography of the picture changes the pixel value of the picture, which will destroy the correlation between adjacent pixels of the original image. Therefore, steganalysis can be performed according to this. Steganalysis determines whether the image has steganalysis by modeling the correlation between adjacent pixels of the image. Traditional steganalysis algorithms are based on manual feature extraction. After continuous research by many scholars, SRM has been able to extract 30,000 multidimensional features from the data through improved high-pass filters (HPFs) [22]. The design of these high-pass filters is mostly based on experience. SRM uses 30 different pixel predictors. The pixel predictor is linear or nonlinear. Each linear predictor is a shift-invariant finite-impulse response filter which is described by a kernel matrix K^{pred}. The residual R is a matrix which has the same dimension as Y:

$$R = K^{\text{pred}} * Y - Y \triangleq K * Y, \tag{1}$$

where the symbol $*$ denotes the convolution with Y mirrorpadded. Thus, R has the same dimension as Y.

There are six types of residuals: first-order, second-order, third-order, SQUARE, EDGE 3×3, and EDGE 5×5 [19]. Table 1 shows the calculation methods of first-order, second-order, and third-order linear residuals. For example, one simple linear residual is $R_{ij}^{h} = y_{ij+1} - y_{ij}$, which is the difference between a pair of horizontally adjacent pixels. In this case, the residual kernel is $K = (-1, 1)$, which means that the pixel value is predicted as its horizontally neighboring pixel. We can use this method to extract the noise features of other directions.

SQUARE, EDGE 3×3, and EDGE 5×5 linear residuals use more directional neighborhood pixels in their calculations. Tables 2 and 3 show SQUARE, EDGE 3×3, and EDGE 5×5 SRM filter kernels.

Our model consists of a two-stream network and a decision network. The RGB stream is used to capture subtle differences, such as contrast differences of a RGB image. The SRM noise stream is used to capture the noise inconsistency between clean samples and adversarial samples. We use 30 typical SRM filters to extract noise features from adjacent pixels. The noise features are used as the SRM noise input of the two-stream model. Then, we use bilinear pooling [23] to fuse the features extracted from the two streams. Bilinear pooling used for it can fuse the features of the two streams while preserving spatial information. Finally, we use the fully connected layer as a decision network to detect adversarial samples.

Our contributions are summarized as follows:

(1) To improve the security of 5G-based deep learning services, we propose a new two-stream adversarial example detection model and perform end-to-end training. This method can obtain rich feature information from noise features and provide additional evidence for adversarial example detection. Even with a powerful CW attack, we can still achieve a detection rate of 91.3%.

(2) We choose the spatial rich model (SRM) to generate linear and nonlinear noise features. The 30 SRM filters could amplify the difference in the noise domain and get additional rich information to help detect adversarial samples.

The rest of this paper is organized as follows. Section 2 describes the related work about adversarial sample attacks and against deep learning models. Section 3 introduces the proposed two-stream network. Section 4 analyzes the experimental results. Finally, Section 5 concludes the paper.

2. Related Work

We briefly review the related work in adversarial attack and adversarial defense in this section.

2.1. Security Issues of 5G-Based Deep Learning Services. Due to the inherent vulnerability of neural networks, 5G-based deep learning services face the same threat of adversarial attacks. Deep learning (including deep reinforcement learning) is vulnerable to adversarial examples

Table 1: The SRM linear residual filters, where R_{ij}^{h} denotes the linear residual of pixel at the position (i, j) in horizontal direction and y_{ij} denotes the pixel at the position (i, j).

Residual type	HPF	Linear residual
First-order	$(1, -1)$	$R_{ij}^{h} = y_{ij+1} - y_{ij}$
Second-order	$(1, -2, 1)$	$R_{ij}^{h} = y_{ij-1} - 2y_{ij} + y_{ij+1}$
Third-order	$(1, -3, 3, -1)$	$R_{ij}^{h} = y_{ij-1} - 3y_{ij} + 3y_{ij+1} - y_{ij+2}$

Table 2: The SQUARE SRM filter kernels.

$$\begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix} \begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix}$$

Table 3: The EDGE SRM filter kernels.

$$\begin{bmatrix} 2 & -1 \\ 4 & 2 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} -2 & 2 & -1 \\ 8 & -6 & 2 \\ -12 & 8 & -2 \\ 8 & -6 & 2 \\ -2 & 2 & -1 \end{bmatrix}$$

[24]. These modified inputs can trick the model into making the wrong decision. Hackers may use this vulnerability to attack services based on deep learning [25]. Adversarial attacks can be divided into black-box attacks and white-box attacks. The black-box attack can only obtain the identification result of the inputs through the API interface of the platform but cannot get the internal parameters of the model. The white-box attack can get all the parameters of the neural network model deployed on the platform. Therefore, white-box attacks also represent the highest level of attack.

