Information Sciences 587 (2022) 794–812

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Small perturbations are enough: Adversarial attacks on time series prediction

Tao Wu [a], Xuechun Wang [b], Shaojie Qiao [c,*], Xingping Xian [a,*], Yanbing Liu [d], Liang Zhang [a]

[a] School of Cybersecurity and Information Law, Chongqing University of Posts and Telecommunications, Chongqing, China
[b] School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China
[c] School of Software Engineering, Chengdu University of Information Technology, Chengdu, China
[d] Chongqing Engineering Laboratory of Internet and Information Security, Chongqing University of Posts and Telecommunications, Chongqing, China

ARTICLE INFO

Article history:
Received 20 April 2021
Received in revised form 31 October 2021
Accepted 2 November 2021
Available online 16 November 2021

Keywords:
Time-series data
Time-series prediction
Adversarial attacks
Adversarial time series

ABSTRACT

Time-series data are widespread in real-world industrial scenarios. To recover and infer missing information in real-world applications, the problem of time-series prediction has been widely studied as a classical research topic in data mining. Deep learning architectures have been viewed as next-generation time-series prediction models. However, recent studies have shown that deep learning models are vulnerable to adversarial attacks. In this study, we prospectively examine the problem of time-series prediction adversarial attacks and propose an attack strategy for generating an adversarial time series by adding malicious perturbations to the original time series to deteriorate the performance of time-series prediction models. Specifically, a perturbation-based adversarial example generation algorithm is proposed using the gradient information of the prediction model. In practice, unlike the imperceptibility to humans in the field of image processing, time-series data are more sensitive to abnormal perturbations and there are more stringent requirements regarding the amount of perturbations. To address this challenge, we craft an adversarial time series based on the importance measurement to slightly perturb the original data. Based on comprehensive experiments conducted on real-world time-series datasets, we verify that the proposed adversarial attack methods not only effectively fool the target time-series prediction model LSTNet, they also attack state-of-the-art CNN-, RNN-, and MHANET-based models. Meanwhile, the results show that the proposed methods achieve a good transferability. That is, the adversarial examples generated for a specific prediction model can significantly affect the performance of the other methods. Moreover, through a comparison with existing adversarial attack approaches, we can see that much smaller perturbations are sufficient for the proposed importance-measurement based adversarial attack method. The methods described in this paper are significant in understanding the impact of adversarial attacks on a time-series prediction and promoting the robustness of such prediction technologies.

* Corresponding authors.
E-mail addresses: sjqiao@cuit.edu.cn (S. Qiao), xxp0213@gmail.com (X. Xian).

https://doi.org/10.1016/j.ins.2021.11.007
0020-0255/

1. Introduction

Time-series data refer to a series of statistical observations arranged in chronological order [6]. Time-series data are ubiquitous in the era of Industry 4.0, where millions of sensors are being deployed to collect sensing data, such as electricity loads, traffic flows, industrial monitoring, and climate trends. As the amount of data increases, time-series analysis has become the focus of numerous research and development projects in data mining [10]. The problem of time-series prediction is aimed at predicting future values by analyzing observations of the past and creating an analogy or extension based on the development process, direction, and trends [3,14]. Time-series prediction has a wide variety of applications, such as predicting weather conditions in the near future [13], forecasting traffic congestion [28], predicting future electricity consumption and determining the amount of electricity required for energy management [25], and detecting anomalous energy usage and unveiling electricity theft [49].

Traditionally, many parametric models have been proposed for time-series prediction, such as autoregressive (AR), exponential smoothing, and structural time-series models. However, the feature engineering of the models is always conducted manually and their results depend on the expertise of the researchers. Recently, the increasing data availability and computing power for machine learning methods have allowed a purely data-driven learning of temporal dynamics [1]. Specifically, because of the impressive performance of deep learning in various applications, including speech and language processing, face recognition and object detection, numerous deep neural network based time-series prediction methods have been developed [24], in which the learning mechanism can capture the dynamic correlations between variables and consider the mixture of short- and long-term repetitive patterns, thus enabling a greater accuracy [29].

