Hindawi
Security and Communication Networks
Volume 2021, Article ID 5537041, 11 pages
https://doi.org/10.1155/2021/5537041

Research Article

An IoT Time Series Data Security Model for Adversarial Attack Based on Thermometer Encoding

Zhongguo Yang,[1] Irshad Ahmed Abbasi,[2] Fahad Algarni,[3] Sikandar Ali,[4,5] and Mingzhu Zhang[1]

[1]*School of Information Science and Technology, North China University of Technology, Beijing, China*
[2]*Department of Computer Science, Faculty of Science and Arts at Belgarn, University of Bisha, Sabt Al-Alaya 61985, Saudi Arabia*
[3]*College of Computing and Information Technology, Faculty of Computing and Information Technology, University of Bisha, Bisha 61922, Saudi Arabia*
[4]*Department of Computer Science & Technology, China University of Petroleum, Beijing 102249, China*
[5]*Beijing Key Laboratory of Petroleum Data Mining, China University of Petroleum-Beijing, Beijing 102249, China*

Correspondence should be addressed to Sikandar Ali; sikandar@cup.edu.cn

Received 21 January 2021; Revised 7 February 2021; Accepted 25 February 2021; Published 10 March 2021

Academic Editor: Shafiq Muhammad

Nowadays, an Internet of Things (IoT) device consists of algorithms, datasets, and models. Due to good performance of deep learning methods, many devices integrated well-trained models in them. IoT empowers users to communicate and control physical devices to achieve vital information. However, these models are vulnerable to adversarial attacks, which largely bring potential risks to the normal application of deep learning methods. For instance, very little changes even one point in the IoT time-series data could lead to unreliable or wrong decisions. Moreover, these changes could be deliberately generated by following an adversarial attack strategy. We propose a robust IoT data classification model based on an encode-decode joint training model. Furthermore, thermometer encoding is taken as a nonlinear transformation to the original training examples that are used to reconstruct original time series examples through the encode-decode model. The trained ResNet model based on reconstruction examples is more robust to the adversarial attack. Experiments show that the trained model can successfully resist to fast gradient sign method attack to some extent and improve the security of the time series data classification model.

1. Introduction

IoT amalgamates well-known products with state of the art infrastructures including distributed data storage, big data solutions, artificial intelligence (AI) utilities, or cloud [1]. Internet of Things (IoT) envisions connected, pervasive, and smart nodes link independently while providing all kinds of services. IoT data are collected at large to aid in decision-making. The IoT consumer products are no longer just the product only; it is the data, the product, the infrastructure, and the algorithms. These IoT products have switched to connected technologies from analog one, therefore, introducing novel risks for consumers regarding potential safety, privacy, and security issues for discriminatory data [2–4].

Moreover, Papernot et al. [5] have found that the adversarial samples are more transferable amongst various machine learning approaches, i.e., support vector machine, logistic regression, decision tree, and deep neural networks.

There are many application scenarios for IoT, as shown in Figure 1, including medical health, electricity, and intelligence device. There are also areas, which are very sensitive to attacks, such as industrial control decision support systems.

In other fields, such as State Grid and Industrial Control, the deep learning model built for them is prone to make decision errors due to data noise and deliberate attacks to modify data. For example, smart grids time series data were analyzed for electricity fraud detection, wherein these use cases perturbed data can succor thieves from being detected.

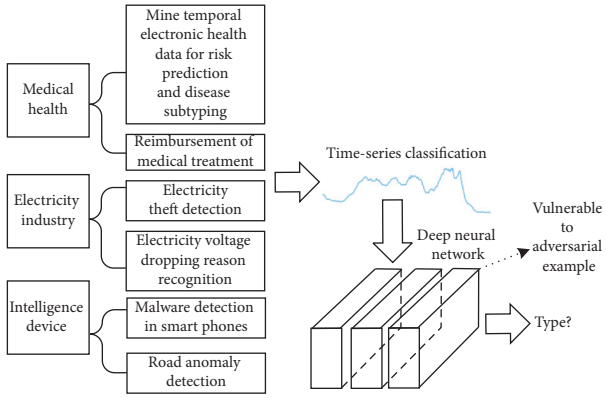

FIGURE 1: Typical application fields of the deep neural network for IoT data.

As illustrated in Figure 1, in some sensitive and crucial systems, time-series data classification models are admired for their vast application. Thus, security and precisely the ability to detect the nodes being compromised, along with collecting and preserving evidence of malicious activities or an attack transpire as a priority in the triumphant deployment of IoT networks. Among these potential risks, AI algorithm security is rarely gained research interest although it is a very hot topic in the domain of AI.

Modern approaches in time-series data classification are based on the deep learning paradigm [6], specifically adversarial examples, which could lead to big recognition errors by adding small perturbation to the original time series.

The reason lies in the high dimension linear design of deep learning models. In order to better combat against the adversarial attack, we applied an encode-decode model to reconstruct time series examples from thermometer encoding of original time series. Although the classification models are not trained on the reconstruction examples, their training and valuation accuracy is the same as the model trained on the original examples. Moreover, we found that the new model is robust to the fast gradient sign method (FGSM) attack to some extent. To summarize, the contributions of this article are three folds as follows:

(1) Summarize some potential risks in the IoT time series classification (TSC) model

(2) Analyze the classification activation map and the attack area in time series

(3) A robust model-based encode-decode and thermometer encoding

The remaining paper is structured as follows. In Section two, we overview TSC works based on deep learning as well as attacks and defense methods in the fields of computer vision. Section three shows some potential risks in IoTTSC from different views. Section four presents a detailed description of our method and some basic theory. In the experiment section, we introduce the datasets, classification model architecture, and attack defense results. Finally, we analyze the defense effectiveness and give our future research directions.

2. Related Works

TSC problems are experienced in numerous real-life data mining tasks ranging from power consumption monitoring [4], food safety [7], and health care [2, 8, 9].

Deep learning has resolved some problems like pattern recognition in temporal and spatial data with higher accuracy that was thought to be impossible a few years ago. Fortunately, TSC tasks can be efficiently framed as deep learning problems; therefore, many researchers have recently begun to adopt deep learning models for TSC tasks [6].

