# Research on adversarial disturbance based on meteorological time series data

Renjie Wang1 Haizhong Liu\*,1 Yang Zhu1

ranger\_w@163.com jiaodalhz@126.com 1466627344@qq.com

(1. School of Mathematics and Physics, Lanzhou Jiaotong University, Lanzhou, China)

## ABSTRACT

When the deep learning model is used to predict time series data, it is easy to be adversarially attacked. The time series data is sensitive to the abnormal disturbance and has strict requirements on the disturbance amount. To solve these problems, we propose to generate adversarial time series by adding disturbance terms to the original time series, and design an adversarial attack algorithm based on the importance measure(AAIM in short), which slightly perturbs the original data to improve the performance of the time series prediction model. The meteorological data of Guangzhou from 1980 to 2020 are selected as the studied time series data, which verifies that the proposed adversarial attack method not only effectively fools the target time series prediction model LSTNet, but also attacks the most advanced models based on CNN and RNN. It can be seen from the results of AAIM that the adversarial disturbance based on the first 5% importance measurement has almost the same effect as the 100% disturbance of the original time series. For the proposed AAIM, even if the disturbance is tiny, it can impact the prediction performance of the deep learning model significantly. Experimental results show that the proposed method achieves good transferability.

## Introduction

In the post industrial era of rapid development, time series data are everywhere. Millions of sensors acquire different signals every day to collect sensing data, such as power load, traffic flow, industrial monitoring and climate trends. With the increase of data volume, time series analysis has become the focus of many research and development projects in the field of data mining. The purpose of time series prediction is to predict the future value by analyzing the past observation results and analogy or extension according to the development process, direction and trend[1,2]. Time series prediction has a wide range of applications, such as predicting the recent weather conditions[3], predicting traffic congestion, predicting the future power consumption and determining the power required for energy management.

Traditionally, many parametric models have been proposed for time series prediction, such as autoregressive (AR), exponential smoothing and structured time series models. However, the feature engineering of these models is always carried out manually, and the results depend on the professional knowledge of the researchers. Recently, the data availability and computing power of machine learning methods have been continuously improved, which makes the learning of time dynamics[4] pure data-driven. Specifically, due to the impressive performance of deep learning in various applications such as speech and language processing, face recognition and object detection, many time series prediction methods based on deep neural networks have been developed. The learning mechanism can capture the dynamic correlation between variables and consider the mixing of short-term and long-term repeated patterns, so as to make it more accurate.[5]

Some studies show that the model based on deep neural network has inherent shortcomings and is vulnerable to adversarial attacks. These attacks attempt to find imperceptible disturbances and generate adversarial examples to produce incorrect outputs with high confidence[6]. That is to say, machine learning models including neural networks will misclassify examples that are only slightly different from the original data. Initially, the concept of adversarial attack was proposed to describe the instability and unreliability of deep neural network (DNN) to the imperceptible disturbance of image pixels. Subsequently, similar adversarial attack schemes against machine learning models have been proved to be effective in various application fields. For example, Sharif et al[7] attempted to attack the face recognition system by optimizing the color of sunglasses appearing in the image. Xian et al.[8] also proposed an adversarial attack method based on depth architecture to combat the link prediction method of graph data. In addition, Huang et al.[9] developed a countermeasure disturbance method to attack the target recognition task in the field of radar signal interpretation.

In view of the increasing number of time series data, the accurate results of time series prediction play a vital role in solving the problems in the real world. Therefore, it is very important to study the robustness of time series prediction model. At the same time, due to the excellent results, deep neural network has become an indispensable part of the new generation of time series analysis model. In order to explore the robustness of the time series analysis model based on deep learning, Fawaz et al.[10] used the existing iterative adversarial attack mechanism to cheat the time series classification model based on residual network. Karim et al.[11] and Harford et al.[12] also proposed an attack method based on adversarial transformation network (ATN) against the traditional time series classification model, including 1-Nearest neighbor dynamic time warping (1-NN DTW), fully connected network and full convolution network (FCN). In order to resist adversarial attacks on the time series classification model, Yang et al.[13] trained an adversarial example detector to distinguish adversarial examples from normal examples. In addition, Siddiqui et al.[14] adopted some proven adversarial defense methods tested on images and evaluated their robustness to time-series data. Although there are few adversarial attacks against time series analysis methods, the most advanced deep time series prediction models, such as the adversarial vulnerabilities of LSTNet and RNN, have not been paid enough attention. Meanwhile, unlike the unknowability of human eyes in the field of image processing, time series data are more sensitive to abnormal disturbances. Therefore, there are stricter requirements for the amount of disturbance. Therefore, in this study, we focus on the adversarial attack of time series prediction, aiming to solve the following problems: 1. Can the most advanced deep time series prediction model be attacked by this adversarial example? 2. How to generate imperceptible adversarial examples to slightly disrupt time series data?

In this study, we propose and formulate the time series prediction of adversarial attack. Here, we assume that data points play different roles in the observed patterns of time series and have disproportionate effects on the learning model. Due to the advanced performance of the long-term and short-term time series network (LSTNet), we introduce a global disturbance based adversarial time series generation algorithm for the depth time series prediction model based on LSTNet. In order to solve the challenge of slight adversarial disturbance, we propose an advanced adversarial time series generation algorithm based on importance measurement. Experiments on real-world time series data sets show that the proposed adversarial attack method achieves satisfactory performance on several advanced time series prediction models.

Organization of the paper

The rest of the paper is organized as follows. In Section 2, we list the existing counter attack methods and their characteristics. In Section 3, we put forward 4 hypotheses and established the models we need, such as LSTNet. In Section 4, according to the model we established before, the framework of time series prediction model is built, and algorithm 1 and algorithm 2 are proposed. The loss function and time complexity are measured. In Section 5, we use our meteorological time series data set to carry out experiments to verify the characteristics of the counter attack and how to improve the attack efficiency in the counter attack.

