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. Deep learning models, which have already shown its effectiveness and efficiency on various real-life applications, is one of the most important tools to help human process time series data. However, they are vulnerable to adversarial attacks. Tiny perturbation added to original time series will lead to the decrease of prediction accuracy, which leads to potential risks of time series application. To improve its robustness, one effective way is to generate adversarial attacks against models, and then find methods to defend them. Yet, researchers have developed various adversarial attacks and also found some methods to improve the robustness of models. In this work, I will summarize several attacks and the methods to defend models.

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] However, due to the huge volume of time series data, it's a big task to process time series data more efficiently. Researchers looked beyond deep learning models, which have nowadays succeeded in many real life application. In comparison to other methods, the performance of deep learning models grows by the amount of data. This feature fits greatly for time series data. Using deep learning models, people can convenient classify and predict the time series. For example, it helps study the values of different variables over time. Imagining a temperature, energy consumption sensors in a machine. Time series model will use combination of temperature and energy consumption to forecast a machine failure.

In the ideal situation, perturbation of input cannot let the model make mistakes. We call this the robustness of a model. However, researchers find out that a tiny perturbation of the input may lead to the wrong output. Although the change is too tiny to be disturb human beings, it will largely misguide the model. This is so-called adversarial attack. For example, predicting the track of trajectory is an essential part of self-driving.[16]. And the prediction is based on the history of past trajectory. When the attacker add perturbation on the coordinate of the

past trajectory, the model will predict the wrong trajectory and affect the decision of self-driving. As is shown in Fig.1, the crafted history lead to the wrong prediction of the other vehicle (OV), making autonomous vehicles (AV) to stop to prevent it from the strike, which will not happen in reality.
In this paper, I will summarize some researches in time series domain. We will

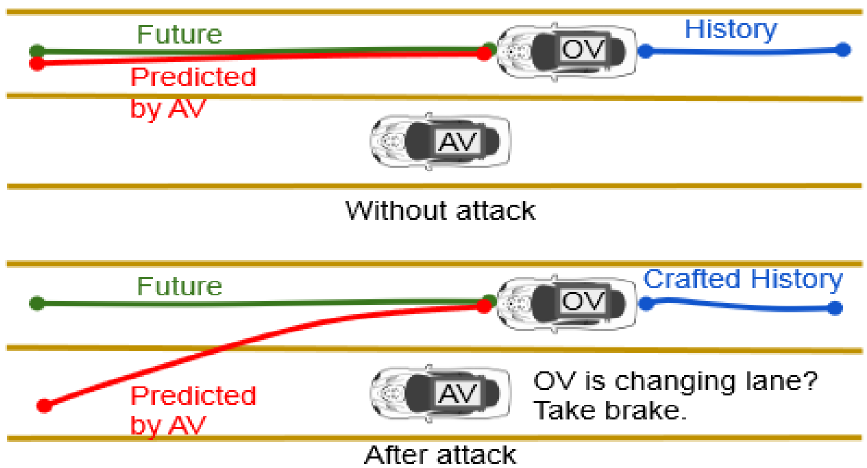

Fig. 1. An example of attack scenarios on trajectory prediction.[16]

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 $X' = X + \eta = [x'_1, x'_2, x'_3, ..., x'_T]$, where η is the perturbation generated by the attacker.

Definition 4 (White-box attack) 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.

Definition 5 (Black-box attack) 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.

Definition 6 (Decision-based attack) 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, Empirical Correlation Coefficient and Root Relative Squared Error. 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*:

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

Here ist $d_i j$ the ED between X_i and X_j and D is a set of $d_i j$.

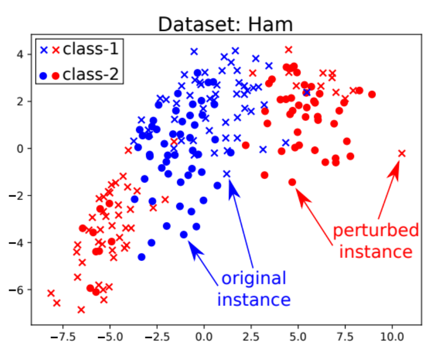

Fig. 2. MDS showing the distribution of perturbed time series on the whole test set of the Ham dataset.[3]

