Time series adversarial attack and defense

Xue Zhechang

[1] Karlsruhe Institute for Technology, Germany
[2] ugupg@student.kit.edu

Abstract. Nowadays, Time series data plays an important role in many fields. With the help of deep learning based model, people can classify and predict time series data, which is useful and efficient in data mining and recovering missing data. However, deep learning models are vulnerable to adversarial attacks. Tiny perturbation added to original time series will lead to the decrease of accuracy. Thus, it's a big topic to generate attacks and find methods to defend them. Yet, researchers have developed various adversarial attacks and also seek the way to improve the robustness of models. In this work, I will introduce several attacks and the methods to defend models from adversarial attacks.

Keywords: Time series · Adversarial attack · Deep Learning.

1 Introduction

Time series is a series of data points indexed in time order. Time series data are widely used in various fields ranging from mathematical statistics, signal processing and pattern recognition to quantitative finance and weather prediction.[1] People notice the big value of analysing time series data. Deep learning models have nowadays succeeded in many real life application such as speech recognition and computer visions. Thus, they are also applied to time series data. There's two main type of time series deep learning model. One is time series classification (TSC) model, which is mainly used to classify categories of time series data. Another is time series regression (TSR) model, which can predict the future data in a proper time point.

However, deep learning models are not robust. For example, researchers modify the original time series data, where the changes are too tiny to be detectable by human. Due to these changes, the accuracy of classification or prediction declines significantly. This is so-called adversarial attack.

To the best of our knowledge, adversarial attacks were first studied by Fazle et al[2]. Their target models are 1-Nearest Neighbor Dynamic Time Warping (1-NN) DTW, a Fully Connected Network and a Fully Convolutional Network (FCN). They trained Adversarial transformation network (ATN) to attack target models and tested with University of California Riverside (UCR) datasets. Finally they proved that TSC models are vulnerable to adversarial attacks. They also proved the vulnerability of multivariate time series[4].

Gautam et al[7]. proved that TSR models are also vulnerable to adversarial

attacks. They transfered existed attacks from computer vision domain to time series domain, which are called fast gradient sign method (FGSM) and basic iterative method (BIM). They also proved the transferability of adversarial attacks to different target models.
Hassan et al[3]. introduced how FGSM and BIM works in time series domain. They used multi-dimensional scaling as the measurement to evaluate the relation between perturbation and accuracy.
Pradeep et al[8]. defined 4 kinds of adversarial attacks: untargeted, targeted, individual universal attacks. Then they introduced the method of targeted attack and universal attack and proved their effectiveness on TSC models based on ResNet. Finally, they introduced that backpropagation algorithm can help increase the robustness of time series model.
Aidong et al[9]. introduced a new method to attack TSC models with higher efficiency. They measured the importance of adversarial samples and selected some of the most important adversarial samples to modify the original data. This method can decline the perturbation of original data but increase the effectiveness. Tiny perturbation of time series data can lead to big difference in classification and prediction. However, few researches are done about detection of adversarial attacks. Mubarak et al[**?**]. introduced a method to detect whether the time series data is adversarial generated by FGSM and BIM.
Shoaib et al[6]. transferred three defensive methods from computer vision domain to time series domain: Adversarial training, TRADES and feature denoising. To prove their effectiveness, they used FGSM and Projected Gradient Descent (PGD) as white-box attacks and noise attack, boundary attack and Simple Black-box Attack (SIMBA) as black-box attacks.
Zhongguo et al[10] introduced a new defend to adversarial attacks and proved its effectiveness by experiment. They used thermometer encoding to non-linear encode original time series data and trained a encode-decode model to decode the modifyed time series data. Then they trained the deep learning model on time series with these output, which can nearly completed ignore the affect of perturbation.
In this paper, I will summarize some researches in time series domain. We will begin with adversarial attacks. On this basis, we will prove the vulnerability of time series model. Then we will show how to detect adversarial attacks from original data. Finally, we will introduce some methods to defense adversarial attacks.