The classification of time series IoT data is a key problem in various application domains. Backing the development of deep learning, investigators have started to work on the vulnerability of deep neural networks to adversarial attacks [10]. In the field of image processing, an adversarial attack alters original images in such a way that the modifications are nearly imperceptible by a human. The altered image is termed as an adversarial image, that will be confused by the neural network and will be misclassified, while that of the original image will be correctly classified. The well-known real-world attack includes modifying a traffic sign image so that it is misconceived by an autonomous vehicle [11]. Alteration of illegal content to make it undetectable by automatic moderation algorithms is another example. The most notable attack is gradient-based attack, where the attacker alters the image in the direction of the gradient of the loss function with reference to the input image and therefore escalates the rate of misclassification [12, 13].

The model of deep learning applied in a real environment on IoT data is fragile which is vulnerable to adversarial attack, and this has become a common problem of deep learning in other areas. At the same time, there are much security works for image processing such as defensive distillation [14], data compression [15], depth compression network [16], data randomization [17], and gradient regularization [18].

There are hardly any comprehensive studies on defense against an attack on temporal data. Fawaz et al. [8] discussed some serious problems in the classification of time-series data using a deep learning model. Different from the image, IoT time-series data own its special characteristics, such as dynamic changing and different sampling scale. Based on the characteristics of IoT data, this paper uses an encode-decode model-based deep neural network.

In the encode-decode stage, we used a thermometer coding method to be the decoded output. The reason to use the thermometer coding is to consider bringing a strong nonlinear transformation to the model. This is inspired by Goodfellow, who showed us the high dimension of the well-structured deep learning model. Buckman's et al. [19] work confirmed that the input discretization approach could repel against adversarial attacks. Inspired by these thoughts, different from the aforementioned works, we try to construct a whole network. In this network, the input is the original curves, and it will learn its original curve through the encode-decode model with its thermometer encoding as input. With the thermometer coding as input to the ResNet to

predict its type, we will show the details of the proposed network and its effectiveness in the following parts.

3. Adversarial Attack in IoT Time Series Data

In this paper, we used Coffee's dataset [20] as typical time series data to illustrate the adversarial attack phenomena in IoT fields and ResNet [21] as a measure for neural network architecture.

3.1. Fast Gradient Sign Method and TSC Adversarial Attacks. Some adversarial examples and definition of the TSC problem were introduced by Fawaz et al. [8]. According to them, time series data can be mathematically represented as set $X = [x_1, x_2, \ldots, x_T]$. Let T is a real number and represents the length of X. Further, there is a well-trained deep learning model $f(\cdot) \in F$: $\mathscr{R}^T \longrightarrow \hat{Y}$. Here, Y is the label space of time series, and $\mathscr{R}$ is a real number space. The adversarial example has to find another example X' to be a perturbed cloned version of X with the restriction that $X - X' < \varepsilon$ and $Y \neq Y'$. A visual illustration of given definitions is visualized in Figure 2.

The most classic adversarial method is the fast gradient sign method (FGSM). FGSM was first introduced by Goodfellow et al. [12] for generating adversarial images that trick the well-known GoogLeNet model. The attack is set up through a one-step gradient update in the direction of the gradient's sign at every single timestamp.

The perturbation procedure shown in Figure 3 can be represented mathematically as follows:

$$\eta = \varepsilon \cdot sign\left(\nabla_x \mathcal{J}\left(X, Y_{true}\right)\right), \tag{1}$$

where ε symbolizes the magnitude of the perturbation. The adversarial time series X' can be computed using $X' = X + \eta$.

Author of the FGSM paper mentioned the underlying reason why FGSM attacks the neural network. Firstly, the influence of disturbance in the neural network will be as big as snowball due to the linear design of the model. At present, ReLU is a kind of linear activation function in neural networks, which makes the whole network tend to be linear. Furthermore, the larger the dimension of input, the more vulnerable will be the model to adversarial attack.

3.2. The Distribution of Data in Adversarial Attack. Multidimensional scaling (MDS) [22] provides a possibility to get insights into the spatial distribution of the input time series. MDS project N-dimensional space into two-dimensional space while keeping the relative distance for any two time series. Given the nearest neighbor classifier achieving low accuracy on the raw time series, Euclidean distance (ED) could not be used directly in the raw data.

However, the high feature learned by the network could be used as a good presentation of the raw time series. Commonly, the perfectly connected layers in the last several layers of the neural network are often used as latent space, where the class-specific region differs for different classes.

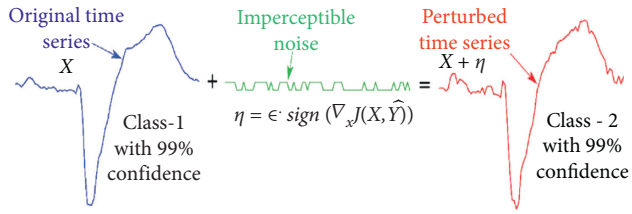

Figure 2: Adversarial examples taken from [8].

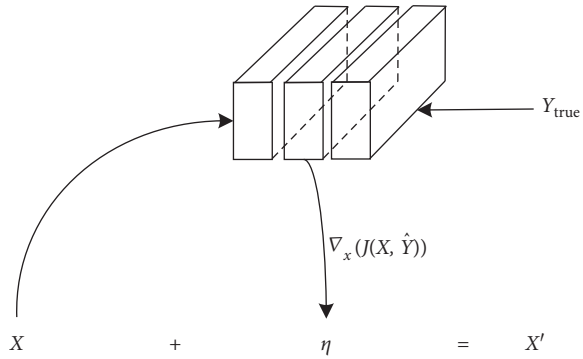

Figure 3: The principle of the fast gradient sign method.

We apply this method on ResNet, which achieves the best accuracy on most of the TSC problems [6]. In the ResNet architecture, there is a global average pooling (GAP) layer preceding the classifier layer. The GAP layer is a learned good representation of the raw time series, which is used to compute ED. When we get the distance for each pair of two time series, the metric MDS is a cost function called stress and can be obtained as follows:

$$\text{Stress}_D\left(X_1, \ldots, X_N\right) = \left(\frac{\sum_{i,j}\left(d_{ij} - x_i - x_j\right)^2}{\sum_{i,j} d_{ij}^2}\right)^{1/2}, \tag{2}$$

where d_{ij} is the ED between the GAP vectors of time series X_i and X_j. In this way, the original raw time series space is largely reduced to two-dimensional space. Each time series X_i is represented as a single data point x_i.

The visualization of MDS shows the distribution of the data in the raw data space to some extent. Here, we used the same technique to show how the adversarial attack works from the data distribution angle. The Coffee dataset is used as an example, and the ResNet is applied as a base neural network. Details are shown in Figure 4.