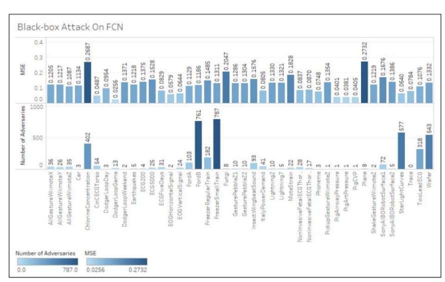

Fig. 3. Black-box attack on FCN.[2]

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.3, 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

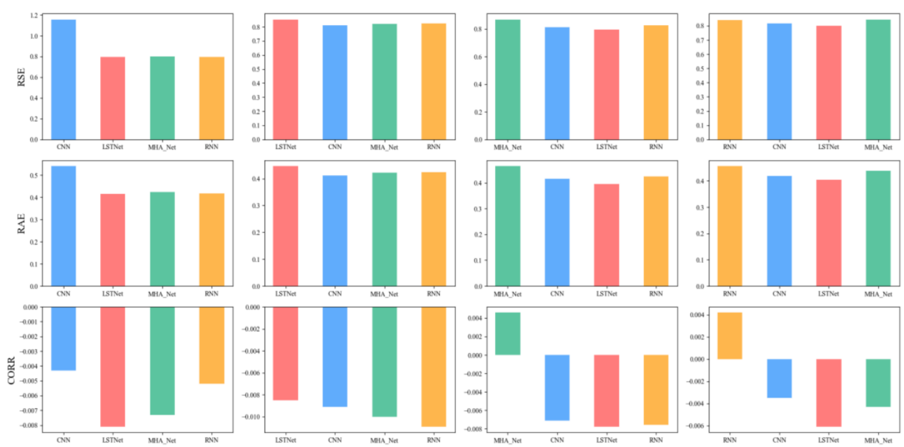

Fig. 4. Transferability validation on the Electricity Dataset[9]

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.4, 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.

3.1 White-box attacks

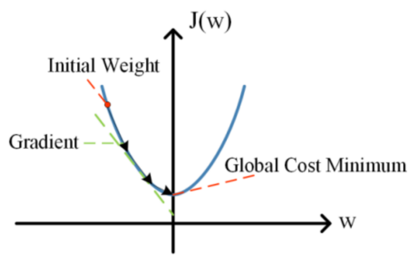

Fig. 5. Gradient descent for the solving of LSTNet model.[9]

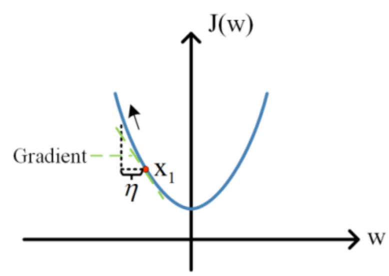

Fig. 6. Gradient-based generation of adversarial samples.[9]

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.5 and Fig.6).
The perturbation generated by FGSM is as follows:

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

where ϵ is the magnitude of the perturbation, $L(X, Y)$ is the loss corresponding to the sample X.

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).

In Algorithm 1, α is the per-step small perturbation, ϵ is the amount of maximum perturbation, and I is the number of iterations.
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.7, an experiment to compare the effectiveness of FGSM and BIM is conducted on FordA dataset, which contains 3601 training instances and another 1320 testing instances of engine noise, and aims to diagnose whether a certain symptom exists or does not

Algorithm 1 Basic iterative method

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

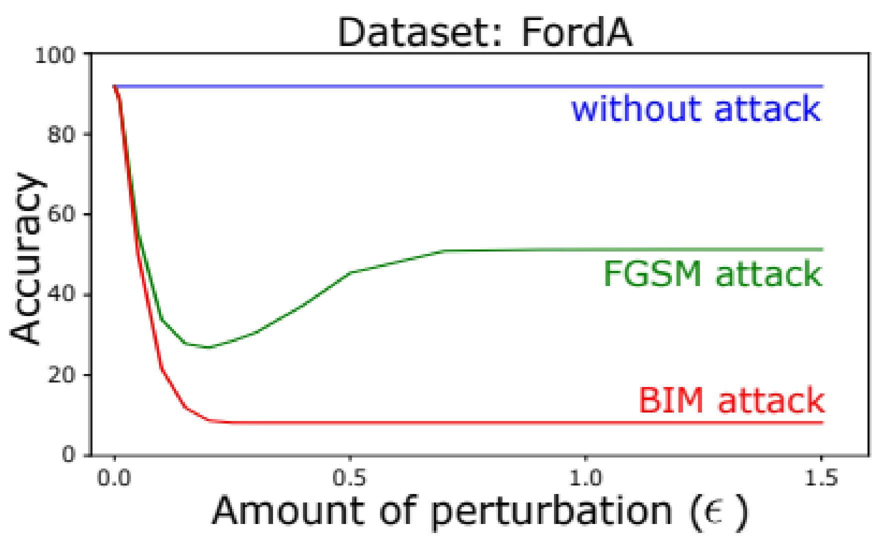

Fig. 7. Accuracy variation with respect to the amount of perturbation for FGSM and BIM attacks on FordA.[3]