2 Background

2.1 Definition

Definition 1 (Time series) Time series data can be mathematically represented as set X = $[x_1, x_2, x_3, ..., x_T]$, where T is the length of this set.

Definition 2 (Time series target) Each time series has a corresponding target series (lable) Y = $[y_1. y_2, y_3, ..., y_T]$, where T is the length of this set.

Definition 3 (Adversarial time series) Given a time series X = [x_1, x_2, x_3, ..., x_T], the adversarial time series $X^{'}$ = X + η = [$x^{'}_1$, $x^{'}_2$, $x^{'}_3$, ..., $x^{'}_T$], where η is the perturbation generated by the attacker.

Definition 4 Decision-based attack is an attack completely depends on the final decision made by target model.

2.2 Metric: Multi-Dimensional Scaling

To evaluate the effectiveness of the adversarial attacks, the researchers have developed various measurements, e.g. Relative Absolute Error (RAE), Empirical Correlation Coefficient (CORR) and Root Relative Squared Error (RSE). Here I will introduce a visible and easy-to-read method: Multi-Dimensional Scaling.
Multi-Dimensional Scaling (MDS) is a method to visualize the the distribution of adversarial samples by locating adversarial samples and original data spatially. It uses Euclidean Distance (ED) on a set of original und adversarial time series to create a similarity matrix and display the result in a 2-dimensional space. And researchers concluded a cost function of MDS called *Stress*:

$$Stress_D(X_1, X_2, ..., X_N) = \frac{\sum_{i,j}(d_ij - ||x_i - x_j||^2)^{1/2}}{\sum_{i,j} d_ij^2} \tag{1}$$

Here ist d_ij the ED between X_i and X_j and D is a set of d_ij.

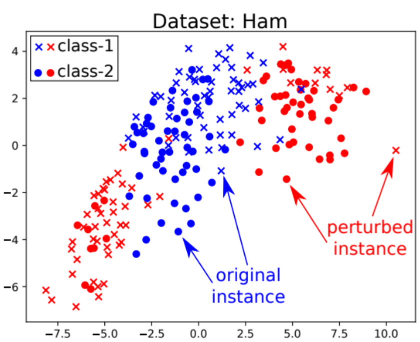

Fig. 1. MDS showing the distribution of perturbed time series on the whole test set of the Ham dataset.[3]

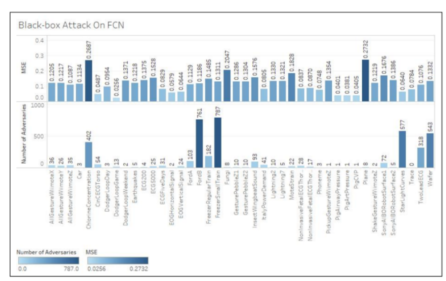

Fig. 2. Black-box attack on FCN.[2]

2.3 Vulnerability

To the best of our knowledge, Fazle et al.[2] firstly proved the vulnerability of time series model. Since TSC can be a black-box model, attackers can't always get the gradient information about the model. Thus, they find out an attack method which is suitable for both black-box model and white-box model. They trained a distilled model to mimic the target model with the input and output, and trained a gradient adversarial transformation network (GATN) to attack the distilled model. As is shown in Fig.2, they attacked a black-box model based on Fully Convolutional Network (FCN). The result shows that this attack is really effective and all the accuracy among 42 datasets decreased to less than 50%.