Recent studies have shown that models based on deep neural networks have inherent disadvantages and are vulnerable to adversarial attacks. These attacks seek to find imperceptible perturbations and generate adversarial examples to produce incorrect outputs with high confidence [47]. That is, machine learning models, including neural networks, misclassify examples that are only slightly different from the original data. Initially, the concept of an adversarial attack was proposed to describe the instability and unreliability of deep neural networks (DNN) against imperceptible perturbations to the pixels of an image [36]. Subsequently, similar adversarial attack schemes against machine learning models have also been proven to be effective in various application domains [33,43,40,41,18]. For example, Sharif et al. [33] sought to attack face recognition systems by optimizing the color of sunglasses appearing in an image. Xian et al. [43] also proposed a deep-architecture-based adversarial attack method against link prediction methods on graph data. In addition, Huang et al. [18] developed an adversarial perturbation method to attack target-recognition tasks in the field of radar signal interpretation.

Given the growing availability of time-series data, the accurate results of time-series prediction play an essential role in solving real-world problems. It is therefore important to study the robustness of time-series prediction models. Meanwhile, owing to the excellent results, deep neural networks have become a fundamental part of the new generation of time-series analysis models. Recently, to explore the robustness of deep-learning based time-series analysis models, Fawaz et al. [9] leveraged the existing iterative adversarial attack mechanism to fool the residual-network based time-series classification model. Karim et al. [19] and Harford et al. [15] also proposed adversarial transformation network (ATN)-based attack methods for traditional time-series classification models, including 1-nearest neighbor dynamic time warping (1-NN DTW), a fully connected network, and a fully convolutional network (FCN). To resist adversarial attacks on a time-series classification model, Yang et al. [46] trained an adversarial example detector to differentiate the adversarial examples from normal examples. In addition, Siddiqui et al. [35] employed some of the well-proven adversarial defense methodologies tested on images and evaluated their robustness for time-series data. Although few adversarial attacks have been proposed against time-series analysis methods, the adversarial vulnerability of state-of-art deep time-series prediction models, such as LSTNet, RNN, and MHANET, has not received sufficient attention. Meanwhile, differing from the imperceptibility to human eyes in the image processing area, time-series data are more sensitive to abnormal perturbations. Thus, there are more stringent requirements for the amount of the perturbations. Therefore, in this study, we focus on the time-series prediction adversarial attack problem and aim to solve the following questions: Can state-of-art deep time-series prediction models be attacked by such adversarial examples? In addition, how can imperceptible adversarial examples be generated to slightly perturb the time-series data?.

In this study, we propose and formulate the time-series prediction adversarial attack problem. Here, we assume that the data points play different roles in the pattern of the observed time series and have a disproportionate impact on the learning models. Because of the state-of-art performance of a long- and short-term time-series network (LSTNet), we introduce a global perturbation-based adversarial time-series generation algorithm for LSTNet-based deep time series prediction model. To address the challenge of the slight adversarial perturbations, an advanced adversarial time-series generation algorithm based on the importance measurement is proposed. Experiments on real-world time-series datasets demonstrate that the proposed adversarial attack methods achieve a satisfactory performance on multiple state-of-art time-series prediction models.

Our main contributions are summarized as follows:

- Effectiveness. We formulate the time-series prediction adversarial attack problem and propose a global perturbation-based adversarial time-series generation algorithm using the gradient information of the prediction models. With the perturbations on original data, the generated adversarial time series can effectively lead to incorrect outputs.