exist in an automotive subsystem. We can conclude that FGSM and BIM have same performance with a small amount of pertuabation. However, BIM is more effective than FGSM with the increase of perturbation. With a large amount of perturbation, BIM can decrease almost double accuracy than FGSM.
As a result, BIM is more effective than FGSM in all situations, but it takes much more time to generate adversarial samples due to iteration.

3.2 Black-box attacks

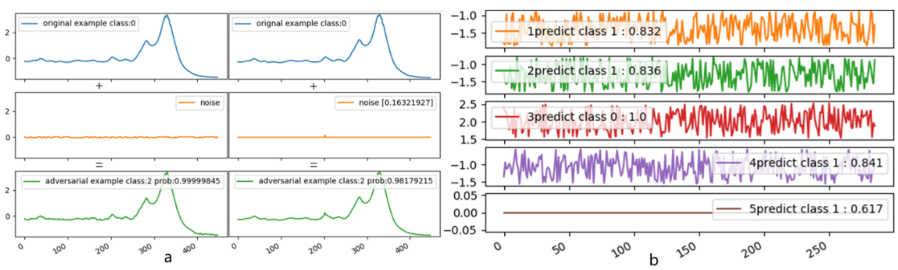

Fig. 8. The noise data can lead to the mistake of prediction with high confidence.[11]

Random noise attack As is shown in Fig.8.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.8.b, there's random noise data and corresponding confidence. It shows that 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 modifying the original data with them can misguide the decision of the model.

Boundary attack Boundary attack is one of the decision-based attack. As is shown in Fig.9, 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. Then attackers can generate the adversarial sample along the boundary, where the perturbation will be minimized.

3.3 Optimizing

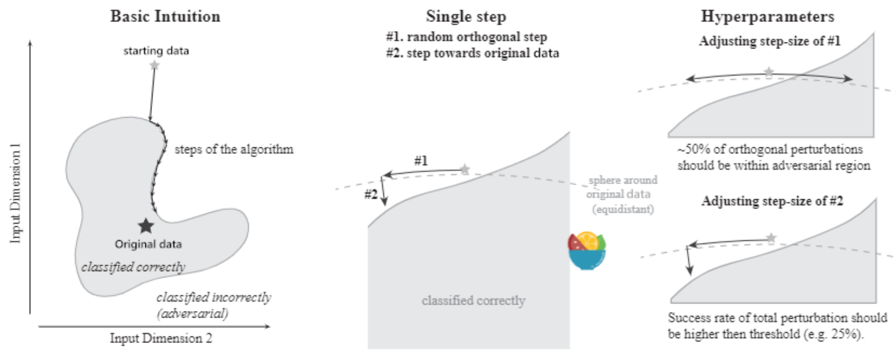

Fig. 9. The theory of boundary attack.[12]

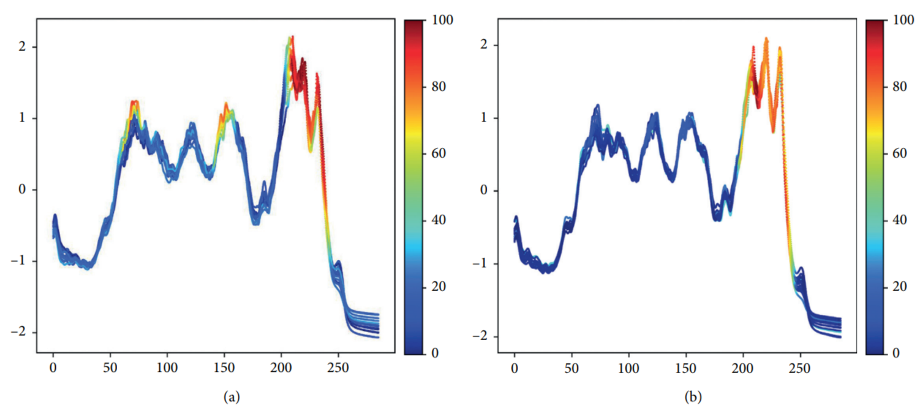

Fig. 10. Classification activation map in Coffee dataset. (a) Coffee dataset for class 0. (b) Coffee dataset for class 1.[10]