2.4 Transferability

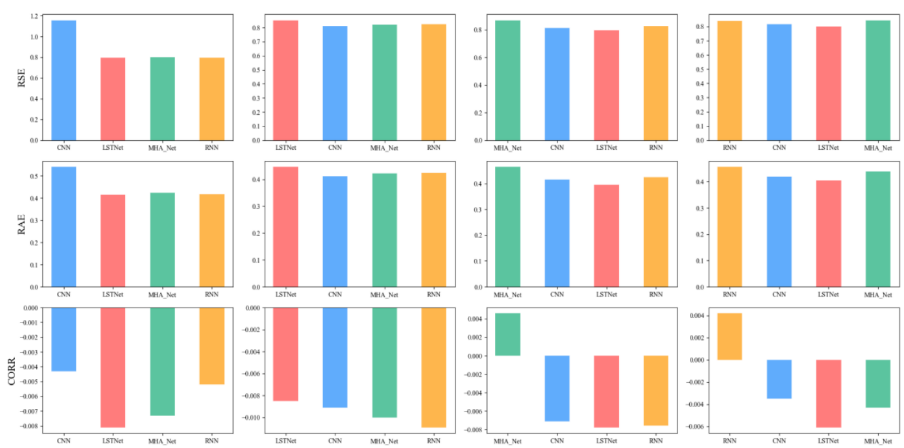

Fig. 3. Transferability validation on the Electricity Dataset[9]

Transferability is one of the important properties of adversarial examples. It means that one adversarial example, whose target model is A, can also decrease the accuracy of model B.

Aidong et al. experimented on 4 different models to prove the transferability of the adversarial examples. For instance, they set the adversarial example targeting CNN as input to other 3 models: LSTNet, MHA_Net and RNN. As is shown in Fig.3, this adversarial example is also effective on other 3 models. However, its efficency on other 3 models is less than on CNN.
Although the efficiency of adveresarial examples will be decreased after transfer, this property is an essential part of generating black-box attack.

3 Adversarial attack methods

Adversarial attacks are normally divided into two categories: White-box attacks and black-box attacks.

In white-box attacks, the attacker has access to all the information about the targeted model. Thus, attacking with gradient is a common method in deep learning. Here we will introduce two attacks based on gradient: Fast gradient sign method (FGSM) and basic iterative method (BIM).
In black-box attacks, the attacker has no information or parameter about the targeted model. Thus, it's impossible to attack the targeted model with gradient. Attackers normally use the relation between input and output of the target to find the way to attack the model.

3.1 White-box attacks

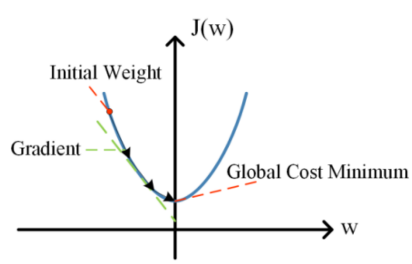

Fig. 4. Gradient descent for the solving of LSTNet model.[9]

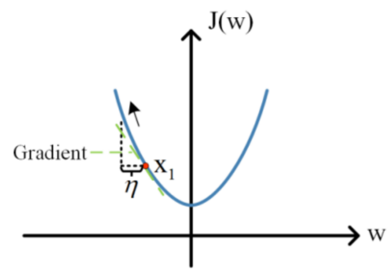

Fig. 5. Gradient-based generation of adversarial samples.[9]

Fast gradient sign method Fast gradient sign method (FGSM) was firstly used in attacking image models and then was transferred to time series field. The perturbation is generated by a one-step gradient update along the direction of gradient's sign at each timestamp (shown in Fig.4 and Fig.5).
There's two kinds of FGSM: Untargeted attack and targeted attack. Untargeted attack means this attack can misguide the model to predict any incorrect classes, while targeted attack means this attack can misguide the model to predict a specified class.
The perturbation generated by untargeted FGSM is as follows:

$$\eta = \epsilon \cdot sign(\nabla_x L(X, Y)) \tag{2}$$

Here is L the loss function. When attackers want to change it to targeted attack, they need to set Y a specified label and make L negative.

$$L_T = -L \tag{3}$$

$$\eta = \epsilon \cdot sign(\nabla_x L_T(X, Y_T)) \tag{4}$$

Basic iterative method Basic iterative method is based on FGSM attack and also known as iterative-FGSM. In this method, the one-step graident update will be iterated in smaller step sizes(shown in Algorithm 1).