## 2. Related works

### 2.1. Deep neural network for time series prediction

Because deep learning is very powerful, time series prediction based on deep neural network has received great attention in recent years. Time series data are naturally interpreted as the sequence of input and target. Many time series prediction methods based on recurrent neural networks (RNNs) have been proposed. Specifically, Rangapuram et al.[15] proposed a new probabilistic time series prediction method, which uses RNN to parameterize a specific linear state space model. Chen et al.[16] proposed a hybrid method of RNNs based on particle swarm optimization and evolutionary algorithm for time series prediction. However, variants of RNN are limited in the long-distance dependence in learning data, so a method based on long-term and short-term memory networks (LSTM) has been developed. In addition, due to the invariance across spatial dimensions, convolutional neural networks (CNN) with multi-layer causal convolution have been proposed for time series prediction. By combining CNN and RNN, Lai et al.[17] proposed a new deep learning framework LSTNet to extract short-term local dependence patterns between variables and find long-term patterns of time series trends. Recently, attention mechanism has been applied to the improvement of time series prediction technology. For example, Ran et al.[18] proposed an LSTM based method with attention mechanism for travel time prediction. In addition, fan et al.[19] Proposed a multi span time series prediction method with temporal attention mechanism to capture patterns in historical data. In this study, instead of considering the time series prediction method, the vulnerability of the time series prediction model based on deep learning is explored through adversarial attacks.

### 2.2. Deep learning and adversarial attack

Due to the wide application of artificial intelligence systems, the security of deep learning has become more and more important, which has attracted the attention of many researchers and practitioners. Szegedy et al.[20] took the lead in exploring the stability of neural networks and revealed their vulnerability to imperceptible disturbances. In addition, Goodfellow et al.[21] tried to explain this phenomenon, thinking that neural networks are susceptible to adversarial disturbances, mainly because they are linear. Since then, many people have devoted themselves to exploring the vulnerability of various deep learning models, such as CNN, LSTM and reinforcement learning (RL). At present, the main application of adversarial attack is in the field of image processing. Eykholt et al.[22] proposed robust physical performances to generate adversarial examples of road signs to ensure that the object recognition system incorrectly recognizes such signs. In addition, Sharif et al.[7] attacked the face recognition model at both the digital and physical levels. In order to deceive the semantic segmentation and object detection models, Xie et al.[23] generated confrontational perturbations to generate incorrect predictions for all output tags. In addition, due to the importance of graph mining and text analysis, there are also adversarial examples for graph structure and text data model. Different from the above research, we focus on adversarial attacks against time series prediction models.

### 2.3. Adversarial attack against time series analysis

Fawaz et al.[10] considered the vulnerability of the deep learning model to resist time series instances, and used the existing iterative adversarial attack mechanism to add imperceptible noise to the original time series, thus reducing the accuracy of the classification model. However, there are some differences between attacking time series models and attacking traditional image classifiers, and the sensitivity of time series data to adversarial disturbances has not been fully considered. Karim et al.[11] proposed to attack various time series classification models using adversarial transformation network (ATN). Harford et al.[12] also proposed to transform the existing ATN on the refined model to attack various multivariate time series classification models. However, such research mainly focuses on the robustness of traditional time series classification models(i.e. 1-NN DTW, FCN), and fully connected network, rather than the most advanced deep time series prediction models, such as LSTNet and RNN. In addition, Yang et al.[13] and Siddiqui et al.[14] studied the defense methods of traditional adversarial attack methods against time-series data. However, the features of time series data have not been fully discussed. Different from the above studies, this study aims to prospectively explore adversarial attacks on the most advanced depth time series prediction models and generate imperceptible adversarial examples to slightly disrupt the time series data.

## 3. Modeling

### 3.1 Hypothetical model

**Hypothesis 1** (Time Series). The time series , is an ordered set of real values, where , is the eigen dimension, and t is the length of X.

**Hypothesis 2** (Time Series Prediction). Given a time series , the time series prediction task is to predict according to , where W is the window size and H is the fixed horizontal plane before the current time point. We express the corresponding true values as . In most cases, the range of prediction tasks is selected according to the requirements of environmental settings.

**Hypothesis 3** (Adversarial Time Series). Given a time series , the attacker generates adversarial perturbations and construct the adversarial time series . Therefore, for the adversarial time series ， The performance of the time series prediction model is obviously worse than the original data.

**Hypothesis 4** (time series prediction of adversarial attack). The accuracy of the prediction model decreases with the increase of the disturbance data or the change of the data distribution. Considering the cost of data operation and the requirement that the increased disturbance is imperceptible, the amount of data operation should be reduced and the distance between the adversarial time series and the original sequence should be reduced as much as possible. Therefore, the time series prediction of adversarial attack can be expressed as a constrained optimization problem.

### 3.2 LSTNet model

LSTNet is a deep learning model for time series prediction[17]. Its overall architecture includes a convolution layer, a recursive layer, a recursive jump layer and a complete connection layer. Convolution layer extracts local features for better performance. The recursive layer finds complex long-term dependencies in the time series to obtain the global pattern, thereby improving the prediction performance. A new recursive structure, called recursion jump, is added to find extremely long-term dependency patterns and make optimization easier by taking advantage of the periodicity of time series. LSTNet includes an attention mechanism to reduce the requirement for a predetermined number of skipped hidden units.

Two variants of RNN model, GRU[24] and LSTM[25], and LSTNet, can make the prediction more accurate by obtaining long-term and short-term models. However, the gradients of GRU and LSTM disappear, which means that the weights and offsets in the network will not be effectively updated in each training step, resulting in the inaccuracy of the whole model. Therefore, LSTNet is more robust than GRU and LSTM.