As shown in Figure 4(a), one can easily separate the set of time series belonging to the two classes by utilizing MDS on the latent representation learned by the network. Yet, in Figures 4(b)–4(d), with the attack ratio eps becoming larger, it becomes harder to classify these two datasets by using linear classifier in the two-dimensional space. With the help of MDS, we could observe that the adversarial attack surprisingly changes the distribution of data.

3.3. Transferability of Adversarial Examples. Transferability is the usual property for adversarial attack examples. The adversarial attack against a neural network can trick neural

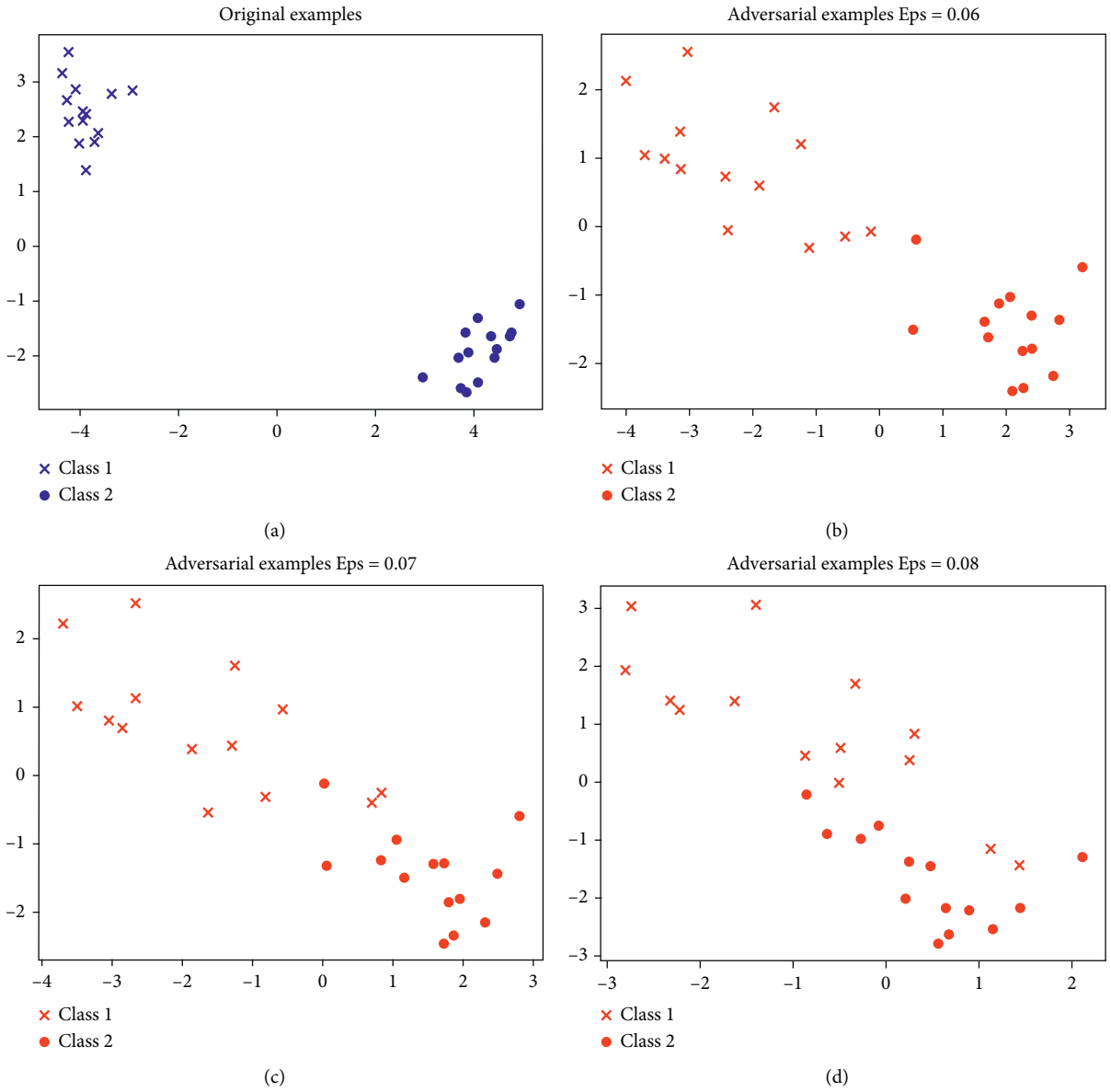

Figure 4: The data distribution of original examples and adversarial examples. (a) The raw time series in ResNet; (b) The adversarial examples in ResNet for eps = 0.06; (c) The adversarial examples in ResNet for eps = 0.07; (d) The adversarial examples in ResNet for eps = 0.08.

networks trained by diverse datasets [23]. Moreover, adversarial attacks for a special architecture can trick other classifiers trained by different machine learning algorithms or even other's neural networks with dissimilar architectures [5].

Recently, Tramèr et al. [24] found that on average, the distance to the model's decision boundary is larger than the distance between two models' boundaries in the same direction which confirms the existence of transferability of adversarial attack examples up to some extent. They also prove, by presenting a counter-example, that transferability is not an intrinsic characteristic of deep neural networks.

Table 1 illustrates that the adversarial examples in time series for one model could also attack other models even they own totally different structures. So, the white-box attack could be launched by generated adversarial examples for other well-known deep neural network models.

As presented in Table 1, the adversarial examples p against FCN could fool ResNet to achieve a low accuracy and vise visa. The experiment shows the transferability of adversarial examples exists in the TSC problem which means the black-box attack could be launched even the details of the backed algorithm are unknown.

3.4. Random Noise Attack. Nguyen et al. [25] uncovered a new type of attack called false positive attack, where adversarial attack examples are misclassified by deep neural networks with the confidence of 99%.

Table 1: Accuracy of adversarial examples in different models.

Eps	Clean	Accuracy for ResNet on ResNet adversarial examples	Accuracy for ResNet on FCN adversarial examples	Accuracy for FCN on FCN adversarial examples	Accuracy for FCN on ResNet adversarial examples
0.08	1	0.6429	0.6429	0.4643	0.7142
0.1	1	0.4286	0.9285	0.3929	0.6071
0.12	1	0.3929	0.8571	0.25	0.5357
0.15	1	0.25	0.5357	0.1429	0.4286

Typically, we trained a machine learning model by the following process, as shown in Figure 5; the trained model is deployed to the industry environment after being evaluated on the prepared test dataset. This is extremely dangerous in the environment of IoT due to its device controlling characteristics.