2.2. Deep Learning Attack. The deep learning algorithm is a weakness of deep learning services based on 5G. Because of the inherent vulnerability of neural networks [26], a small adversarial disturbance can cause a huge change in the output of the model. In addition, scholars have proposed attacks on the model optimization process and regularization process [27, 28].

Goodfellow et al. [18] proposed fast gradient notation (FGSM) to generate adversarial examples. The main idea of FGSM is to make the disturbance direction consistent with the gradient direction to maximize the change in the loss function value and then maximize the change in the classification result of the classifier.

Moosavi-Dezfooli et al. [28] proposed DeepFool to find the shortest distance from the clean images to the decision boundary of the adversarial images. DeepFool is a nontargeted attack method to generate an adversarial example by iteratively perturbing an image. Experiments show that the DeepFool method produces less interference than FGSM and has similar fool rates.

Carlini and Wagner [27] proposed the CW method to generate adversarial examples. CW is an adversarial example attack algorithm based on objective function optimization. It restricts the L_{∞}, L_2, and L_0 norms to make the disturbance undetectable. Experiments show that defensive distillation is completely unable to defend against these three kinds of attacks. The CW method is a strong attack which is difficult to defend.

2.3. Deep Learning Defense. Recent research [29] has shown that adversarial examples not only mislead classifiers in electronic data but also have the same effect in the physical world. In view of the great harm of adversarial examples, many scholars have studied the defense of adversarial examples.

The adversarial training proposed by GoodFellow et al. [18] is an earlier measure to defend against samples. The main method is to train the adversarial samples and the normal samples simultaneously in the training stage to enhance the robustness of the model. However, confrontational training requires a large number of adversarial examples to ensure a high detection rate, which will lead to huge training costs.

Hinton et al. [15] first introduced the distillation defense method for small models to imitate large models. Later, Papernot et al. [30] adopted the distillation method to defend adversarial samples. This method makes the decision boundary of the model smoother and can effectively defend against the adversarial samples generated by FGSM, BIM, and other algorithms. Compared with adversarial training, its training cost is lower. However, this method is not effective against CW attacks.

Dziugaite et al. [16] studied the defensive effect of JPG feature compression against FGSM attacks. The limitation of this defense method is that large-scale compression which will lead to the decline of the accuracy of legal sample classification. For the sample with large interference, it is not enough to eliminate interference.

Due to the difficulty of reclassification of antagonistic samples, many researchers turn to the detection of adversarial samples.

Liang et al. [17] proposed an adversarial sample detection method based on adaptive denoising. In this method, antagonistic interference is regarded as artificial noise, and noise reduction technology is used to reduce its adversarial effect. Liu et al. [31] proposed to apply steganalysis to detect adversarial examples. This work found a relevant connection between computer vision and adversarial examples of steganalysis. Feinmanet et al. [32] proposed a method to detect adversarial examples using the Bayesian uncertainty of the network and the method of kernel density estimation.

2.4. Proposed Method. We designed a two-stream network to detect adversarial examples. As shown in Figure 1, the RGB stream directly uses the RGB image for input and the SRM stream uses the noise features extracted by the SRM filter as the input. Then, we use bilinear pooling to fuse