### 3.3 Other models

(1) Convolutional neural network(CNN). CNN was originally developed to solve the problem of computer vision; However, recent studies have shown that they can also be well used in sequence prediction. As shown in Figure 1, the CNN model mainly includes a convolution layer, a pooling layer and a full connection layer. The convolution layer automatically extracts the features through the convolution kernel, and the pooling layer subsamples the extracted features to condense the feature matrix, while retaining the key information more useful for the final prediction. The full connection layer is used to calculate the data processed by the convolution layer and the pooling layer and obtain the final prediction result. The output results of CNN model are as follows:

where ReLU is the activation function, , W is the weight matrix, and X is the input data.

(2) Recurrent neural network (RNN): RNN was originally used in the field of natural language processing to model text data. Text data has contextual relevance in time and space. Accordingly, text data has temporal order. RNNs can capture the dependencies between samples in time series. In other words, the RNN has a recursive connection in the hidden state, and the loop constraint ensures that the sequence information in the input data is captured. Which is shown in Fig. 2. Therefore, RNN can obtain long-term macro information. The prediction of RNN model at time t can be expressed as:

where represents the output of the hidden layer at time t, is the activation function of the hidden layer, and is the activation function of the output layer.

## 4. Gradient-based generation of adversarial time series

### 4.1. Framework

The adversarial attack of time series prediction aims to generate specific adversarial time series to fool time series prediction methods. Specifically, given the time signal in a period of time, the time series prediction model can accurately predict the direction of the future trend. However, the attack based method can generate adversarial disturbance to disturb the original time series and cheat the prediction method. The adversarial time series and the original time series should be as close as possible. In the present study, we tried to minimize the distance between two sequences by adjusting the perturbation. Adversarial attack usually includes three parts: adversarial time series generator, adversarial attack and transferable attack. The details will be discussed in detail below.

Original time series

Convolution layer

Pool

Full-connected

Output

Figure 1 Structure of convolutional neural network

…

Unfold

…

Figure 2 structure diagram of cyclic neural network

* **Generating adversarial time series.** The goal of generating adversarial time series is to generate imperceptible but effective adversarial disturbances in the original time series for the time series prediction model. Assuming that the attacker knows the loss function of the time series prediction method, the attacker can obtain the gradient information by the partial derivative of the loss value.
* **Time series prediction of adversarial attacks.** The adversarial attacks implemented against time series cannot accurately predict the future value. Considering the cost of data manipulation and the imperceptibility of adversarial-perturbance, the target of enemy attack is to destroy the accuracy of time series prediction method as much as possible.
* **Transferable attack.** Many advanced deep neural networks have been applied to time series prediction. If the adversarial time series generated for a specific prediction model is effective for the model itself, and other time series prediction methods fail, the process is called "transferable adversarial attack". In this paper, we analyze the possibility of transferable adversarial attack and conduct a comprehensive experiment to verify the transferability of this method.

Figure 3 framework of adversarial time series prediction

### 4.2.  LSTNet model for time series prediction

The nonlinear characteristics of convolution layer and recursion layer in deep neural network model make the output size of neural network model insensitive to the input size. However, the time series in the real world often change in an aperiodic mode, which significantly reduces the prediction accuracy of the neural network model. In order to solve this problem, LSTNet model [26] decomposes the final prediction into nonlinear and linear components. The nonlinear part is solved by neural network and the linear part is solved by autoregressive (AR) model. The outputs of the neural network component and the AR component are accumulated to obtain the final LSTNet prediction. Specifically, for the time series . The AR model is predicted as follows:

 (6)

where: represents the predicted value of AR model at time t, q is the window size, and W and B are the model coefficients. In addition, the nonlinear component is calculated as follows:

 (7)

where, is the predicted value of the neural network at time t, is the weight matrix of the recursive layer, is the output of the recursive layer at time t, P is the number of hidden states of the recursive skip layer, is the weight matrix of the th hidden state, B represents the bias unit, and is the output of the th hidden state. The outputs of the neural network and the AR component are accumulated to obtain a final prediction,

 (8)

The L1-loss objective function of LSTNet is defined as:

 (9)

Where: is the true value at time t and W is the parameter of the model.

### 4.3 Generating adversarial time series

In order to obtain the generalization ability, the LSTNet model is trained by using the stochastic gradient descent optimization strategy, which uses the gradient to continuously update the parameters to make the loss function as small as possible. This process is repeated until convergence and the final value of the parameter is learned. Here, the objective function of LSTNet in equation (9) is expressed as , and the parameters of the model are initialized randomly. Then the parameter value is updated in each iteration and finally converges to the optimal solution. When training the model, the minimum value of the objective function is found along the opposite direction of the gradient. That is, the gradient update is used to change the value of the model parameter W, thereby minimizing the overall cost.

In order to attack the LSTNet model, the attacker needs to generate adversarial perturbations η And construct the adversarial time series ，. The formula is as follows:

 (10)

where is the true value corresponding to the time series X, is corresponding to the adversarial time series prediction result of X̂. ， and are the prediction results of nonlinear and linear components, respectively. The norm represents the matrix norm, which can be equal to 2 or ∞. This constraint limits the amount of disturbance, making the distance between and X as small as possible.

According to the gradient based model attack strategy, we can generate the adversarial time series X̂ according to the gradient information of the mode. In other words, if we want to attack this model, we can calculate the gradient of the loss function relative to the input X and calculate the gradient according to the gradient sign (e.g., the sign function . Given the learned time series prediction model, for a specific time series X, the gradient direction with the fastest growth of the loss function can be generated. Then, according to Fig. 4, based on the disturbance amplitude ε, generate adversarial disturbance η and adversarial time series X̂, so as to cheat the target model. Finally, according to the above discussion, the detailed information of generating adversarial time series is summarized in algorithm 1.

|  |
| --- |
| **Algorithm 1:** generation of adversarial time series |
| **Input:** original time series X, original time series label Y, disturbance term ε，time series prediction model f.**Output:** adversarial time series ;1: Train the input data X to obtain the learned time series prediction model ;2: According to , calculate the gradient ;3: Generate disturbance according to the sign of gradient ;4: Constructing adversarial time series based on perturbation , ;5: Return adversarial time series . |