Algorithm 1 Basic iterative method

Require: original time series X and its corresponding label Y
  $X^{'} \leftarrow X$
  for i = 1 to I **do**
    $\eta = \epsilon \cdot sign(\nabla_x L(X, Y))$
    $X^{'} = X + \eta$
    $X^{'} = \min\{X + \epsilon, \max\{X - \epsilon, X^{'}\}\}$
  end for
return adversarial sample $X^{'}$

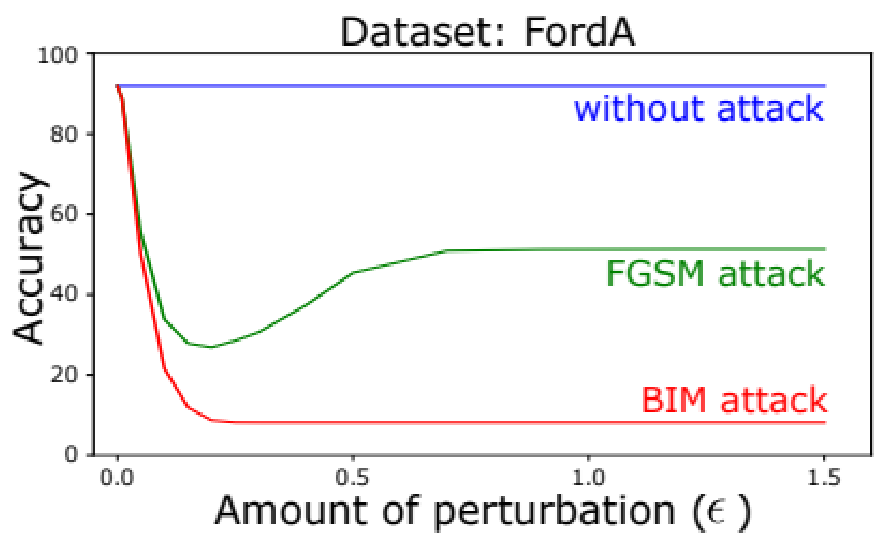

Fig. 6. Accuracy variation with respect to the amount of perturbation for FGSM and BIM attacks on FordA.[3]

Due to the iteration, the perturbation will be minimized and the adversarial attack will be closer to original time series. And as is shown in Fig.6, BIM is more effective than FGSM. However, this method costs much longer time to generate adversarial samples.

3.2 Black-box attacks

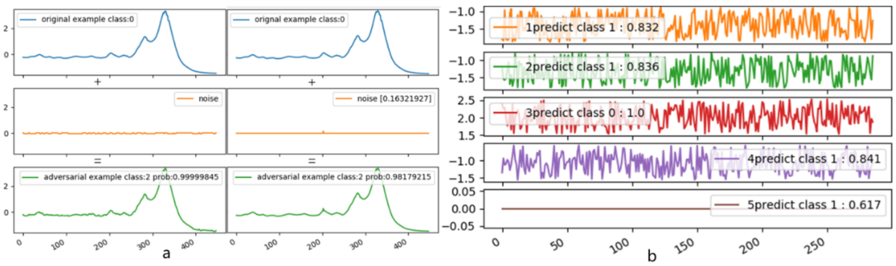

Fig. 7. The noise data can lead to the mistake of prediction with high confidence.[11]

Random noise attack As is shown in Fig.7.a, even a tiny perturbation to original data, which is unrecognizable to human beings, can lead to big mistake of prediction with high confidence. In Fig.7.b, even a zero value time series data will be classified by the model with high confidence. However, in the correct situation, these time series data should be rejected by the classifier. This shows the potential risk of the model. Thus, generating random noise and modify the original data with them can be an attack to time series models.