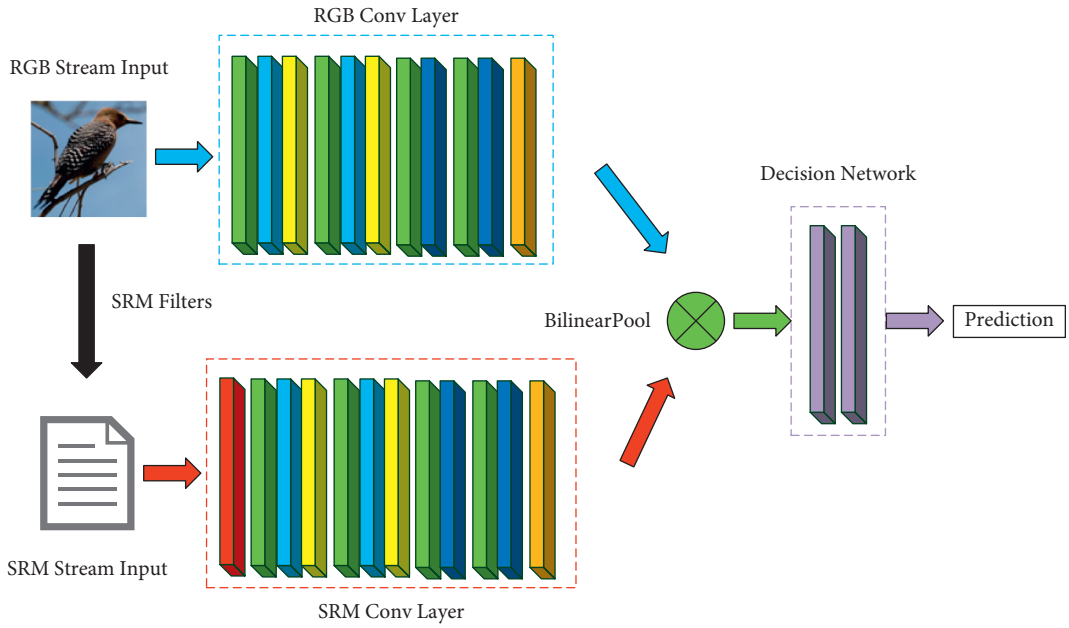

Figure 1: Illustration of our two-stream network (FNet). Color code used is as follows: light green = Conv, light blue = batchnorm + tanh, deep blue = batchnorm + ReLU, yellow = avg pooling, orange = max pooling, and purple = fully connected layers. The RGB stream uses original images as input and captures subtle difference like contrast difference and unnatural pixels from the RGB features. The noise stream first obtains noise feature maps through SRM filter layer and leverages the noise features to provide additional evidence for adversarial image detection. Bilinear pooling combines the features extracted by the two streams. Finally, passing the combined features through a decision network, the network generates the predicted label and determines whether the input image is adversarial or not.

the features extracted by the two CNNs. Finally, we input the final features into the decision network for classification. In addition, our trained model maintains a good detection rate against the adversarial examples generated by black-box models.

2.5. SRM Feature Extraction. Both adversarial attacks and steganography on images make perturbations on the pixel values, which alter the dependence between pixels. However, steganalysis can effectively detect modifications caused by steganography via modeling the dependence between adjacent pixels in natural images. So, we can also take advantage of steganalysis to identify deviations due to adversarial attacks. Inspired by the work of Liu [31] and Zhou [33], we decided to use the SRM filter to amplify the local noise disturbance of the image and use it as additional evidence to assist the decision-making network.

Although the work of Liu et al. [31] has used the SRM filter for adversarial sample identification and proved the effectiveness of the method, these works only consider the linear SRM noise features. In our work, we simulated the manual extraction method of SRM linear residuals. The linear residuals are obtained by convolving the image with a high-pass filter with a shift-invariant kernel. Specifically, we used 30 basic SRM filters and then convolved this convolution kernel with the original image to gather the basic noise features. The definition of residual filters is shown in Tables 1–3. Further, we divided the filters into five categories: first-order, second-order, third-order, SQUARE 3×3 +EDGE 3×3, and SQUARE 5×5 +EDGE 5×5. The number of filters for each category is 8, 4, 8, 5, and 5, according to the different directions of pixel feature extraction. Specifically, for the first- and third-order, we used 8 filters to extract pixel features in eight directions $\{\uparrow, \downarrow, \leftarrow, \longrightarrow, \nearrow, \swarrow, \nwarrow, \searrow\}$; for the second-order, EDGE 3×3, and EDGE 5×5, 4 filters were used to extract pixel features in four directions $\{\uparrow, \downarrow, \leftarrow, \longrightarrow\}$; for SQUARE 5×5 and SQUARE 5×5, we used 1 filter to extract pixel features. Based on the point view of more comprehensive characteristics of SRM's nonlinear residual statistical characteristics, we used the linear residual features obtained by SRM filters in the spatial domain to obtain nonlinear residual features by nonlinear processing.