Figure 4. Generation of adversarial time series based on gradient

### 4.4 Adversarial attack under importance measure

According to algorithm 1, the time series prediction model can be attacked based on the time series obtained by the global disturbance. However, despite the maximum disturbance ε The number of is small, but the attack in algorithm 1 will change every point of the original data, which will not ensure the imperceptibility of the disturbance. Here, we assume that the data points in the time series data play different roles in the potential pattern, and some data points have a random impact on the regularity of the data, and then we can disturb the time series data based on a limited number of data points. At the same time, Samek et al.[27] proposed the perturbation curve area (AOPC) method to quantify the importance of features to model performance. Specifically, the feature sequence is obtained by arranging the features in descending order of importance, and then, the most relevant first rule is used to perturb the features sequentially. The recursive formula is defined as follows:

 (11)

where , function g represents the perturbation of feature . AOPC is defined as follows:

 (12)

where represents the predicted value of the interference sample, is the mean of all data. Therefore, perturbations of more important features will produce larger aopc values.

Based on the above discussion, we believe that by perturbing important data points and minimizing the difference between the perturbed time series and the original time series, the model can be effectively deceived. In order to find the important data points of adversarial attacks, this paper proposes an attack method based on importance measurement. Algorithm 2 gives the details of this method. As shown in algorithm 2, and to measure the importance of data points in the time series. The greater the distance, the greater the contribution of I to time series prediction. Finally, based on the disturbance ratio p, the most important data point is selected for adversarial attack.

|  |
| --- |
| **Algorithm 2:** counter attack under importance measure |
| **Input**: original time series X, original time series label y, adversarial time series , time series prediction model f, disturbance ratio P;**Output**: adversarial time series ;1: Construct iteratively for each data point to estimate its importance, X\_ T represents the original data point at time t;2: When t varies between [1, t]3: Calculate the importance of the data point at time t ;4: End5: Arrange the data points in descending order according to and select the most important P% data points6: Perturb the most important P% data points in the original time series according to so as to obtain 7: Return adversarial time series X ̂'. |

### 4.5 Loss function

In this section, we mainly discuss how to select the loss function. Normally, -loss function and -loss functions can be used as loss functions. For noise points, the -loss function will square the error, so the calculated error value is sensitive to the noise in the data set. In contrast, the -loss function is robust to noise and is usually not affected by noise. In this study, we used -loss function and -loss function is used to analyze the robustness of the prediction model to explore the advantages and disadvantages of the proposed adversarial attack methods

 (13)

 (14)

where represents the target value and represents the predicted value of the model.

### 4.6 Time complexity analysis

The most time-consuming part of algorithm 1 is the gradient calculation in step 2. For the time series of length T, the gradient calculation adopts , and the complexity of generating adversarial disturbance can be ignored. Therefore, the total time complexity of algorithm 1 is . In algorithm 2, the time cost mainly comes from obtaining the prediction results of the adversarial time series in step 1, calculating the importance of all data points, ranking them according to the importance scores in steps 2 to 5, and disturbing the most important time points in step 6. For the adversarial time series corresponding to all data points, the estimated cost of the importance of each data point to the adversarial attack is . The complexity of the sorting operation in step 5 is , and the disturbance operation cost in step 7 is . Therefore, the cost of algorithm 2 is close to .

## 5. Experiment

In this section, we will conduct experiments on three depth time series prediction models, and use four evaluation indicators and daily meteorological data of Huangpu District, Guangzhou City, Guangdong Province to verify the performance of the proposed adversarial attack method

### 5.1. Experimental conditions

#### 5.1.1 Dataset

This study uses the daily meteorological data of Huangpu District, Guangzhou City, Guangdong Province as the time series data set for evaluation. The data set can be divided into training set, verification set and test set, with the proportions of 0.6, 0.2 and 0.2 respectively.

The original data set is the daily meteorological data of Huangpu District, Guangzhou City, Guangdong Province from January 1, 1980 to December 31, 2020, with a total of 14976 sets of data. The data indicators include the local average surface temperature, the maximum and minimum surface temperature of the day, the maximum wind speed, the rainfall in each time period, the local maximum and minimum temperature, the average air pressure, the average relative humidity and other indicators. Among them, we selected the indicator "rainfall" from many indicators for prediction, mainly because it is quantitative data.

#### 5.1.2 Evaluation indicators

In the selection of evaluation indicators, relative square root error (RSE), relative absolute error (RAE) and empirical correlation coefficient (CORR) are used for performance evaluation. In addition, for simplicity, the mean square error (MSE) is used to directly measure the impact of a single point disturbance on the performance of the model. These indicators are the most commonly used accuracy measures of the difference between the predicted value of the model and the actual observation value. The values of RSE, RAE and MSE are all non-negative, that is, within the range of [0, +∞). When the predicted value and the real value are completely consistent, they are equal to zero, which is an ideal model. The larger the error, the larger the value. The value range of CORR is between - 1.0 and 1.0. The value -1.0 of correlation coefficient indicates complete negative correlation, while the value of 1.0 indicates complete positive correlation. A correlation of 0.0 indicates that there is no linear relationship between the movements of the two variables. The purpose of adversarial attack is to deceive the prediction model and make it produce inaccurate results, which means that effective attack methods should lead to large error values and correlation values close to zero. Specifically, the evaluation indicators are defined as follows.

(1) The calculation formula of relative square error (RSE) is:

(2) The calculation formula of relative absolute error (RAE) is:

(3) The calculation formula of CORR is:

where and are the real value and the predicted value, respectively.

(4) The calculation formula of MSE is:

The purpose of the adversarial attack method is to fool the prediction model while minimizing the interference. In order to quantify the number of adversarial perturbance, Frobenius norm (F-norm for short, expressed as ) is used to quantify the distance between the adversarial time series and the original time series.

The formula of F-norm is as follows:

 (19)

where indicates the adversarial time series, and X indicates the original time series.