- Imperceptibility. To ensure that adversarial time series are imperceptible, we develop an importance-measurement-based adversarial attack approach to minimize the difference between the adversarial examples and the original data. Compared with the existing adversarial attack approaches, the proposed method can attack state-of-art deep time-series prediction models with much smaller perturbations.
- Applicability and Transferability. Our approaches not only can be used for a specific time-series prediction model, they can also be applied to other prediction models. Moreover, the adversarial time-series generated for the target model can also be used to attack other time-series prediction models.
- Risk Analysis. We demonstrated the vulnerability of time-series prediction models to adversarial attacks. The transferability validation also shows that a priori knowledge of the target model is not required for the adversarial attacks. Thus, there are high security risks in practical time-series-prediction-based applications.

The remainder of this paper is structured as follows: Section 2 provides the background and related studies. Section 3 introduces the problem definition and target models. Section 4 describes our methods. Section 5 details the extensive experiments, and Section 6 provides some concluding remarks.

2. Background and related work

2.1. Deep neural networks for time-series prediction

Given the impressive performance of deep learning, deep neural network based time-series prediction has received significant attention in recent years. Because of the natural interpretation of time-series data as sequences of inputs and targets, many time-series prediction methods have been proposed based on recurrent neural networks (RNNs). Specifically, Rangapuram et al. [31] presented a novel approach to probabilistic time-series forecasting that parameterizes a particular linear state space model using an RNN. Chen et al. [2] proposed a hybrid of the particle swarm optimization and evolutionary algorithm based RNNs for time-series prediction. However, the RNN variants suffer from limitations in learning long-range dependencies in the data, and hence long short-term memory network (LSTM)-based methods [39] were developed. Furthermore, because of the invariance across spatial dimensions, convolutional neural networks (CNNs) with multiple layers of causal convolutions were proposed for time-series prediction [20]. By combining CNNs with RNNs, Lai et al. [23] proposed a novel deep learning framework, LSTNet, to extract short-term local dependence patterns between variables and discover long-term patterns of time-series trends. Attention mechanisms have recently been applied to the improvements of time-series prediction techniques. For instance, Ran et al. [30] proposed an LSTM-based method with an attention mechanism for travel time prediction. In addition, Fan et al. [8] proposed a multi-horizon time-series forecasting method with a temporal attention mechanism to capture the patterns in historical data. In this study, rather than considering the time-series prediction methods, the vulnerability of deep-learning-based time-series prediction models is explored through adversarial attacks.

2.2. Deep learning and adversarial attacks

Owing to the widespread application of intelligent systems, the security of deep learning has become increasingly important and has drawn the attention of many relevant researchers and practitioners. Szegedy et al. [36] pioneered the exploration of the stability of neural networks and revealed their vulnerability to imperceptible perturbations. In addition, Goodfellow et al. [12] attempted to explain this phenomenon and argued that neural networks are vulnerable to an adversarial perturbation primarily because they are linear. Since then, many efforts have been dedicated to exploring the vulnerabilities of various deep learning models (e.g., CNNs [26], LSTMs [27], and reinforcement learning (RL) [11]). Currently, the main applications of adversarial attacks are in the field of image processing. Eykholt et al. [7] proposed Robust Physical Perturbations (RP2) for generating adversarial examples of road signs to ensure that the object recognition system identifies such signs incorrectly. In addition, Sharif et al. [33] attacked face recognition models on both the digital and physical levels. To fool the semantic segmentation and object detection models, Xie et al. [45] generated adversarial perturbations to produce an incorrect prediction on all output labels. Moreover, because of the importance of graph mining and text analysis [42,44], adversarial examples also exist against the models of graph-structured [43] and text [5] data. Differing from the above studies, we focus on the adversarial attacks against time-series prediction models.