We take the features extracted by the second-order SRM filter as an example to introduce how to obtain nonlinear residual features. First, we can predict pixel $Y_{i,j}$ from its horizontal or vertical neighboring pixels, thus obtaining 2 horizontal and 2 vertical residual. Then, we get 4 direction residuals: $R_{ij}^{\longrightarrow}$, $R_{ij}^{\leftarrow}$, $R_{ij}^{\uparrow}$, and $R_{ij}^{\downarrow}$. Secondly, we can use these 4 residuals to compute 2 nonlinear "minmax" residuals as follows:

$$R_{ij}^{(\max)} = \max\left(R_{ij}^{\rightarrow}, R_{ij}^{\leftarrow}, R_{ij}^{\uparrow}, R_{ij}^{\downarrow}\right),$$
$$R_{ij}^{(\min)} = \min\left(R_{ij}^{\rightarrow}, R_{ij}^{\leftarrow}, R_{ij}^{\uparrow}, R_{ij}^{\downarrow}\right). \tag{2}$$

The other four types of nonlinear residual features are calculated in the same way. In this way, the nonlinear residual features of the 10 channels are obtained. Finally, we get the linear features of 30 channels and the nonlinear

features of 10 channels. Then, we combine the nonlinear and linear features to get noise features of 40-channel as the input of the noise stream.

2.6. Two-Stream Network. We adopt a two-stream network with RGB stream and spatial rich model (SRM) noise stream to detect adversarial images. The model structure designed is shown in Figure 1. The role of the RGB input in our network is to catch significant disturbances. However, for tampered images that have been carefully processed to hide stitching boundaries and reduce contrast differences, it will be difficult to accomplish the task using RGB streams alone. In addition, when generating adversarial samples, we often use the L_p norm to restrict adversarial disturbances. Therefore, the generated adversarial sample disturbances are often invisible to the human eye. In this case, it is difficult to complete the detection of antagonistic samples only by relying on the RGB stream. Inspired by the work of Liu [31] and Zhou [33], we adopt SRM noise features as additional evidence to determine whether the input image is adversarial or not.

In our model, we used RGB stream to simulate visual tampering and detect image disturbance with large disturbance; SRM noise stream is used to extract and amplify noise features by SRM filters, which is used as the additional evidence for adversarial image detection (see Table 4 for the specific network structure). In the structure of the SRM noise stream network, we add an ABS layer to reduce the influence of symbols on the model decision. Then, we use bilinear pooling [23] to fuse the SRM noise stream and the RBG stream. Bilinear pooling is proposed for the fine-grained classifier. It has two CNN network structures, and the features extracted by the two CNN networks are fused through bilinear pooling. After the fused features pass through a decision network consisting of two fully connected layers, the final predicted result is obtained (see Figure 1). We use cross-entropy loss that leads to the following objective function:

$$L = L_{\text{cross}}\left(f_D\left(BP\left(f_{\text{RGB}}(x), f_N\left(f_{\text{SRM}}(x)\right)\right)\right), y\right), \quad (3)$$

where x is the input image, y is x's label, f_{SRM} denotes the SRM network with fixed weights, f_{RGB} and f_N are the RGB stream network and the noise stream network, BP denotes the bilinear pooling, f_D denotes the decision network, and L_{cross} denotes the cross-entropy loss.

The model accepts $32 * 32 * 3$ (width $*$ height $*$ channels) images and inputs two types of labels. This model is trained by the SGD optimizer. The parameters of the SGD optimizer are set as follows: momentum = 0.9 and weight_decay = 0. The hyperparameters of the network are set as follows: lr = 0.1 and batch_size = 64, and the training is designed to be 100 epoch, and lr is automatically changed to 0.1 times the original value every 30 epoch.

3. Experiments

In this section, we present an experimental evaluation of our method and compare it with several detection methods.

3.1. Experimental Setting

3.1.1. Dataset. We evaluate the detection method on the CIFAR-10 dataset. The CIFAR-10 dataset contains 60,000 32×32 color pictures, of which include 50,000 images for training and 10,000 images for testing.

For adversarial example datasets, we adopted three attack methods of FGSM, CW, and DeepFool. We used three methods to attack VGG16 [34], Resnet50 [35], and LeNet [36] and finally got 9 adversarial example datasets. For convenience, we named the adversarial example dataset generated by the VGG16 as "Adv-VGG16-Set." We used the Adv-VGG16-Set to train the two-stream network model (FNet) and baseline models. The datasets generated by attacking ResNet50 and LeNet are used for black-box testing to test the models trained on Adv-VGG16-Set. The parameter settings of the three attack methods are shown in Table 5. Previous work showed that nontargeted attack is easier to succeed, results in smaller perturbations, and transfers better to different models. So, we tested our method by nontargeted adversarial examples.

3.1.2. Classifier. For the CIFAR-10 dataset, we trained three models: VGG16 [34], ResNet50 [35], and LeNet [36]. These models were trained by the SGD optimizer (momentum = 0.9; weight_decay = 0), and the hyperparameters are set as follows: lr = 0.01, batch_size = 64, epoch = 30, and set lr to be multiplied by 0.1 times for every 10 epochs.