#### 5.1.3 Parameter setting

Parameter ε in algorithm 1 represents the total amount of disturbance in the counter time series. If the parameter is set to a large value, the antagonistic disturbance will significantly reduce the performance of the time series prediction model. However, due to the existence of a large number of antagonistic interferences, even if the purpose of fooling the prediction model is achieved, they can be easily identified and eliminated as noise. Therefore, the purpose of the adversarial attack method is always to attack the target model with imperceptible disturbances. Therefore, ε The value of should not be too large. Here, in order to evaluate the performance of the proposed adversarial attack method, we will ε 0.05, 0.10, 0.15 and 0.20, respectively.

### 5.2 The features of adversarial attack

#### 5.2.1 Effectiveness

Table 1 and table 2 respectively show the -Loss function and -Loss function according to LSTNet that algorithm 1 (i.e., the above-mentioned adversarial time series generator, ATSG) is set as a adversarial attack method. When ε= 0.00, it can be seen that the prediction performance of LSTNet on the original time series is better, which is reflected in small error and high correlation coefficient. With the increasing of ε, it can be seen that the error of the prediction value increases and the correlation coefficient decreases.

That is, the prediction on time series of LSTNet will be worsen with the increase of perturbation ε. In addition, as can be seen from Tables 1 and 2, the performance difference between LSTNet models adopting -Loss and -Loss is small, and at the same time, -Loss has more robustness to anomalies generated in time series data. Therefore, the subsequent experiments in this paper uses -Loss to evaluate the countermeasures.

|  |  |  |  |
| --- | --- | --- | --- |
| Indicators | RSE | RAE | CORR |
|  | 0.00 | 0.102 | 0.058 | 0.871 |
| 0.05 | 0.809 | 0.404 | 0.003 |
| 0.10 | 0.850 | 0.446 | -0.008 |
| 0.15 | 0.882 | 0.491 | 0.002 |
| 0.20 | 0.958 | 0.556 | 0.006 |

Table 1. -Loss of adversarial attack towards LSTNet

|  |  |  |  |
| --- | --- | --- | --- |
| Indicators  | RSE | RAE | CORR |
|  | 0.000 | 0.101 | 0.059 | 0.875 |
| 0.050 | 0.806 | 0.406 | 0.001 |
| 0.100 | 0.874 | 0.466 | -0.008 |
| 0.150 | 0.942 | 0.541 | 0.001 |
| 0.200 | 1.057 | 0.642 | 0.002 |

Table 2. -Loss of adversarial attack towards LSTNet

#### 5.2.2 Applicability

The above proposed adversarial attack method ATSG is predicted on three prediction models, RNN, CNN and LSTNet, to evaluate its applicability. Table 3 is the correlation coefficient between the true value and the predicted value under the adversarial attack. The table shows the CORR value between the predicted result based on the adversarial time series and the result based on the real time series in the three prediction models. When perturbation ε= 0.00, the time series data are not disturbed, and the error between the prediction result and the true value is large. When perturbation ε becomes larger, the data is disturbed and the correlation decreases.

|  |  |  |  |
| --- | --- | --- | --- |
| Models | RNN | CNN | LSTNet |
|  | 0.00 | 0.900 | 0.911 | 0.871 |
| 0.05 | 0.007 | 0.005 | 0.003 |
| 0.10 | 0.004 | -0.004 | -0.008 |
| 0.15 | 0.005 | -0.002 | 0.002 |
| 0.20 | 0.005 | -0.005 | 0.006 |

Table 3 correlation coefficient (CORR) between true value and predicted value under adversarial attack

Based on the above assertions, table 4 shows the results with different perturbations ε Fig. 5 is a corresponding histogram. For the above three prediction models, the error value of the prediction method will change with the disturbance ε This is in line with our earlier statement on "effectiveness", indicating that the proposed ATSG method is applicable to CNN, RNN and LSTNet.

|  |  |  |  |
| --- | --- | --- | --- |
|  | RSE |  | RAE |
| RNN | CNN | LSTNet |  | RNN | CNN | LSTNet |
| 0.00 | 0.0408 | 0.0411 | 0.0417 |  | 0.1020 | 0.1030 | 0.1010 |
| 0.05 | 0.4028 | 0.4137 | 0.4019 |  | 0.7570 | 0.8550 | 0.7560 |
| 0.10 | 0.4244 | 0.5218 | 0.4196 |  | 0.8110 | 1.1320 | 0.8490 |
| 0.15 | 0.4670 | 0.6604 | 0.4700 |  | 0.8500 | 1.1510 | 0.8510 |
| 0.20 | 0.5250 | 0.8239 | 0.5253 |  | 0.9140 | 1.8960 | 1.0740 |

Table 4. the RSE and RAE value of three prediction models under different ε



Figure 5(a). The bar chart of RSE value of 3 models under different ε



Figure 5(b). The histogram of RSE value of 3 models under different ε

Except for disturbance ε, F-norm can also be used to quantify the distance between the adversarial time series and the original time series. Table 5 shows the prediction accuracy of the model obtained under different f-norms. From left to right, it is RNN, CNN and LSTNet. From top to bottom, it is the obtained RSE value, RAE value and CORR value. The corresponding trend chart is shown in Figure 6. The horizontal axis is the value of f-norms (varying from 0 to 1), and the vertical axis is RSE and RAE. With the increase of F-norm, the error value of the prediction model increases (shown as the gradual increase of RSE and RAE), and the correlation between the prediction value and the true value weakens (shown as the gradual decrease of the correlation coefficient CORR). According to the above experimental results, we can conclude that ATSG is effective for all four prediction models, thus achieving good applicability.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | F-norm | RSE | RAE | CORR |
| RNN | 0.000 | 0.083 | 0.342 | 0.902 |
| 0.771 | 0.788 | 0.401 | 0.001 |
| 0.829 | 0.803 | 0.457 | 0.003 |
| 0.912 | 0.832 | 0.490 | 0.005 |
| 1.000 | 0.904 | 0.517 | 0.010 |
| CNN | 0.000 | 0.084 | 0.342 | 0.898 |
| 0.819 | 0.897 | 0.440 | 0.011 |
| 0.842 | 1.167 | 0.558 | 0.002 |
| 0.873 | 1.504 | 0.703 | 0.011 |
| 1.000 | 1.948 | 0.867 | 0.012 |
| LSTNet | 0.000 | 0.083 | 0.342 | 0.900 |
| 0.848 | 0.789 | 0.398 | 0.009 |
| 0.899 | 0.848 | 0.452 | 0.011 |
| 0.949 | 0.906 | 0.506 | 0.008 |
| 1.000 | 0.965 | 0.560 | 0.001 |