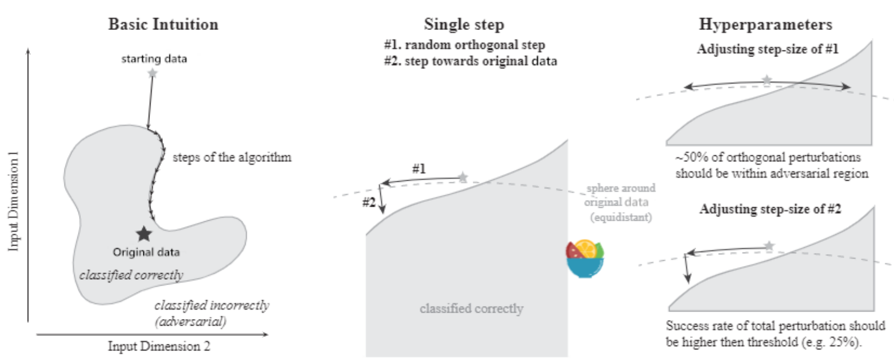

Fig. 8. The theory of boundary attack.[12]

Boundary attack Boundary attack is one of the decision-based attack. As is shown in Fig.11, the aim of this attack is to get the whole decision boundary of the label of original data.

This attack starts with a random input. This random input should not be classfied as the given label, which means it is already adversarial. The random input walks towards the original input, until it arrives at the decision boundary. At this moment, it walks orthogonal for a short distance, then starts walking towards original data again. After loop of this process, attackers will know the whole decision boundary, which is essential for black-box attack.

3.3 Optimizing

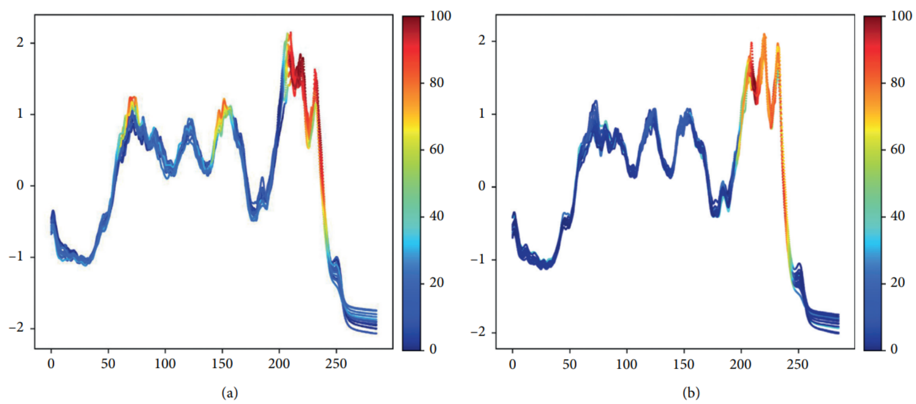

Fig. 9. Classification activation map in Coffee dataset. (a) Coffee dataset for class 0. (b) Coffee dataset for class 1.[10]

Class activation map Class activation map(CAM) is able to show the susceptible region of a time-series data. CAM can only be utilized by the models with a global average pooling(GAP) layer, which can help identify possible regions of the input.
We assume that $A_m(t)$ is the univariate time series with variable $m \in [1, M]$, and w_m^c is the weight between label c and variable m. Then CAM_c will be calculated as follows:

$$CAM_c(t) = \sum_m w_m^c A_m(t) \tag{5}$$

Fig.9 is a CAM based on Coffee dataset. It shows that red part of the map is the most susceptible region of the time-series data. The perturbations modified on these areas are most efficient for adversarial attack.

The importance of adversarial sample Aidong et al.[9] introduced a method to evaluate the importance of adversarial sample. They were inspired by feature importance ranking and assumed that different samples have different effect on model performance. The distance between y_i and $y_i^{'}$ is the critical in measuring the importance of adversarial sample. The effect of adversarial sample is proportional to the distance.

Thus, they introduced a method to optimize the adversarial attack. After generating the adversarial sample, each distance between y_i and $y_i^{'}$ will be calculated and ranked in descending order. Then a proportion P will be set to determine the count of original time seires x to be replaced. Finally, P percent of the most important adversarial samples will replace the corresponding original time series sample.
In conclusion, this method is similar to class activation map. They both select the most valuable adversarial samples instead of replacing all the original time series sample. With the help of this importance measurement or CAM, the adversarial attack will be less recognizable, but more effcient.