We trained a ResNet model to classify some randomly generated noise data along with the time series data. Unmistakably, the random time series data will be rejected by the classifier with low confidence. However, the random noise data classified were classified as class two with high confidence that prove that there is a potential risk in the model. Some predicted labels of the samples of noise time series examples are visualized in Figure 6.

As illustrated by Figure 6, we notice that even zero values or random noise also can lead to high confidence output. As a result, the model cannot be used directly for intelligent devices.

3.5. Class Activation Map and Adversarial Examples. Class activation map (CAM) proposed by Zhou et al. [26] was exploring to find the discriminative and susceptible field of an image. Later, Wang et al. [9] proposed a one-dimensional CAM application in TSC. Here, we use the CAM method to highlight the susceptible region of a time-series data. Consequently, the susceptible fields of the time-series data are continually distributed which potentially enable that some preprocessing method could improve the robustness of the model.

This method describes the classification of a definite deep learning model to underline the subsequences that contribute the most to a specific classifier. It is to be noted that utilizing CAM is only feasible for the models with a GAP layer prior to the softmax classifier. That is the reason, in this section, we only measured the ResNet model that achieves the highest accuracy for majorities of the datasets. ResNet benefits from the CAM approach using a global average pooling (GAP) layer that helps identify possible regions of an input time series data that contribute to the certain classifier.

Let $A(t)$ be the result of the last convolutional layer MTS with M variables. $A_m(t)$ is the univariate time series for the variable $m \in [1, M]$, where $m \in [1, M]$ is the result of applying the mth filter. Let w_m^c be the weight between the output neuron of class c and the mth filter. As a GAP layer is utilized, therefore, the input to the neuron of class c, i.e., (Z_c) can be computed using the following equation:

$$Z_c = \sum_m w_m^c \sum_t A_m(t). \tag{3}$$

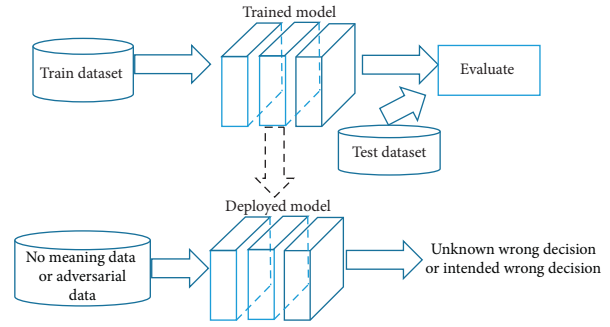

Figure 5: The process of training a model and the potential threats in the IoT device.

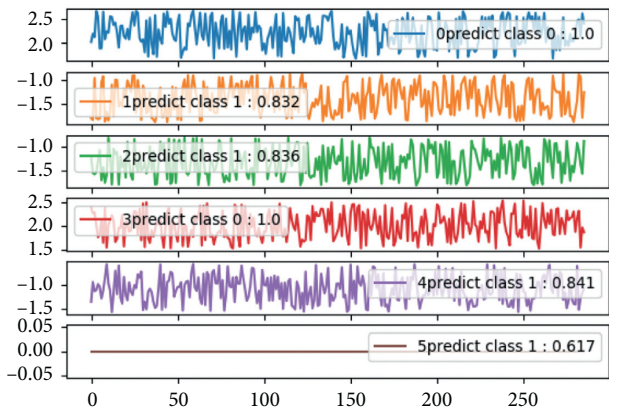

Figure 6: The noise data prediction is a valid class with high confidence.

The second summation contributes the averaged time series to the whole time dimension. For simplicity, the denominator is omitted here. The input Z_c can also be represented in equation form as follows:

$$Z_c = \sum_m \sum_t w_m^c A_m(t). \tag{4}$$

Lastly CAM_c, the class activation map, that explains the classification as label c is given by the equation as follows:

$$\mathrm{CAM}_c(t) = \sum_m w_m^c A_m(t). \tag{5}$$

Here, CAM is a univariate time series in which each item at a certain timestamp $t \in [1, T]$ is equal to the weights being learned by the neural network, i.e., weighted sum of the data points M at time t. Figure 7 shows the result of applying CAM, respectively, on the Coffee dataset.

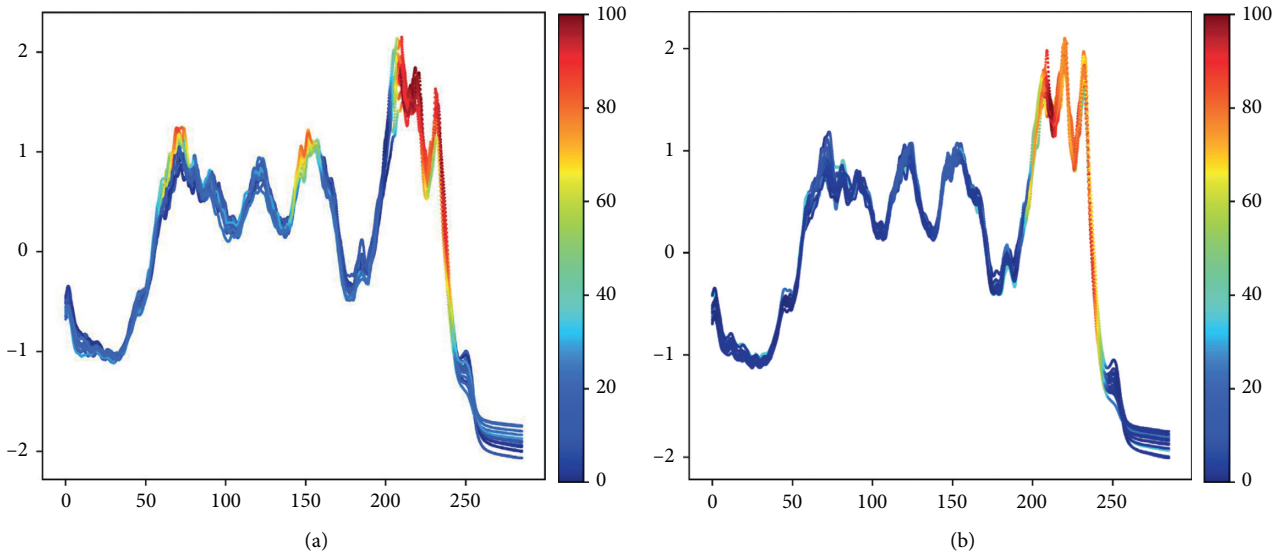

Figure 7: Classification activation map in Coffee dataset. (a) Coffee dataset for class 0. (b) Coffee dataset for class 1.