2.3. Adversarial attacks on time series analysis

Compared with the increasing interest in studying attack and defense mechanisms on images, graphs, and text data, pioneering studies have only recently been conducted to generate adversarial examples for time-series classification models. Specifically, Fawaz et al. [9] considered the vulnerability of deep learning models to adversarial time-series examples and leveraged the existing iterative adversarial attack mechanism to add imperceptible noise to the original time series, thereby 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 perturbations has not been

fully considered. Karim et al. [19] proposed the use of an adversarial transformation network (ATN) to attack various time-series classification models. Harford et al. [15] also proposed transforming the existing ATN on a distilled model to attack various multivariate time-series classification models. However, such studies are focused on the robustness of traditional time-series classification models, i.e., 1-NN DTW, FCN, and a fully connected network, rather than state-of-art deep time-series prediction models, such as LSTNet, RNN, and MHANET. In addition, Yang et al. [46] and Siddiqui et al. [35] studied the defense methodologies against traditional adversarial attack methods on time-series data. However, the characteristics of time-series data have not been adequately explored. Different from the above studies, this study aims to prospectively explore adversarial attacks on state-of-art deep time-series prediction models and generate imperceptible adversarial examples to slightly perturb the time-series data.

3. Problem definition and target models

3.1. Problem definition

Definition 1 (*Time series*). The time series $\mathrm{X}, \mathrm{X} = [x_1, x_2, \ldots, x_i, \ldots, x_T]$ is an ordered set of real values, where $x_i \in \mathbb{R}^n$ and n are the feature dimensions, and T is the length of X.

Definition 2 (*Time-series prediction*). Given a time series $\mathrm{X} = [x_1, x_2, \ldots, x_T]$, the time-series prediction task is to forecast the value of x_{T+h} based on $[x_{T-w}, x_{T-w+1}, \ldots, x_T]$, where w is the window size and h is the fixed horizon ahead of the current time stamp. We denote the corresponding true values as $\mathrm{Y} = [x_{T+1}, x_{T+2}, \ldots, x_{T+i}, \ldots, x_{T+h}], x_{T+i} = y_i \in \mathrm{Y}$. In most cases, the forecasting task horizon is chosen according to the demands of the environmental settings.

Definition 3 (*Adversarial time series*). Given a time series $\mathrm{X} = [x_1, x_2, \ldots, x_T]$, the attacker generates adversarial perturbations η and constructs adversarial time series $\hat{\mathrm{X}} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_T], \hat{\mathrm{X}} = \mathrm{X} + \eta$. Consequently, for the adversarial time series $\hat{\mathrm{X}}$, the time-series prediction models perform significantly worse than on the original data.

Definition 4 (*Time-series prediction of adversarial attacks*). The accuracy of the prediction models decreases as the perturbed data increase or the data distribution changes. Given the cost of data manipulation and the requirement that the added perturbations are imperceptible, the amount of data manipulations should be reduced and the distance between the adversarial time series and the original series should be minimized as much as possible. Hence, the time-series prediction of adversarial attacks can be formulated as a constrained optimization problem

$$\begin{aligned} &\eta^* = \arg\max_{\eta} ||\mathrm{Y} - \hat{\mathrm{Y}}|| \\ &\text{s.t.} \quad ||\mathrm{X} - \hat{\mathrm{X}}|| \leqslant \varepsilon, \mathrm{Y} = f(\mathrm{X}), \hat{\mathrm{Y}} = f\left(\hat{\mathrm{X}}\right) \end{aligned} \tag{1}$$

Here, the attacker tries to generate the optimal adversarial time series $\hat{\mathrm{X}}$ against the time-series prediction model f, while minimizing the distance between $\hat{\mathrm{X}}$ and X. Here, ε indicates the cost of an adversarial attack.

3.2. Long- and short-term time-series network model

LSTNet is a deep learning model for a time-series prediction [23]. Its overall architecture consists of a convolutional layer, a recurrent layer, a recurrent-skip layer, and a fully connected layer. The convolutional layer extracts local features for a better performance. The recurrent layer discovers complex long-term dependencies in the time series to acquire global patterns, thereby improving the predictive performance. A new recurrent structure called recurrent-skip has been added to uncover extremely long-term dependency patterns and to make the optimization easier by exploiting the periodicity of the time series. LSTNet incorporates an attention mechanism to alleviate the requirement regarding a predefined number of hidden cells being skipped.