4 Defense

I have already introduced some adversarial attack methods and the way to optimize them. With the help of the theory of adversarial attack, it's easier to defense it. Researchers have already created some defenses against adversarial attacks or transferred some methods from computer vision field. In this section, I will introduce several methods to defense adversarial attacks.

4.1 Adversarial training

Adversarial training is one of the most widely used defense in the world. The concept of adversarial training is simple: Training the model with adversarial samples rather than the original time series. After training, the model's robustness will be improved.
The key of adversarial training is to find the parameter vector θ with high adversarial sensitivity, and then minimize it. The formula of minimizing the paramater vector is as follows[14]:

$$\theta = \arg\min E_{(x,y)\sim D}[\max_{\delta \in S} L(f(x+\delta,\theta),y)] \tag{6}$$

E is the expected value of the maximum loss change, D is the time series set with D = [x_1, x_2, x_3, ..., x_T], y is a data input and its ground truth, f is the neural network with parameter vector θ and loss function L, δ is a set of perturbation with δ = [δ_1, δ_2, δ_3, ..., δ_T] and S is the constraint set with S = $\{\delta : ||\delta||_2 \leq \epsilon\}$.

4.2 Feature Denoising

By observing feature maps of adversarial samples, Xie et al.[**?**] concluded that adversarial samples are noiser than original data. Thus, denoising the adversarial samples will add the accuracy of the model.
Denoising operators before max-pooling layers is a way to denoise. Here I will introduce Gaussian Non-Local Means (GNLM), which is the most effective selection. Its formal is as follows:

$$y_i = \frac{1}{\sum_{\forall j \in N} f(x_i,x_j)} \sum_{\forall j \in N} f(x_i,x_j) \times x_j \tag{7}$$

$$f(x_i, x_j) = e^{\frac{1}{\sqrt{d}}\theta(x_i)^t \phi(x_j)}, \theta(x_i) \in R^{64}, \phi(x_j) \in R^{64} \tag{8}$$

y_i is here the $i^t h$ output and N is the collection of all the spatial locations on feature map. $f(x_i, x_j)$ means the similarity between x_i and x_j, $\theta(x_i)$ and $\phi(x_i)$ input after two different 1×1 convolution and d is the number of channels. Researches have used this denoising operator to generate a denoising layer before every pooling layer and find out that this donoising layer can also be trained during adversarial training.

4.3 Non-linear transfer

Zhongguo et al.[10] proposed a method

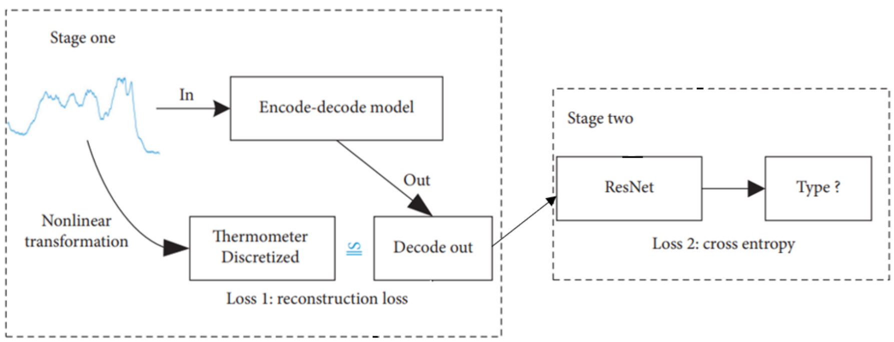

Fig. 10. The theory of boundary attack.[12]