From Figure 7, we found that the key classification activation fields are in the same points where the vision difference exists. However, the adversarial examples are not trying to modify these places to defraud the classifier.

In Figure 8, we found the adversarial example is a tiny difference from the original time series and the changing place is not in the key area learned by the neural network.

4. Proposed Method

The results of the analysis in Section 3 provides evidence for some potential risks that exist in deep learning models besides the fact that best performance can be achieved in time series classification. Furthermore, in the IoT field, it is extremely dangerous if these algorithms are deployed in devices. We designed a new training strategy based on the encode-decode model to increase the robustness of the model.

Our method consists of two main parts: one is to encode-decode model and the second is a traditional deep neural network model for classification. In the encode-decode model, the input is the nonlinear transformation of the original time series. Here, we applied the thermometer encoding method as the nonlinear transformation. The decoded output is the original time series that is recovered from its thermometer encoding forms. The reason to use the encode-decode model is to take advantage of its nonlinear transforms to remove some noise and adversarial perturb which is based on linear gradient signs. The schema of our proposed method is shown in Figure 9.

Our method tries to bring nonlinear transformation by the encode-decode model which will defend the traditional adversarial attacks. The network consists of two main parts, one is encode-decode part. In this part, the network tries to learn a noise and nonlinear function which tries to minimize the loss of original example with the thermometer discretized examples.

The encoder maps the input to a fixed-length vector (which needs to contain all the input information) and the decoder then outputs the translation. In the model, the encoder learns a coding sequence representing the semantic information of time series, and the decoder maps the sequence to the original time series.

First, the time series will be discretized into an average of ten evenly spaced levels. Additionally, the thermometer encoding method is applied to the discretized curves. Based on the thermometer encoding time series, the encode-decode model is trained to reconstruct time series. Therefore, the loss function we used here is the mean-square error (MSE).

In the process of training the encode-decode model, we add some random noise to the time series that increase the reconstruction ability.

Figure 10 shows that the encoder part tries to learn some robust illustration of the input time series. The decoder tries to map the input to its original time series. Here, we add some random noise to the original input time series to increase the robustness of the encode-decode model.

In order to discretize the input time series x without losing the relative distance information, Buckman [3] proposed thermometer encodings. For an index $j \in \{1, \ldots, k\}$, let $\tau(j) \in \mathscr{R}^k$ be the thermometer vector defined as follows:

$$\tau(j)_l = \begin{cases} 1, & \text{if } l \geq j, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Then, the discretization function f is defined for a time index point $i \in \{1, \ldots, n\}$ as follows:

$$f_{\text{therm}}(x)_i = \tau(b(x_i)) = \text{Sum}(f_{\text{onehot}}(x_i)), \tag{7}$$

where Sum is the cumulative sum function and $f_{\text{onehot}}(x_i)$ is the one-hot coding method. The thermometer encoding

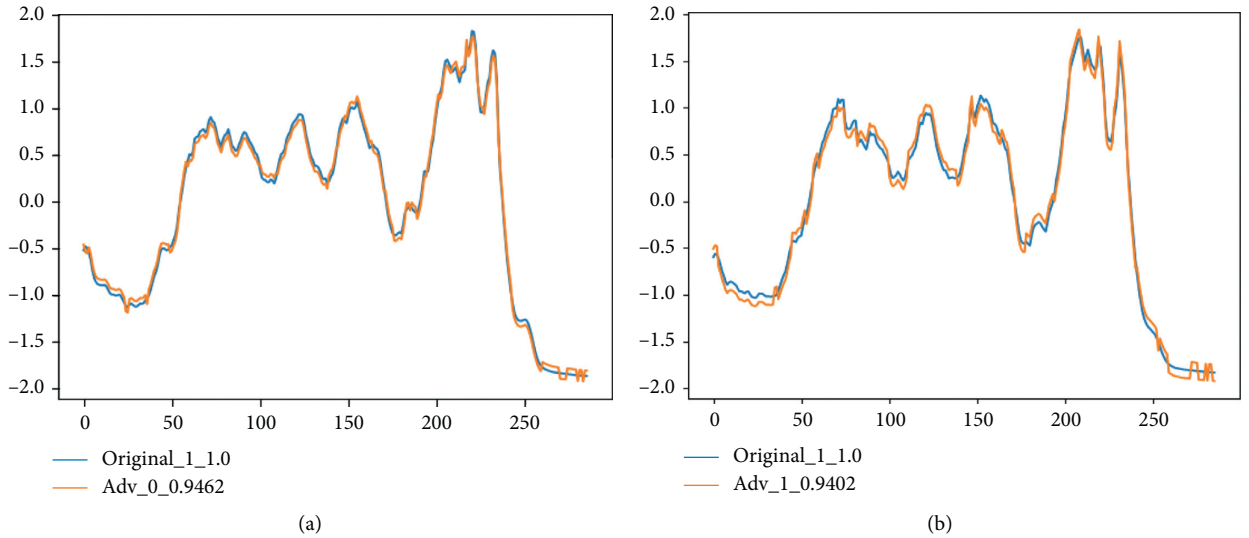

Figure 8: Two typical adversarial examples in the Coffee dataset.

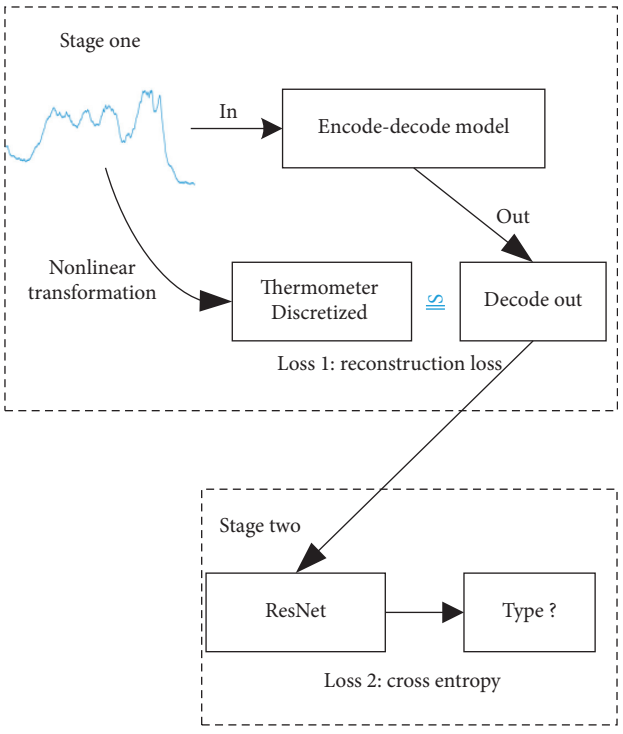

Figure 9: The procedure of the proposed method.