The two variants of the RNN model, GRU[4] and LSTM[17], as well as LSTNet, can make the predictions more accurate by obtaining both long- and short-term patterns. However, GRU and LSTM have the problem of vanishing gradient, which means that the weights and biases in the network will not be efficiently updated during each training step, causing the entire model to be inaccurate. Therefore, LSTNet has a stronger robustness than GRU and LSTM.

3.3. Other methods

(1) Convolutional Neural Networks: CNNs were originally developed to solve computer vision problems; however, it has recently been shown that they can also work well for sequence prediction problems. As shown in Fig. 1, the CNN model mainly contains a convolutional layer, a pooling layer, and a fully connected layer. The convolutional layer extracts features

automatically through convolutional kernels, and the pooling layer subsamples the extracted features to condense the feature matrix while preserving the key information that will be more useful for the final prediction. The fully connected layer is used to compute the data processed by the convolutional and pooling layers and obtain the final prediction results. The output of the CNN model is as follows:

$$h(x) = \mathrm{ReLU}(\mathrm{W} * \mathrm{X} + b) \tag{2}$$

where ReLU is the activation function, $\mathrm{ReLU}(x) = \max(0, x)$, W is the weight matrix, and X is the input data.

(2) Recurrent Neural Networks: RNNs were originally adopted in the field of natural language processing to model textual data, which has contextual correlations in time and space, and accordingly the textual data have a chronological order. RNNs can capture the dependency between the samples in the time series. That is, the RNNs have a recurrent connection in the hidden state, and the looping constraint ensures that sequential information is captured in the input data. This is shown in Fig. 2. Thus, RNNs can obtain long-term macro information. The prediction of the RNN model at time t can be expressed as follows:

$$h_t = f(\mathrm{W}_{xh} x_t + \mathrm{W}_{hh} h_{t-1}) \tag{3}$$

$$y_t = g(\mathrm{W}_{hy} x_t) \tag{4}$$

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

(3) Multi-Head Attention Network (MHANet): In 2017, Vaswani et al. [38] proposed the multi-head attention mechanism, which uses multiple self-attention in different representation spaces to extract sequence features in parallel, as illustrated in Fig. 3. MHANet has the advantage of understanding the input series from different perspectives for capturing long-term trends, and the computational complexity is low. Attention is calculated using the following equation:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

where Q, K, and V denote the three vectors obtained by mapping from the input sequence X, and d_k represents the number of dimensions of the vector.

4. Gradient-based adversarial time-deries generators

In this section, we present the framework of time-series prediction of adversarial attacks and then describe how to generate the adversarial time series based on the gradient information of the model.

4.1. Framework

The adversarial attacks of time series prediction aim to generate well-crafted adversarial time series to fool the time-series prediction methods. Specifically, given the time signals over a period of time, the time-series prediction model can accurately predict the direction of future trends. However, the attack-based approach can generate adversarial perturbations to perturb the original time series and deceive the prediction methods. The adversarial time series and the original time series should be as close as possible. In this study, we attempt to minimize the distance between the two sequences by regulating the perturbations. Fig. 4 shows the framework of time-series prediction adversarial attacks. An adversarial attack

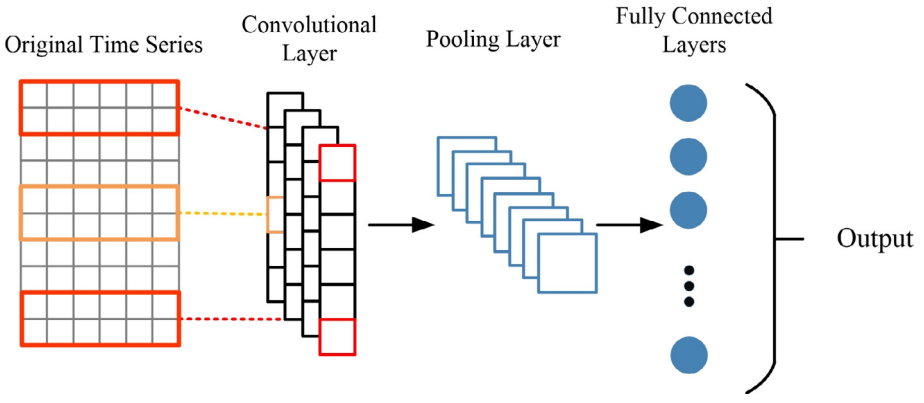


Fig. 1. Convolutional neural network architecture.