Table 5. RSE, RAE and CORR values of model prediction accuracy under different F-norms



Fig.6(a) The RSE value of model prediction accuracy under different F-norms



Fig.6(b) The RAE value of model prediction accuracy under different F-norms



Fig.6(c) CORR values of actual and predicted values in the model obtained under different F-norms

#### 5.2.3 Transferability

The transferability attack will generate adversarial time series for a specific time series prediction model, but it will also fool other prediction models. On the climate dataset, we confirmed the transferability of the proposed adversarial attack method ATSG in three time series prediction models. Specifically, in order to evaluate the performance of ATSG against transferability attacks, table 6 (a) gives out ε= 0.00 and ε= 0.10. Next, table 6 (b) shows the results of the transferable attack, in which one model generates adversarial time series, and then uses the adversarial time series as input to other models. Considering the experimental results in tables 6 (a) and 6 (b), we can see that adversarial attacks can shift between models. That is, even if the specific target model is unknown, the confrontation time series generated for other prediction models are effective for the target prediction model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Indicators | RSE | RAE |
|  | RNN | 0.097 | 0.055 |
| CNN | D.099 | 0.053 |
| LSTNet | 0.102 | 0.058 |
|  | RNN | 0.840 | 0.455 |
| CNN | 1.154 | 0.540 |
| LSTNet | 0.850 | 0.446 |

Table 6(a). the adversarial attack result at ε= 0.00 and ε=0.10

|  |  |  |
| --- | --- | --- |
|  | RSE | RAE |
| RNN | CNN | 0.817 | 0.419 |
| LSTNet | 0.799 | 0.403 |
| CNN | RNN | 0.795 | 0.418 |
| LSTNet | 0.797 | 0.414 |
| LSTNet | RNN | 0.824 | 0.423 |
| CNN | 0.810 | 0.412 |

Table 6(b). The transferability attack result at

In order to observe the effect of the transferable attack more directly, we give the relevant experimental results in Table 7 and Figure 7.

|  |  |  |
| --- | --- | --- |
|  | RSE | RAE |
| RNN | RNN\_0.0 | 0.1230 | 0.0625 |
| RNN\_0.1 | 0.8438 | 0.4635 |
| CNN\_0.1 | 0.8110 | 0.4271 |
| LSTNet\_0.1 | 0.8130 | 0.4290 |
| CNN | CNN\_0.0 | 0.1220 | 0.0625 |
| CNN\_0.1 | 1.1280 | 0.5621 |
| RNN\_0.1 | 0.8128 | 0.4372 |
| LSTNet\_0.1 | 0.8124 | 0.4271 |
| LSTNet | LSTNet\_0.0 | 0.1042 | 0.0627 |
| LSTNet\_0.1 | 0.8385 | 0.4479 |
| RNN\_0.1 | 0.8385 | 0.4479 |
| CNN\_0.1 | 0.7917 | 0.4063 |

Table 7. The result verification of transferability attack results



Fig.7(a) The RNN results under different attack modes



Fig.7(b) The CNN results under different attack modes



Fig.7(c) The LSTNet results under different attack modes

Figs. 7 (a) - (c) respectively show the performance of the prediction models RNN, CNN and LSTNet under various attack methods. In Fig.7 (a), the blue part indicates the RSE and RAE values of the use of RNN\_0.0 as the attack method. That is, when ε= 0.0 and there is no adversarial disturbance, the RNN is the target model, and the adversarial time series will be generated for the RNN model. The orange part indicates the RSE and RAE values of the use of RNN\_0.1 as the attack method. In other words, when ε= 0.1, that is, in the event of an adversarial attack, the RNN, as the target model, will generate adversarial time series for the RNN model. Similarly, the gray and yellow parts respectively indicate the value of RSE and RAE of the use of CNN\_0.1 and lstnet\_0.1 as the attack method. In other words, when ε= 0.1, RNN will generate adversarial time series for CNN and LSTNet. It can be seen from Fig. 7 that for all prediction models, they are generally much higher than those in the absence of adversarial interference (i.e ε= 0.0). This shows that the adversarial interference to other prediction models will also affect the inherent mode of the input data, thus affecting the performance of the target model.

#### 5.2.4 Preliminary summary on features

Effectiveness, applicability and transferability are the three main features of the adversarial attack model.

The main purpose of effectiveness is to introduce a perturbation term ε. By introducing the perturbation term ε, we can know when ε increases, the error index RSE and RAE increase as well, and the correlation coefficient CORR decreases by its increase. When ε= 0, RSE and RAE are decreased to the lowest value, and the correlation degree CORR between the actual value and the predicted value is pushed to a high level.

In the discussion of applicability, we: 1. judge whether the law of the above-mentioned perturbation term ε is applicable to other models; 2. see if there are other statistics to replace ε but not destroy the above-mentioned laws.