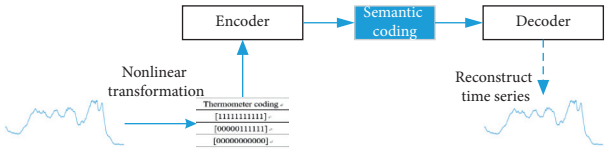

Figure 10: The encode-decode model in the proposed method.

could retain the order information for the time series, i.e., for pixels i and j, if $b(x_i) \neq b(x_j)$ and $x_i < x_j$, then $\tau(b(x_i))_2 < \tau(b(x_j))_2$.

This characteristic is very important for time series that hold the order and shape information of the original time series. Figure 11 shows the discretize process of a time series. Table 2 shows the thermometer encoding result of a continuous value.

The time series can be disseized by the average bin method and transformed into other code. The coding method is highly nonlinear, which could defense the attack for the gradient-based attack method.

The curve with certain noise can be restored normally after discretization and encode-decode model. The well-trained encode-decode model could recover the original time series from its thermometer encoding. We showed the example of the Coffee dataset to illustrate its effectiveness in Figure 12.

As illustrated in Figure 12, the reconstruction time series contains all the information of the original time series. The difference is the high-frequency part of the time series, which looks to link random noises. We showed that the deep learning model trained on these reconstruction examples could show high accuracy and ability to defend from adversarial attacks.

5. Experiment and Evaluation

In this section, we present an attack method FGSMs and ResNet [21] architecture. We then use FGSMs to generate adversarial time series attack examples for the ResNet model.

5.1. Data Sets and Comparison Method. 85 datasets of the UCR archive are utilized in experiments [27]. These datasets encapsulate diverse time series data from fields like electricity industry, food security, image, and sensors.

One of the dataset is electronic devices known as smart meters, which record detailed electricity consumption data. A previous study [28] showed that these electricity data could be used to analyze the type of electric device. The

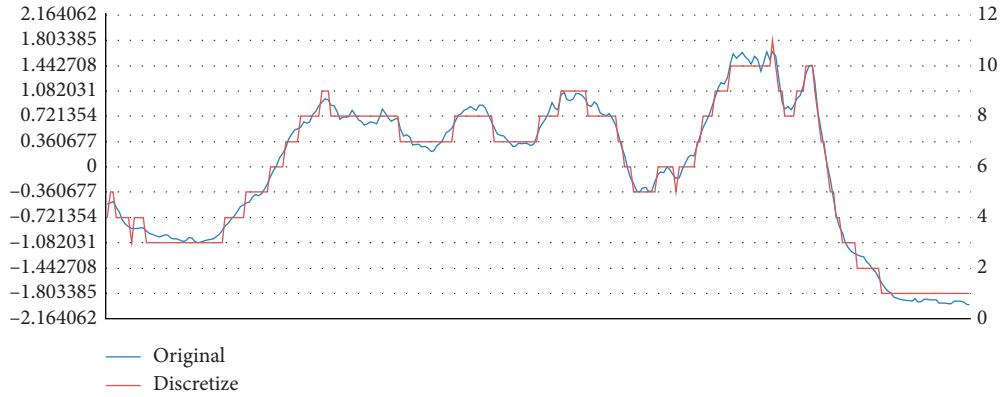

Figure 11: Examples of mapping continuous-valued inputs to quantized inputs and thermometer codes with ten evenly spaced levels.

Table 2: Examples of mapping continuous-valued inputs and thermometer codes with ten evenly spaced levels.

Continuous value	Quantized	Thermometer encoding
0.13	0	[11111111111]
0.54	0.5	[00000111111]
0.96	1	[00000000000]

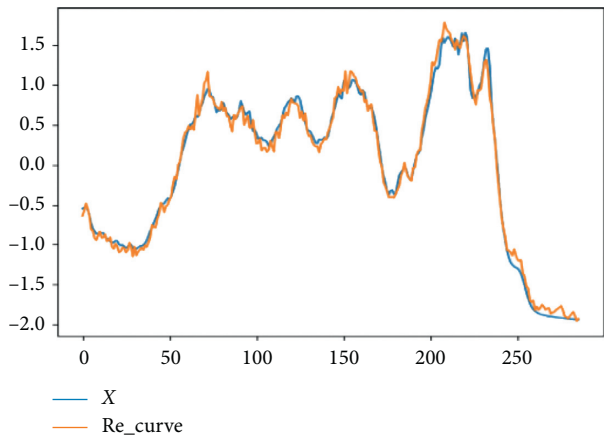

Figure 12: The reconstruction curve from the original time series.

consumption time series could be modified by adding some small perturbation to misguide the classifier in the device. The aim to collect and analyze the electricity consumption data is to monitor the device being used by the citizens' homes and in future to reduce carbon footprint. 375 univariate time series come under the umbrella of the dataset. The classes are Microwave, Toaster, and Kettle of length 720. The dataset classically illustrates IoT time series attack example and is a vital task in the intelligent device.

ResNet architecture the same as [8] has been employed for the comparison process. Details about the architecture and its parameter are shown in Table 3.

The block of ResNet is illustrated in Figure 13.

In ResNet, time series act as input and the possible classes K serve as an output. The convolution kernel size is 8, 5, and 3 for every individual block of the ResNet which indicates that for extracting some useful features, it will have the neighbor size 8, 5, and 3. The ResNet we employed,

Table 3: Layers details in one block of the ResNet.

name	Layer	Parameter
Conv_x	Conv1D	Filters = 64, kernel size = 8, stride = 1
Conv_x	BatchNormalization	
Conv_x	Activation	Function = ReLU
Conv_y	Conv1D	Filters = 64, kernel size = 5, stride = 1
Conv_y	BatchNormalization	
Conv_y	Activation	Activation function = ReLU
Conv_z	Conv1D	Filters = 64, kernel size = 3, stride = 1
Conv_z	BatchNormalization	

comprises of three blocks, and they have 64, 128, and 128 filters, respectively.