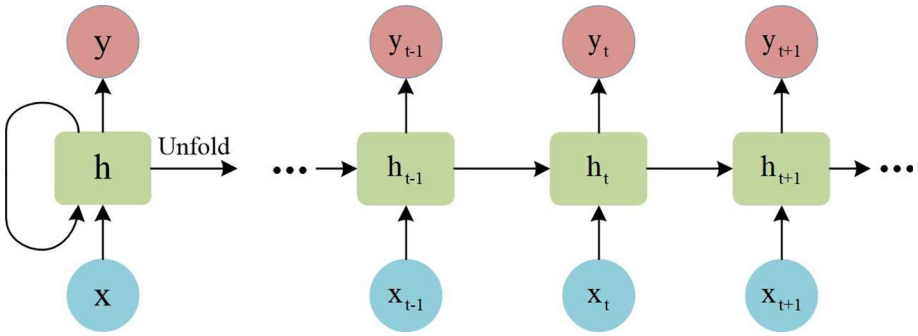


Fig. 2. Recurrent neural network architecture.

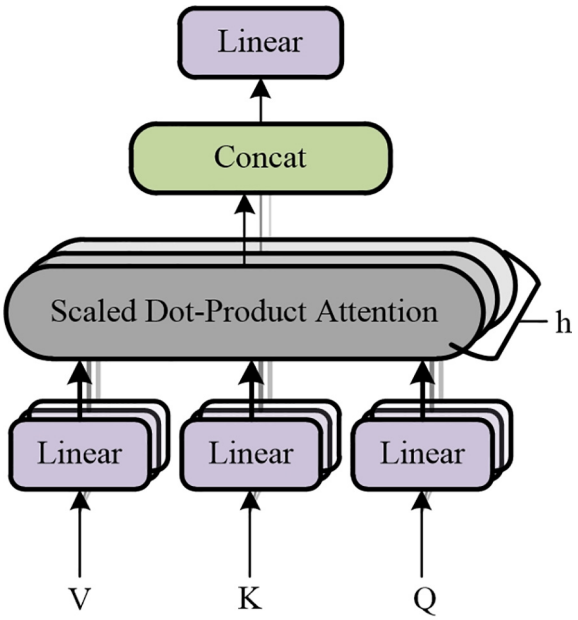


Fig. 3. Architecture of multi-head attention.

always includes three components: an adversarial time series generator, an adversarial attack, and a transferable attack. The details of the framework are discussed in detail below.

- **Adversarial time series generator.** The objective of an adversarial time series generator is to generate imperceptible but effective adversarial perturbations in the original time series against 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 through a partial derivation of the loss value.
- **Time series prediction adversarial attack.** Adversarial attacks implemented by the adversarial time series prevent future values from being accurately predicted. Given the cost of data manipulation and the imperceptibility of adversarial perturbations, the objective of an adversarial attack is to undermine the accuracy of time series prediction methods as much as possible.
- **Transferable Attack.** Many state-of-art deep neural networks have been applied to the time-series prediction problem. If the adversarial time series generated for a specific prediction model are effective against the model itself while making other time-series prediction methods fail, the process is referred to as a "transferable adversarial attack." In this paper, we analyze the possibility of transferable adversarial attacks and conduct comprehensive experiments to verify the transferability of such methods.