Class activation map Class activation map(CAM) is able to show the susceptible region of a time-series data. We assume that $A_m(t)$ is the multivariate time series with m variables $m \in [1, M]$, and w^c_m is the weight between label c and variable m. Then CAM_c will be calculated as follows:

$$CAM_c(t) = \sum_m w^c_m A_m(t). \tag{3}$$

Fig.10 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]:

$$\theta = \arg\min E_{(x,y)\sim D}[\max_{\delta\in S} L(f(x+\delta,\theta),y)]. \tag{4}$$

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:

$$y_i = \frac{1}{\sum_{\forall j\in N} f(x_i,x_j)} \sum_{\forall j\in N} f(x_i,x_j)\times x_j. \tag{5}$$

$$f(x_i,x_j) = e^{\frac{1}{\sqrt{d}}\theta(x_i)^t\phi(x_j)}, \theta(x_i)\in R^{64}, \phi(x_j)\in R^{64}. \tag{6}$$

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 to defend models from gradient-based attack. They trained a encode- decode model in front of time series model, which aims to reconstruct the input data to be non-linear. Then they trained time series model with this reconstructed data to avoid the perturbation of adversarial samples.
In this section, I will first introduce thermometer coding, and then introduce how this encode-decode model works.

thermometer coding Thermometer coding, which is a kind of unary coding, can transfer continuous input to discrete input. It is similar to one-hot coding to transfer a real number to a set of bits with fixed length. As is shown in Fig.11, after the transformation, the data will be non-linear. However, thermometer coding has more discretization levels. As is shown in Table 1, there's more 1 in one group, which can avoid losing information of original data.

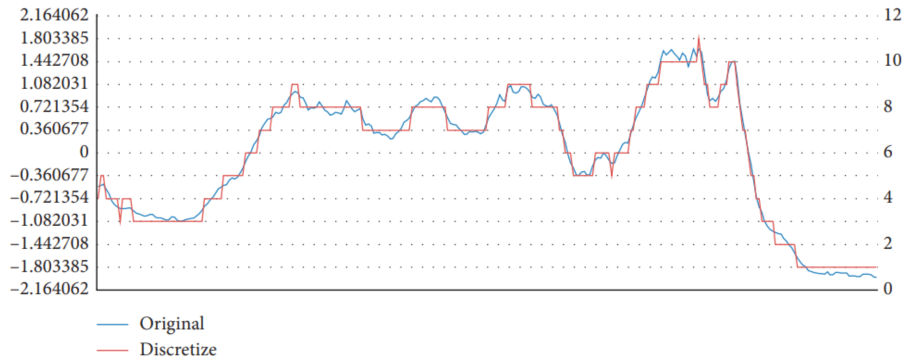

Fig. 11. Examples of mapping continuous-valued inputs to quantized inputs and thermometer codes with ten evenly spaced levels.[10]

Buckman et al. defined thermometer vector $\tau(j)_l$ for a index $j \in \{1, ..., k\}$ with 2 situations: If l $\geq j$,$\tau(j)_l = 1$, otherwise $\tau(j)_l = 0$, and then defined discretization function f as follows:

$$f_{therm}(x)_i = \tau(b(x_i)). \tag{7}$$

In this function, 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 retains the order of time series, which helps keep the shape information of the original time seires.

Real-valued	Quantized	Discretized (one-hot)	Discretized (thermometer)
0.13	0.15	[0100000000]	[0111111111]
0.66	0.65	[0000001000]	[0000001111]
0.92	0.95	[0000000001]	[0000000001]

Table 1. Example of difference between one-hot and thermometer encoding[9]

encode-decode model The input will firstly be transferred to non-linear data with thermometer encoding. The mission of encode-decode model is to reconstruct the non-linear data to a continuous time series. Since gradient-based attacks don't work on non-linear data, this model will not be affected by them. Because of the characteristic of thermometer encoding, this model can be trained to recover the original time series. As is shown in Fig.12, this model will successfully recover almost all the information of original time series. As is shown in Fig.13. After decoding, they trained time series model with decoded data. In this situation, the time seires model will not be misguided, for all the input are "purified" by encode-decode model.