5.2. Result and Analysis of Attack and Defense. The experiments are conducted on Keras 2.1 and TensorFlow 1.8. The number of samples in training and testing phase is decided by the original public available dataset (UCR). We trained the encode-decode ResNet network and extracted the ResNet part as the attack target. The input of the attack model is the original time series; the gradient of this model is computed in the same way as illustrated in the work. Although in our method, we used a thermometer as the input to train the encode-decode model, and the comparison model is the same ResNet as illustrated in [8]. The experiments in this manuscript are carried out to show the efficiency of the encode-decode model in the defense part. The results of the defense are shown in Table 4. During the attack and defense stage, the perturbation ratio ε is set to 0.1.

In Table 4, we could see that the accuracy of most of the datasets is largely improved compared with encode-decode training. The result shown in Table 4 reveals that our method could resist the attack of FGSM in the TSC problem to some extent.

To future analyze, the encode-decode model could defend against the attack by FGSM, and we get the accuracy of a typical sensor dataset, i.e., Coffee dataset under different ε. First, we generate some adversarial examples using FGSM, and then the time series examples are smoothed or

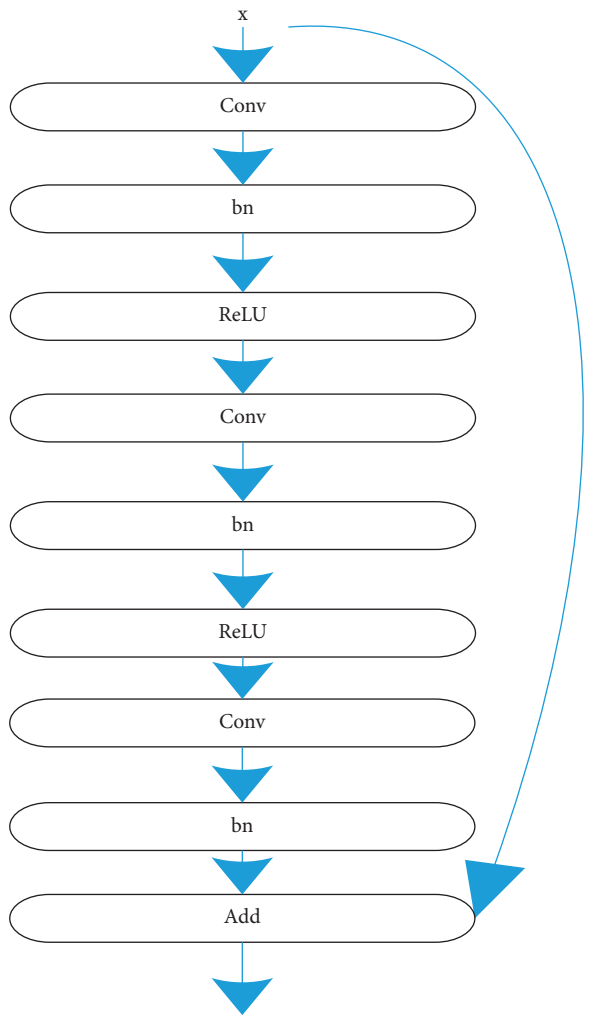

Figure 13: The block building in ResNet.

Table 4: Defense results of our method for ResNet and FGSM.

Dataset	ResNet_ori (Fawaz, et al, 2019)	ResNet_fgsm_adv (Fawaz, et al. 2019)	Encode-decode ResNet_fgsm_adv
50words	73.2	17.1	**49.22**
ADIAC	83.1	3.1	**10.16**
ArrowHead	85.1	33.1	**75**
Beef	76.7	20	**30**
BeetleFly	85	15	**60**
BirdChicken	95	55	**65**
Car	93.3	21.7	**35**
CBF	98.9	86.1	**98**
Coffee	100	50	**75**
SmallKitchenAppliance	78.9	40.5	**65.4**

transformed in different ways. Second, these processed examples are evaluated on these models. Details of accuracy on the Coffee dataset are shown in Figure 8. Coffee dataset [20] is a two-class problem that discriminates between Arabica and Robusta coffee beans.

The encode-decode ResNet shows a good defense result for this dataset, and the accuracy curve is shown in Figure 14. In this figure, we could find that the accuracy of these two datasets decreased slowly as the number of perturbation increases. It means the attack of FGSM still works here, but its effectiveness is largely reduced. The reason lies in the thermometer encoding because it is a highly nonlinear transformation. The thermometer encoding discretizes the time series and retains the order information about the original curve.

5.3. Preprocessing Method for Defense Adversarial Attack. Actually, in an industrial environment, we could apply some practical preprocessing methods such as time series smooth method to weaken the fluency of adversarial examples. Here,

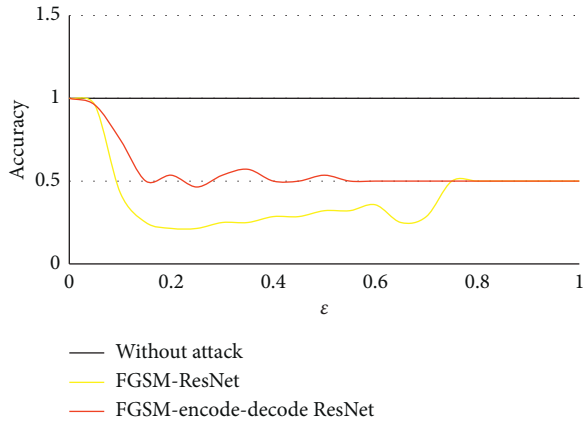

Figure 14: The accuracy of Coffee dataset in encode-decode ResNet with FGSM attack in different perturbation ratios.