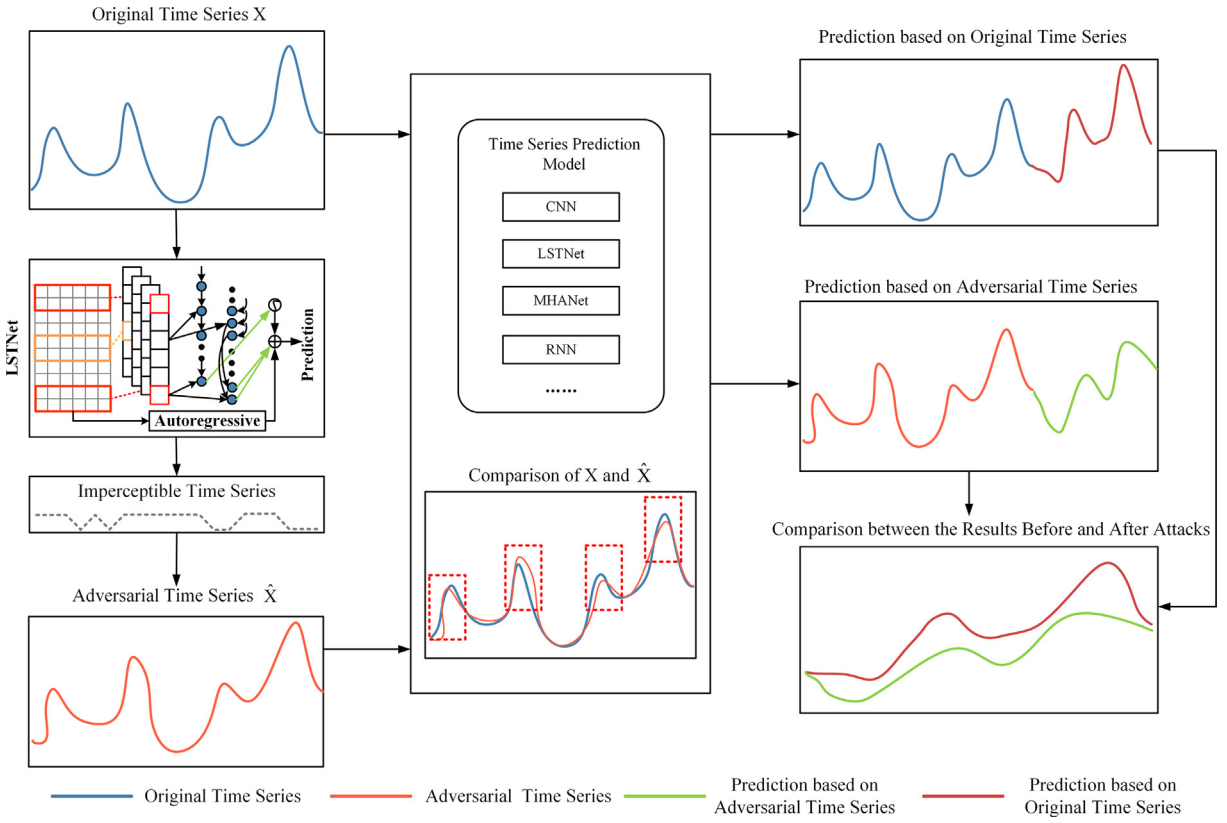


Fig. 4. Framework of time-series prediction adversarial attacks.

4.2. LSTNet model for time series prediction

The nonlinear characteristic of the convolutional and recurrent layers in a deep neural network model causes a problem in which the size of the output of the neural network model is insensitive to the size of the input. However, real-world time series are frequently changing in a non-periodic pattern, which significantly decreases the prediction accuracy of neural network models. To solve this problem, the LSTNet model [23] decomposes the final prediction into nonlinear and linear components, with the nonlinear part solved by neural networks and the linear part solved by the 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 $X \in \mathbb{R}^{T\times n}$, the AR model predicts as follows:

$$h_t^{AR} = \sum_{i=0}^{n}\sum_{j=0}^{q-1} W_j^{AR} X_{t-j,i} + b^{AR} \tag{6}$$