3.1.3. Baseline Models. We compared our method (FNet) with other detection methods including RGB-Net, SRM-Net, and KD + BU [32]. RGB-Net is a single-stream network that only inputs RGB images for judgment. It has the same network structure as the RGB stream part of the two-stream network we designed; similarly, SRM-Net only has the SRM part of our network. KD + BU [32] uses Bayesian uncertainty and model kernel density estimation to determine whether the sample is adversarial. For convenience, we use FNet to refer to our method.

3.1.4. Attack Methods. We adopted three attack methods from the Adversarial Robustness Toolbox designed by Microsoft [37]: FGSM [18], CW [27], and DeepFool [28]. For RGB-NET, SRM-NET, KD + BU, and FNet, we all used Adv-VGG16-Set for training. In training, we train a new detector using only one attack method each time.

3.1.5. Evaluation Metric. We use precision score, recall score, and area under ROC curve (AUC) score as the evaluation metric of the model [38]. The closer the AUC score is to 1, the larger the area under ROC curve and the better the model.

The calculation formula of the metric is as follows:

Table 4: The detailed two-stream network architecture for CIFAR-10. Conv (d, k, s) denotes the convolutional layer with d as dimension, k as kernel size, and s as stride.

RGB stream	SRM stream
Conv (64, 3, 1)	Conv (64, 3, 1)
BatchNorm layer, Tanh	ABSLayer
Avg pooling	BatchNorm layer, Tanh
Conv (128, 3, 1)	Avg pooling
BatchNorm layer, Tanh	Conv (128, 3, 1)
Avg pooling	BatchNorm layer, Tanh
Conv (256, 3, 1)	Avg pooling
BatchNorm layer, ReLU	Conv (256, 3, 1)
Conv (256, 3, 1)	BatchNorm layer, ReLU
BatchNorm layer, ReLU	Conv (256, 3, 1)
MAX pooling	BatchNorm layer, ReLU
	MAX pooling
Full connected 4096, ReLU, dropout (0.5)	
Full connected 4096, ReLU, dropout (0.5)	
Softmax 2	

Table 5: Parameter setting of three attack methods in adversarial robustness toolbox.

Attack method	Parameter
FGSM	Norm = 2, eps = 2.0, eps_step = 0.1
DeepFool	Eps = 0.1
CW	Lr = 0.2, confidence = 0.1

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \qquad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{4}$$

where TP denotes true positive rate, FP denotes false positive rate, and FN denotes false negative rate.

4. Experimental Results

On the CIFAR-10 dataset, Tables 6–8 show the precision and recall scores of different detection methods on the normal images and adversarial images. The bold values in tables represent the results of experiments conducted by our method (FNet). We can see that RGBNet and our method (FNet) have excellent effects in defending against both white-box attacks and black-box attacks. The precision score of FNet reaches 93.2% on adversarial images generated by DeepFool on VGG16. Experimental results show that it is difficult to detect adversarial examples generated by the CW method. SRM-Net is almost invalid against CW. KD + BU and RGB-Net achieve low scores when detecting CW. However, FNet improves KD + BU by more than 30%, and the precision score of FNet reaches 90.8% on detecting adversarial examples generated by CW. As we mention above, all detectors are trained on Adv-VGG16-Set. In addition, we can see that our method with good transferability can well detect adversarial samples generated by black models. SRM-Net and KD + BU are almost invalid against adversarial examples generated on black models, while the precision score of FNet reaches 90.1% against CW.

Figure 2 shows ROC curves of detection methods on CIFAR-10. The figures in the first row show the detector performance of the three attack methods of different detection methods on the adversarial samples generated by the white-box model (VGG16). We can see that the AUC score of FNet is from 0.963 to 0.976, and the AUC score of KD + BU is 0.795 to 0.837. The figures show that it is difficult to detect CW attack. The AUC score of SRM-Net is 0.525, and the AUC score of KD + BU is 0.795. Our method achieves the best performance when detecting the adversarial samples generated by the white-box model. The figures in the second and third rows show the detector performance of different methods on the adversarial samples generated by the black-box model. Experimental results show that the AUC score of FNet reaches 0.954. KD + BU and SRM-Net achieve low AUC score when detecting samples generated by the black-box model. The best AUC score of KD + BU is 0.715 when detecting DeepFool attack on ResNet.

Combining precision, recall, and AUC scores, RGB-Net performs second to our method (FNet) and KD + BU ranks third. In Figure 2, we can clearly see that the performance of FNet is better than that of other detection

Table 6: Performance of normal images and their adversarial examples generated by FGSM on CIFAR-10.