1. We introduced three prediction models, RNN, CNN and LSTNet, and predicted them to evaluate their applicability. For the three evaluation indicators mentioned above, RSE, RAE and CORR, the results of CORR are reflected in Table 3, and the results of RSE and Rae are reflected in Table 4 and figure 5. RSE, RAE and CORR all meet the above laws;

2. By introducing F-norm, we can measure the distance between matrices, and it is also applicable to quantify the distance between the adversarial time series and the original time series (that is, how much disturbance is added)

In the first two features, disturbance terms are added artificially, and transferability replaces the subject of adding disturbance terms with the model itself, that is, RNN, CNN and LSTNet models, which add disturbance term ε to each other. By comparing the RSE and RAE obtained from the experiment, it can be seen that the model error under has increased significantly compared with that under ε=0.0.

### 5.3 Imperceptibility of adversarial attack

#### 5.3.1 Adversarial attack based on importance measurement

In order to generate small adversarial samples and slightly disturb the time series data, we use the adversarial attack under the importance measure (i.e., algorithm 2, AAIM algorithm) to draw the adversarial time series diagram generated by algorithm 1. We estimate the importance of data points in the adversarial time series by measuring the impact of data points on the accuracy of the prediction model. Then, the most important p% data points are selected to perturb the original time series. AAIM further reduces the difference between the adversarial time series and the original data, and achieves a better attack effect with a minimal disturbance cost. The adversarial time series under ε=0.1 generated by algorithm 1 can produce the following results:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Perturbance percentage(%) | RSE | Perturbance percentage (%) | RAE | Perturbance percentage (%) | CORR |
| 0 | 0.107 | 0 | 0.075 | 0 | 1.471 |
| 1 | 0.312 | 1 | 0.171 | 1 | 0.689 |
| 5 | 0.873 | 5 | 0.498 | 5 | 0.000 |
| 15 | 0.869 | 15 | 0.513 | 15 | 0.000 |
| 20 | 0.868 | 20 | 0.513 | 20 | 0.000 |
| 100 | 0.870 | 100 | 0.514 | 100 | 0.000 |

Table 8 Attack effects under different perturbances



Figure 8(a). Attack result RSE under different perturbations



Figure 8(b). Attack result RAE under different perturbations



Figure 8(c). Attack result CORR under different perturbations

Fig. 8 shows the prediction effect of AAIM on the original data set. The horizontal axis represents the percentage of disturbance (between 0 and 100%), where 0% represents the model prediction based on the original time series, and 100% represents the prediction based on the antagonistic time series generated by algorithm 1. The vertical coordinates represent the evaluation indexes RSE, RAE and CORR respectively. It can be seen from Fig. 8 that the adversarial perturbance based on the first 5% importance measurement has almost the same effect as the 100% disturbance of the original time series. Therefore, the importance measure based adversarial attack algorithm, i.e. AAIM, significantly reduces the cost of disturbance, and only a small amount of disturbance is sufficient for adversarial attack.



Fig. 9 influence of important adversarial data points on prediction error

Fig. 9 shows the influence of important adversarial data points on the prediction error, wherein the red line represents the original time series and the blue line represents the adversarial time series. By calculating the importance of the perturbance in the adversarial time series, the important adversarial data points are obtained. Here, for the MSE evaluation index, we set n = 1 so as to directly measure the impact of antagonistic interference on the results of LSTNet. The abscissa represents the antagonistic interference arranged according to the importance. Three subsequences were randomly selected from each data set in order to widely evaluate counter attacks. According to figure 9, we can find that the more important hostile perturbation points force LSTNet to produce more error results, that is, the MSE value is larger. At the same time, we can observe that only a few perturbation points have a significant impact on the prediction of the model.

#### 5.3.2 BIM attacking method

In order to verify the superiority of the AAIM method (i.e., algorithm 2), we use BIM[21] as a comparison method for comparative experiments.

Through the existing theoretical methods, it can be seen that the process of obtaining the adversarial samples is conduct through the method of , that is, the perturbation is added to the original sequence x. The hypothesis theory in FGSM holds that the target loss function is assumed to be approximately linear with x, that is, .

At this time we need to make change most by adding a slight perturbance η to x. Since , the most direct way to maximize the change of is to make .

However, the above linear assumption is not necessarily true. If and x are not linear, we need to find some perturbance in , so that the increase of is the largest. At this time, the modification of pixel x can be smaller than ε, so Alexey Kurakin et al. proposed an iterative method to determine the perturbance of each pixel, instead of modifying all pixels at once, that is, an iterative FGSM(I-FGSM) or a basic iterative method (BIM) was obtained. The BIM formula is as follows:

Clip here means that during the iterative updating process, with the increase of the iterations, some pixel values of the sample may overflow and exceed the original range (for example, beyond 0 to 1). At this time, these values need to be replaced by 0 or 1, and finally an effective image can be generated. This process ensures that each pixel of the new sample will not have image distortion in a certain area of each pixel of the original sample. The iteration is specifically reflected in that each time on the basis of the adversarial sample in the previous step, each pixel increases α (or decrease), and then clip to ensure that each pixel of the new sample is in the neighborhood of x.

This iterative method has the possibility to find the adversarial sample when the change of each pixel is less than ε. If the finding process is impossible, the worst effect will be the same as the original FGSM.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Methods | F-norm | RSE | RAE | CORR |
| BIM | 217.696 | 0.869 | 0.620 | 0.008 |
| ATSG | 218.765 | 0.904 | 0.641 | -0.012 |
| AAIM(1%) | 28.845 | 0.850 | 0.584 | -0.023 |
| AAIM(3%) | 47.050 | 0.852 | 0.585 | -0.015 |
| AAIM(5%) | 58.389 | 0.854 | 0.586 | 0.005 |

Table 9 Comparison of counter attack methods

The attack effect and adversarial interference amount of these methods are tested on real time series data sets, and the results are shown in Table 9. The F-norm sequence represents the sum of the adversarial perturbations applied to the time series data points. The last three lines show that AAIM uses the first 1%, the first 3% and the first 5% perturbations based on importance measures to adversarial attacks. The experimental results show that the method greatly reduces the amount of disturbance required (the number of disturbance points against the attack is reduced by 90%, and the total disturbance value of all disturbance points is reduced by 70%), while maintaining the impact of the attack on the time series prediction model.