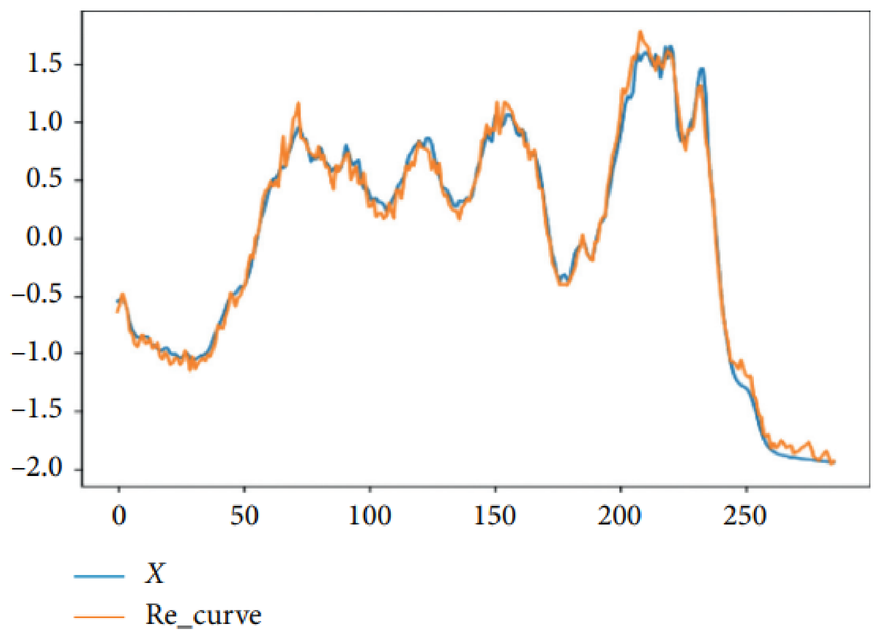

Fig. 12. The reconstruction curve from the original time series.[10]

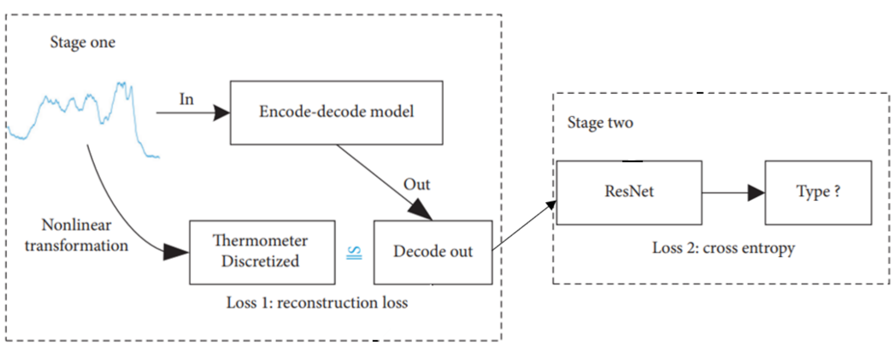

Fig. 13. The procedure of the non-linear transfer method.[10]

5 Conclusion

This work firstly introduces some properties of time series models, and then focuses on the summary of various adversarial attacks and different kinds of defends.

In future, researchers should put more emphasis on optimizing adversarial attacks. Recent researches have paid more attention on generating various adversarial attacks. However, with the improvement of robustness, to a great extent, normal adversarial samples cannot misguide models. Thus, analyzing the pattern of adversarial samples and then optimizing adversarial attacks should be the next target. There's only a few researches about the susceptible regions of adversarial samples such as CAM, and there's no application of it. To make adversarial attacks unrecognizable and effective, only modifying original data of susceptible regions is a practical method.

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_50
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
15. Jacob Buckman, Aurko Roy, Colin Raffel, Ian Goodfellow, Google Brain, Mountain View, CA, THERMOMETER ENCODING: ONE HOT WAY TO RESIST ADVERSARIAL EXAMPLES
16. Qingzhao Zhang, Shengtuo Hu, Jiachen Sun, Qi Alfred Chen, Z. Morley Mao, University of Michigan, 2University of California, Irvine On Adversarial Robustness of Trajectory Prediction for Autonomous Vehicles URL:https://arxiv.org/abs/2201.05057