Model		Method	Normal images		Adv images	
			Precision	Recall	Precision	Recall
White model	VGG16	RGB-Net	0.896	0.928	0.864	0.807
		SRM-Net	0.748	0.773	0.571	0.538
		KDBU [32]	0.902	0.643	0.580	0.876
		FNet	**0.926**	**0.926**	**0.868**	**0.868**
Black model	ResNet	RGB-Net	0.888	0.928	0.874	0.809
		SRM-Net	0.692	0.773	0.543	0.440
		KDBU [32]	0.648	0.643	0.426	0.431
		FNet	**0.912**	**0.926**	**0.879**	**0.854**
	LeNet	RGB-Net	0.919	0.928	0.821	0.801
		SRM-Net	0.731	0.773	0.359	0.309
		KDBU [32]	0.697	0.643	0.269	0.319
		FNet	**0.927**	**0.926**	**0.819**	**0.822**

The bold values represent the results of experiments conducted by our method (FNet).

Table 7: Performance of normal images and their adversarial examples generated by CW on CIFAR-10.

Model		Method	Normal images		Adv images	
			Precision	Recall	Precision	Recall
White model	VGG16	RGB-Net	0.912	0.856	0.843	0.903
		SRM-Net	0.539	1.000	0.000	0.000
		KDBU [32]	0.852	0.525	0.617	0.893
		FNet	**0.913**	**0.922**	**0.908**	**0.898**
Black model	ResNet	RGB-Net	0.916	0.856	0.840	0.906
		SRM-Net	0.544	1.000	0.000	0.000
		KDBU [32]	0.545	0.525	0.457	0.478
		FNet	**0.883**	**0.922**	**0.901**	**0.855**
	LeNet	RGB-Net	0.943	0.856	0.792	0.914
		SRM-Net	0.625	1.000	0.000	0.000
		KDBU [32]	0.589	0.525	0.330	0.390
		FNet	**0.903**	**0.922**	**0.865**	**0.835**

Table 8: Performance of normal images and their adversarial examples generated by DeepFool on CIFAR-10.

Model		Method	Normal images		Adv images	
			Precision	Recall	Precision	Recall
White model	VGG16	RGB-Net	0.913	0.861	0.848	0.905
		SRM-Net	0.766	0.776	0.734	0.724
		KDBU [32]	0.857	0.526	0.619	0.898
		FNet	**0.908**	**0.944**	**0.932**	**0.888**
Black model	ResNet	RGB-Net	0.917	0.861	0.845	0.907
		SRM-Net	0.649	0.776	0.650	0.498
		KDBU [32]	0.554	0.526	0.466	0.495
		FNet	**0.868**	**0.944**	**0.926**	**0.828**
	LeNet	RGB-Net	0.940	0.861	0.806	0.913
		SRM-Net	0.661	0.776	0.510	0.370
		KDBU [32]	0.577	0.526	0.341	0.390
		FNet	**0.881**	**0.944**	**0.900**	**0.798**

The bold values represent the results of experiments conducted by our method (FNet).