## 6. Conclusion

In this paper, aiming at the attacking of time series prediction model, the ATSG method based on model gradient information is proposed as a representative deep time series prediction method. The advantage of gradient based counter attack methods is that they can quickly generate counter attack samples that make the learning model produce wrong results. However, this method cannot control the optimal amount of counter interference. Therefore, on the basis of ATSG, this paper proposes the AAIM method, which ensures the attack effect and significantly reduces the amount of disturbance.

The experimental results of this paper prove the rationality of the hypothesis that the data points of time series have disproportionate influence on the learning model. According to the experimental results of AAIM method, the adversarial disturbance based on the first 5% importance measurement has almost the same effect as the 100% disturbance of the original time series. Therefore, AAIM algorithm significantly reduces the disturbance cost, and only a small amount of disturbance is enough to resist the attack. This method has achieved satisfactory performance in the adversarial attack and can significantly reduce the number of perturbance.

## References

[1] Shyi-Ming Chen, Xin-Yao Zou, Gracius Cagar Gunawan, Fuzzy time series forecasting based on proportions of intervals and particle swarm optimization techniques, Inf. Sci. 500 (2019) 127–139.

[2] Xiaoxiang Guo, Yutong Sun, Jingli Ren, Low dimensional mid-term chaotic time series prediction by delay parameterized method, Inf. Sci. 516 (2020) 1–19.

[3] Aditya Grover, Ashish Kapoor, Eric Horvitz, A deep hybrid model for weather forecasting, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 379–386.

[4] Nesreen K. Ahmed, Amir F. Atiya, Neamat El Gayar, Hisham El-Shishiny, An empirical comparison of machine learning models for time series forecasting. Econom. Rev. 29 (5–6) (2010) 594–621.

[5] Shaojie Qiao, Nan Han, Jianbin Huang, Kun Yue, Rui Mao, Hongping Shu, Qiang He, Xindong Wu, A dynamic convolutional neural network based shared-bike demand forecasting model, ACM Trans. Intell. Syst. Technol. 1 (1) (2021) Article 1.

[6] Xiaoyong Yuan, Pan He, Qile Zhu, Xiaolin Li, Adversarial examples: attacks and defenses for deep learning, IEEE Trans. Neural Networks Learn. Syst. 30 (9) (2019) 2805–2824.

[7] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, Michael K. Reiter, Accessorize to a crime: real and stealthy attacks on state-of-the-art face recognition, in: Proceedings of the 2016 ACM Sigsac Conference on Computer and Communications Security, 2016, pp. 1528–1540.

[8] Xingping Xian, Tao Wu, Shaojie Qiao, Wei Wang, Chao Wang, Yanbing Liu, Guangxia Xu, Deepec: adversarial attacks against graph structure prediction models, Neurocomputing 437 (2021) 168–185.

[9] Teng Huang, Yongfeng Chen, Bingjian Yao, Bifen Yang, Xianmin Wang, Ya Li, Adversarial attacks on deep-learning-based radar range profifile target recognition, Inf. Sci. 531 (2020) 159–176.

[10] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, Pierre-Alain Muller, Adversarial attacks on deep neural networks for time series classification, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), 2019, IEEE, pp. 1–8.

[11] Fazle Karim, Somshubra Majumdar, Houshang Darabi, Adversarial attacks on time series, IEEE Trans. Pattern Anal. Mach. Intell., Early Access (2020).

[12] Samuel Harford, Fazle Karim, Houshang Darabi, Adversarial attacks on multivariate time series. arXiv preprint arXiv:2004.00410, 2020.

[13] Zhongguo Yang, Han Li, Mingzhu Zhang, Jingbin Wang, Chen Liu, A method for resisting adversarial attack on time series classifification model in iot system, in: Proceedings of the International Conference on Web Information Systems and Applications, 2020, pp. 559–566.

[14] Shoaib Ahmed Siddiqui, Andreas Dengel, Sheraz Ahmed, Benchmarking adversarial attacks and defenses for time-series data, in: Proceedings of the 34nd International Conference on Neural Information Processing, 2020, pp. 544–554.

[15] Syama Sundar Rangapuram, Matthias Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, Tim Januschowski, Deep state space models for time series forecasting, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 7796–7805.

[16] Xindi Cai et al. Time series prediction with recurrent neural networks trained by a hybrid pso–ea algorithm, Neurocomputing 70 (13–15) (2007) 2342–2353.

[17] Guokun Lai, Wei-Cheng Chang, Yiming Yang, Hanxiao Liu, Modeling long-and short-term temporal patterns with deep neural networks, in Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, 2018, pp. 95–104.

[18] Xiangdong Ran, Zhiguang Shan, Yufei Fang, Chuang Lin, An lstm-based method with attention mechanism for travel time prediction, Sensors 19 (4) (2019) 861.

[19] Chenyou Fan, Yuze Zhang, Yi Pan, Xiaoyue Li, Chi Zhang, Rong Yuan, Di Wu, Wensheng Wang, Jian Pei, Heng Huang, Multi-horizon time series forecasting with temporal attention learning, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2527–2535.

[20] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, Rob Fergus, Intriguing properties of neural networks. arxiv 2013. arXiv preprint arXiv:1312.6199, 2013.

[21] Alexey Kurakin, Ian Goodfellow, Samy Bengio, Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533, 2016.

[22] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, Dawn Song, Robust physical world attacks on deep learning visual classifification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1625–1634.

[23] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, Alan Yuille, Adversarial examples for semantic segmentation and object detection, in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1369–1378.

[24] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, 2014..

[25] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780.

[26] Guokun Lai, Wei-Cheng Chang, Yiming Yang, Hanxiao Liu, Modeling long-and short-term temporal patterns with deep neural networks, in: Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, 2018, pp. 95–104.

[27] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, Evaluating the visualization of what a deep neural network has learned, IEEE Trans. Neural Networks Learn. Syst. 28 (11) (2016) 2660–2673.