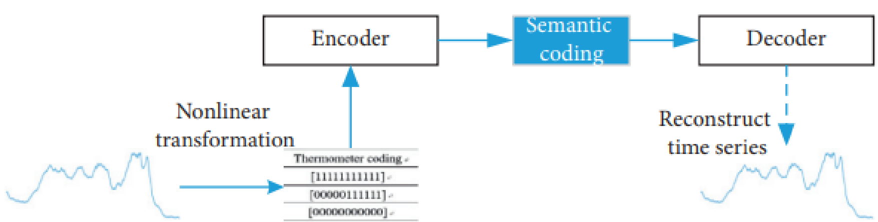

Fig. 11. The theory of boundary attack.[12]

thermometer encoding

encode-decode model

Method

5 Conclusion

References

1. Wikipedia contributors. Time series, 2022. URL: https://en.wikipedia.org/wiki/Time_series
2. Fazle Karim and Houshang Darabi, Adversarial Attacks on Time Series, 2019 URL: https://ieeexplore.ieee.org/abstract/document/9063523
3. Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar and Pierre-Alain Muller IRIMAS, Universite Haute-Alsace, Mulhouse, France, Adversarial Attacks on Deep Neural Networks for Time Series Classification, URL: https://ieeexplore.ieee.org/abstract/document/8851936
4. Samuel Harford, Fazle Karim, and Houshang Darabi, Adversarial Attacks on Multivariate Time Series, URL: https://arxiv.org/abs/2004.00410
5. Mubarak G. Abdu-Aguye, Walid Gomaa, Yasushi Makihara, Yasushi Yagi, Cyber Physical Systems Lab, Egypt Japan University of Science and Technology, Egypt, Faculty of Engineering, Alexandria University, Egypt, The Institute of Scientific and Industrial Research, Osaka University, Japan, Detecting adversarial attacks in time-series data, URL: https://ieeexplore.ieee.org/abstract/document/9053311
6. Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed, German Research Center for Artificial Intelligence (DFKI), 67663 Kaiserslautern, Germany, Benchmarking adversarial attacks and defenses for time-series data, URL: https://link.springer.com/chapter/10.1007/978-3-030-63836-8_45
7. Gautam Raj Mode and Khaza Anuarul Hoque, Department of Electrical Engineering & Computer Science, University of Missouri, Columbia, MO, USA, Adversarial Examples in Deep Learning for Multivariate Time Series Regression, URL: https://ieeexplore.ieee.org/abstract/document/9425190
8. Pradeep Rathore, Arghya Basak, Sri Harsha Nistala, Venkataramana Runkana, TCS Research, Pune, India, Untargeted, Targeted and Universal Adversarial Attacks and Defenses on Time Series, URL: https://ieeexplore.ieee.org/abstract/document/9207272
9. Aidong Xu, Xuechun Wang, Yunan Zhang, Tao Wu, Xingping Xian. Adversarial Attacks on Deep Neural Networks for Time Series Prediction, URL: https://dl.acm.org/doi/10.1145/3485314.3485316
10. Zhongguo Yang, Irshad Ahmed Abbasi, Fahad Algarni, Sikandar Ali, and Mingzhu Zhang, An IoT Time Series Data Security Model for Adversarial Attack Based on Thermometer Encoding URL: https://www.hindawi.com/journals/scn/2021/5537041/
11. Zhongguo Yang, Han Li, Mingzhu Zhang, Jingbin Wang and Chen Liu, School of Information Science and Technology, North China University of Technology, Beijing, China. A Method for Resisting Adversarial Attack on Time Series Classification Model in IoT System URL: https://link.springer.com/chapter/10.1007/978-3-030-60029-7$_5$0

12. Wieland Brendel, Jonas Rauber Matthias Bethge, Werner Reichardt Centre for Integrative Neuroscience, Eberhard Karls University Tübingen, German, DECISION-BASED ADVERSARIAL ATTACKS: RELIABLE ATTACKS AGAINST BLACK-BOX MACHINE LEARNING MODELS URL: https://openreview.net/forum?id=SyZI0GWCZ
13. Xie, C., Wu, Y., Maaten, L.v.d., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 501–509 (2019)
14. Zhiyuan Zhang, Wei Li, Ruihan Bao, Keiko Harimoto, Yunfang Wu, Xu Sun, ASAT: Adaptively Scaled Adversarial Training in Time Series URL: https://arxiv.org/abs/2108.08976