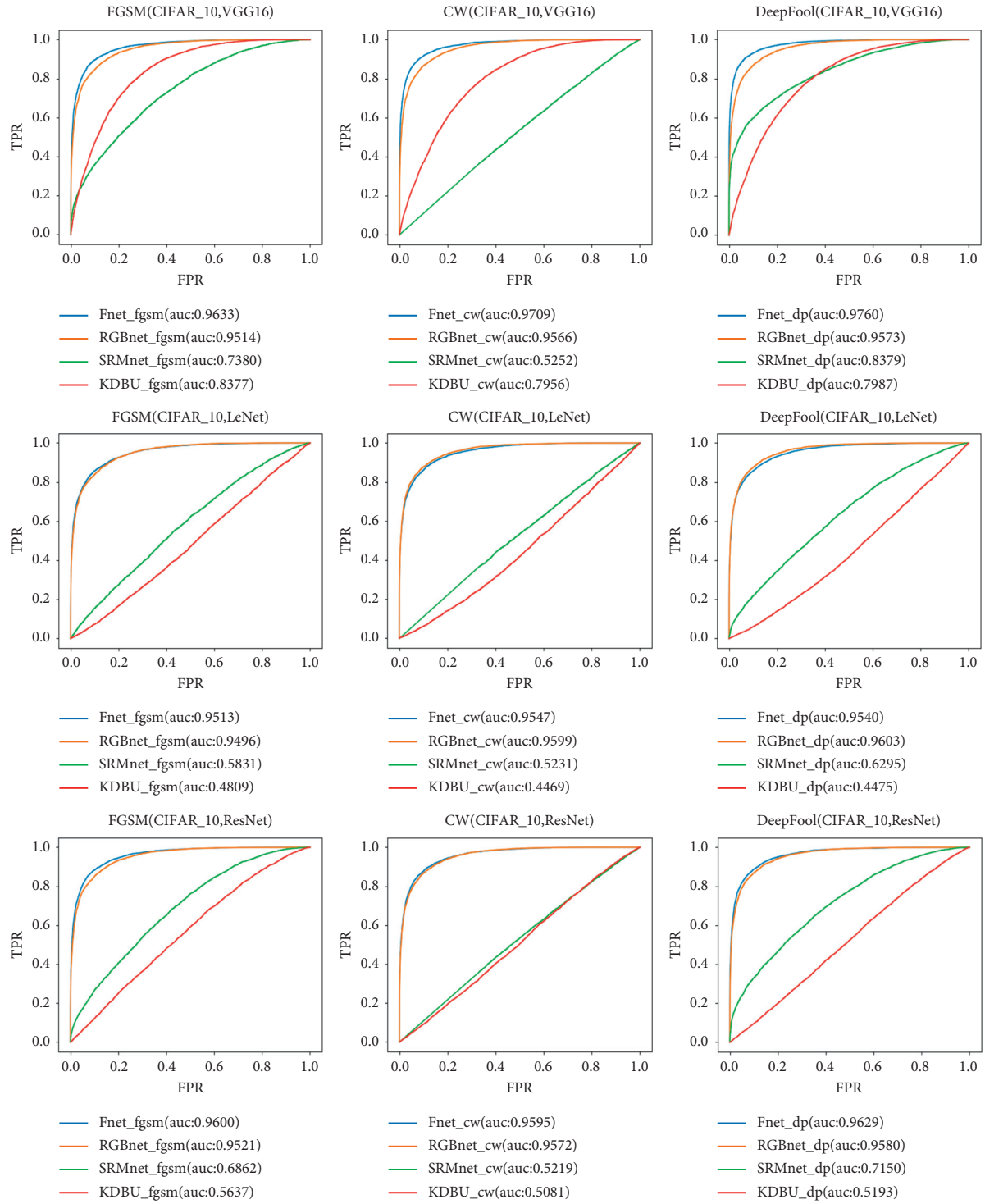

Figure 2: ROC curves of detection methods on CIFAR-10. We choose to display the ROC curves of detection methods on three different attacks. We can intuitively see that our method (FNet) is better than other methods in all cases.

methods in all cases. RGB-Net and KD + BU perform well when defending against white-box attacks, but they do not perform well against black-box attacks.

5. Conclusion

In this paper, we propose a two-stream model including RGB stream and SRM residual stream to improve the security of deep learning services based on 5G. We use 30 SRM filters to extract linear noise features and perform nonlinear processing to obtain rich features of 40 channels. We input these noise features into the network as additional evidence for detection. Experiments show that our method performs well against both black-box and white-box attacks. Moreover, our method has good transferability.

Although our method is effective to detect adversarial examples on images, the method is not effective on all types of data. The spatial rich model (SRM) feature extraction method is suitable for image data. Our two-stream network adopts SRM steganalysis to obtain features as additional evidence for adversarial detection, which is only effective for images. Therefore, our method is only effective for deep learning services which provide services such as image recognition. In the future work, we will explore more effective methods of adversarial sample detection and recovery.

Data Availability

The experiment data used to support the findings of this study are included within the article. The experiment data are described in Section 4 in detail.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially sponsored by the National Key R&D Program of China (No. 2019YFB2101700), the National Science Foundation of China (Nos. 62172297 and 61902276), the Key Research and Development Project of Sichuan Province (No. 21SYSX0082), Tianjin Intelligent Manufacturing Special Fund Project (No. 20201159), Natural Science Foundation of Tianjin City grant (No. 19JCQNJC00200), and the Australian Research Council Linkage Project (No. LP190100676).

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[2] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA, USA, June 2015.

[3] M. Shahverdy, M. Fathy, R. Berangi, and M. Sabokrou, "Driver behavior detection and classification using deep convolutional neural networks," *Expert Systems with Applications*, vol. 149, Article ID 113240, 2020.

[4] Z. Zhang, J. Geiger, J. Pohjalainen, A. Mousa, and B. Schuller, "Deep learning for environmentally robust speech recognition: an overview of recent developments," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 5, pp. 1–28, 2018.