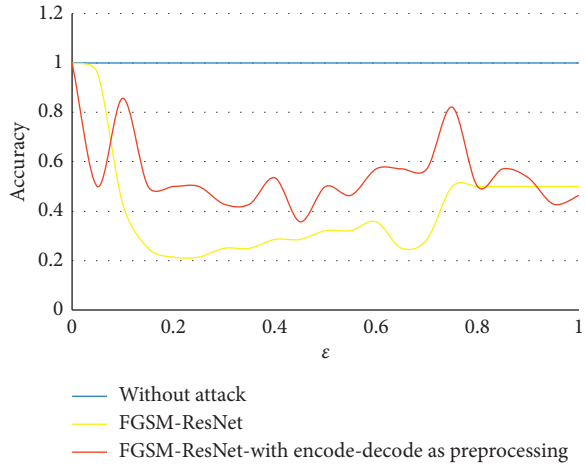

Figure 15: The accuracy of Coffee dataset in ResNet with the encode-decode model as a preprocessing method for defense FGSM attack.

we show two methods known as smooth and encode-decode to assist the attack.

In the experiment, we first applied the thermometer encoding method to transform the adversarial examples; then, the encode-decode model is used to map the thermometer encoding back into the original time series. Of course, the reconstructed time series is different from the original time series. The recognition accuracy is shown in Figure 15.

As illustrated in Figure 15, the yellow line is below the red line, which means the encode-decode model improves the accuracy of attacks by the FGSM. This result hints that the encode-decode model could be used as a data preprocessing method before being put into the classification model.

6. Conclusions

The proposed method of this paper is of using encode-decode model joint training strategy to strengthen the robustness of the deep learning model. The experiments reveal that our model can resist FGSM attacks to some extent. Moreover, the encode-decode model could be used as a way of preprocessing to weaken the attack from FGSM.

Though, it is not easy to eliminate the white-box attack launched by FGSM. Our method improves the robustness of the trained model but fails to resist the attack completely. To check the effectiveness of our method, more experiments on other datasets are required as well.

Moreover, we found that different trained models own different power against the same attack, and it is hard to evaluate the goodness of the model. Fundamentally, there are no theoretical studies on how to quantify the goodness or robustness of a trained model. Therefore, given the popularity of applying the deep learning method to IoT data analysis, it still needs more research to focus on the interpretability of deep learning models. Our future research directions include how to evaluate the defensive capability to adversarial examples in the area of IoT data.

Data Availability

The data used to support the findings of this study are included in the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Projects of International Cooperation and Exchanges NSFC (grant no. 62061136006), Beijing Natural Science Foundation (no. 4202021), and Scientific Research Initiation Funds (grant nos. 2462020YJRC001 and 110051360002).

References

[1] M. Conti, A. Dehghantanha, K. Franke, and S. Watson, "Internet of things security and forensics: challenges and opportunities," *Future Generation Computer Systems*, vol. 78, 2018.

[2] S. M. Abdelfattah, G. M. Abdelrahman, and M. Wang, "Augmenting the size of EEG datasets using generative adversarial networks," in *Proceedings of the International Joint Conference on Neural Network*, pp. 1–6, Rio de Janeiro, Brazil, August 2018.

[3] M. Shafiq, Y. Z. Tian, and M. X. GuizaniDu, "Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city," *Future Generation Computer Systems*, vol. 107, pp. 433–442, 2020.

[4] M. Shafiq, Z. Tian, A. K. Bashir et al., "CorrAUC: a malicious bot-IoT traffic detection method in IoT network using machine learning techniques," *IEEE Internet of Things Journal*, vol. 14, no. 99, pp. 1606–1615, 2018.

[5] N. Papernot, P. D. Mcdaniel, and I. J. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *Cryptography and Security*, vol. 2016, 2016.

[6] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. Muller, "Deep learning for time series classification: a

review," *Data Mining and Knowledge Discovery*, pp. 1–47, Springer, Berlin, Germany, 2019.

[7] J. L. Agnieszka Nawrocka, "Determination of food quality by using spectroscopic methods," in *Advances in Agrophysical Research*, S. G. A. A. Stepniewski, Ed., IntechOpen, London, UK, 2013.

[8] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. Muller, "Adversarial attacks on deep neural networks for time series classification," 2019.

[9] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: a strong baseline," in *Proceedings of the International Joint Conference on Neural Network*, pp. 1578–1585, Anchorage, AK, USA, May 2017.

[10] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: attacks and defenses for deep learning," *IEEE Transactions on Neural Networks*, vol. 30, no. 9, pp. 1–20, 2019.

[11] K. Eykholt, I. Evtimov, E. Fernandes et al., "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1625–1634, Salt Lake City, UT, USA, June 2018.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proceedings of the International Conference on Learning Representations*, San Diego, CA, USA, May 2015.

[13] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proceeding of the International conference on learning representations*, Toulon, France, April 2017.

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

[15] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, "A study of the effect of JPG compression on adversarial images," in *Proceedings of the Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, June 2016.

[16] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," in *Proceedings of the International Conference on Learning Representations*, Banff, Canada, April 2014.

[17] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, October 2017.

[18] A. S. Ross and F. Doshivelez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," in *Proceedings of the National Conference on Artificial Intelligence*, pp. 1660–1669, New Orleans, LA, USA, October 2018.

[19] J. Buckman, A. Roy, C. Raffel, and I. J. Goodfellow, "Thermometer encoding: one hot way to resist adversarial examples," in *Proceedings of the International Conference on Learning Representations*, Vancouver, Canada, May 2018.

[20] R. Briandet, E. K. Kemsley, and R. H. Wilson, "Discrimination of Arabica and Robustain instant coffee by fourier transform infrared spectroscopy and chemometrics," *Journal of Agricultural and Food Chemistry*, vol. 44, no. 1, pp. 170–174, 1996.

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

[22] J. B. Kruskal and M. Wish, "Multidimensional scaling," 1978.

[23] C. Szegedy, W. Zaremba, I. Sutskever et al., "Intriguing properties of neural networks," in *Proceedings of the International Conference on Learning Representations*, Banff, Canada, April 2014.

[24] F. Tramer, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. Mcdaniel, "The space of transferable adversarial examples," *Machine Learning*, vol. 2017, 2017.

[25] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: high confidence predictions for unrecognizable images," in *Proceedings of the Computer Vision and Pattern Recognition*, pp. 427–436, Boston, MA, USA, June 2015.

[26] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the Computer Vision and Pattern Recognition*, pp. 2921–2929, Las Vegas, NV, USA, June 2016.

[27] A. Bagnall, H. A. Dau, J. Lines et al., "The UEA multivariate time series classification archive, 2018," 2018.

[28] P. O. A. R. Foreman, *Powering the Nation: Household Electricity Using Habits Revealed*, Energy Saving Trust/DECC/DEFRA, London, UK, 2012.