[5] D. Amodei, S. Ananthanarayanan, and R. Anubhai, "Deep speech 2: end-to-end speech recognition in English and Mandarin," in *Proceedings of the International Conference on Machine Learning*, pp. 173–182, PMLR, New York, NY, USA, June 2016.

[6] J. Guo, H. He, H. Tong et al., "GluonCV and GluonNLP: deep learning in computer vision and natural language processing," *Journal of Machine Learning Research*, vol. 21, no. 23, pp. 1–7, 2020.

[7] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[8] R. G. Lipsey, K. I. Carlaw, and C. T. Bekar, *Economic Transformations: General Purpose Technologies and Long-Term Economic Growth*, OUP Oxford, Oxford, UK, 2005.

[9] M. Chen, U. Challita, W. Saad, C. Yin, and M Debbah, "Machine learning for wireless networks with artificial intelligence: a tutorial on neural networks," 2017, https://arxiv.org/abs/1710.02913.

[10] M. G. Kibria, K. Nguyen, G. P. Villardi, O. Zhao, K. Ishizu, and F. Kojima, "Big data analytics, machine learning, and artificial intelligence in next-generation wireless networks," *IEEE access*, vol. 6, pp. 32328–32338, 2018.

[11] K. Jia, M. Kenney, J. Mattila, and T. Seppala, "The application of artificial intelligence at Chinese digital platform giants: baidu, alibaba and tencent," *ETLA Reports*, vol. 81, 2018.

[12] Y. Deng, T. Zhang, G. Lou, X. Zheng, J. Jin, and Q. L. Han, "Deep learning-based autonomous driving systems: a survey of attacks and defenses," *IEEE Transactions on Industrial Informatics*, vol. 17, 2021.

[13] Y. Deng, X. Zheng, T. Zhang, C. Chen, G. Lou, and M. Kim, "An analysis of adversarial attacks and defenses on autonomous driving models," in *Proceedings of the 2020 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–10, IEEE, Pisa, Italy, May 2020.

[14] G. Xu, G. H. Xin, L. Jiao et al., "A semi-black-box android adversarial sample attack framework against DLAAS," 2021, https://arxiv.org/abs/2105.11593.

[15] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, https://arxiv.org/abs/1503.02531.

[16] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, "A study of the effect of jpg compression on adversarial images," 2016, https://arxiv.org/abs/1608.00853.

[17] B. Liang, H. Li, M. Su, X. Li, W. Shi, and X. Wang, "Detecting adversarial image examples in deep neural networks with adaptive noise reduction," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, 2018.

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

[19] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[20] J. Fridrich and M. Goljan, "Practical steganalysis of digital images: state of the art," *International Society for Optics and Photonics*, vol. 4675, pp. 1–13, 2002.

[21] N. F. Johnson and S. Jajodia, "Exploring steganography: seeing the unseen," *Computer*, vol. 31, no. 2, pp. 26–34, 1998.

[22] S. Wu, S. Zhong, and Y. Liu, "Deep residual learning for image steganalysis," *Multimedia Tools and Applications*, vol. 77, no. 9, pp. 10437–10453, 2018.

[23] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1449–1457, Santiago, Chile, December 2015.

[24] V. Behzadan and A. Munir, "Vulnerability of deep reinforcement learning to policy induction attacks," in *Proceedings of the International Conference on Machine Learning and Data Mining in Pattern Recognition*, pp. 262–275, Springer, New York, NY, USA, July 2017.

[25] P. Madani and N. Vlajic, "Robustness of deep autoencoder in intrusion detection under adversarial contamination," in *Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security*, pp. 1–8, Raleigh, NC, USA, April 2018.

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

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

[28] S.-M. Moosavi-Dezfooli, A. Fawzi, and F. Pascal, "DeepFool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574–2582, Las Vegas, NV, USA, June 2016.

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

[30] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, IEEE, San Jose, CA, USA, May 2016.

[31] J. Liu, W. Zhang, Y. Zhang et al., "Detection based defense against adversarial examples from the steganalysis point of view," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4825–4834, Long Beach, CA, USA, June 2019.

[32] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," 2017, https://arxiv.org/abs/1703.00410.

[33] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1053–1061, Salt Lake City, UT, USA, 2018.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[36] 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.

[37] M.-I. Nicolae, M. Sinn, M. N. Tran et al., "Adversarial robustness toolbox v1. 0.0," 2018, https://arxiv.org/abs/1807.01069.

[38] N. Carlini, A. Athalye, N. Papernot et al., "On evaluating adversarial robustness," 2019, https://arxiv.org/abs/1902.06705.