where h_t^{AR} denotes the predicted value of the 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:

$$h_t^{D} = W^{R} h_t^{R} + \sum_{j=0}^{p-1} W_j^{RS} h_{t-j}^{RS} + b \tag{7}$$

where h_t^{D} is the prediction value of the neural network at moment t, W^R is the weight matrix of the recurrent layer, h_t^R is the output of the recurrent layer at time t, p is the number of hidden states of the recurrent-skip layer, W_j^{RS} is the weight matrix of the j-th hidden state, b represents the bias units, and h_{t-j}^{RS} is the output of the j-th hidden state. The outputs of the neural network and AR components are accumulated to obtain the final prediction,

$$Y_t' = h_t^{D} + h_t^{AR}. \tag{8}$$

The objective function of LSTNet with L1-Loss is defined as

$$\arg\min_{W}\sum_{t=1}^{T}|Y_t - Y'_t|, \tag{9}$$

where Y_t is the true value at time t, and W denotes the parameters of the model.

4.3. Adversarial time series generator

To acquire the generalization capability, the LSTNet model is trained using a stochastic gradient descent optimization strategy that applies the gradient to update the parameters continuously for the loss function to be as small as possible. The process is repeated until convergence and the final values of the parameters are learned. Here, we denote the objective function of LSTNet in (9) as $J(W,X)$ and initialize the parameters of the model randomly. The parameter values are then updated in each iteration and eventually converge to the optimal solution. As shown in Fig. 5, when training the model, the opposite direction of the gradient is followed to find the minimum value of the objective function $J(W,X)$. That is, the gradient updating is used to change the values of the model parameters W, thereby minimizing the global cost.

To attack the LSTNet model, the attacker needs to generate adversarial perturbation η and construct adversarial time series $\hat{\mathrm{X}}, \hat{\mathrm{X}} = \mathrm{X} + \eta$, which can be formulated as follows:

$$\arg\max_{\eta}\sum_{t=1}^{T}|Y_t - Y'_t| \quad \text{s.t.} \quad \|\mathrm{X} - \hat{\mathrm{X}}\|_{\text{norm}} \leqslant \varepsilon, \tag{10}$$

where Y_t is the true value corresponding to time series X, Y'_t is the prediction result corresponding to adversarial time series $\hat{\mathrm{X}}, Y'_t = h_t^D\left(\hat{X}\right) - h_t^{AR}\left(\hat{X}\right)$, and $h_t^D\left(\hat{X}\right)$ and $h^{AR}\left(\hat{X}\right)$ are the prediction results of the nonlinear and linear components, respectively. Here, $||\cdot||_{\text{norm}}$ denotes the matrix norm and can be equal to 2 or ∞, and the constraint limits the amount of perturbation to make the distance between $\hat{\mathrm{X}}$ and X as small as possible.

Inspired by the gradient-based model attack strategy [12,21], the adversarial time series $\hat{\mathrm{X}}$ can be generated based on the gradient information of the model. In other words, if we want to attack the model, we can calculate the gradient of the cost function with respect to the input X and perturb it according to the direction of the gradient sign, i.e., $\text{sign}\left(\nabla_x\sum_{t=1}^{T}|Y_t - Y'_t|\right)$. As shown in Fig. 6(a), given the learned time series prediction model, for the specific time series x, the direction of the gradient where the loss function increases most rapidly can be produced. Then, according to Fig. 6(b), based on the perturbation magnitude ε, the adversarial perturbation η and the adversarial time series $\hat{\mathrm{X}}$ are generated to fool the target model. Finally, according to the above discussions, the details for the adversarial time series generator are as summarized in Algorithm 1.

Algorithm 1: Adversarial time series generator (ATSG)

Input: Original time series X, original time series label Y, perturbation magnitude ε, time series prediction model f;

output: Adversarial time series $\hat{\mathrm{X}}$;

1: Train on input data X and obtain the learned time series prediction model f^*;

2: Given the model f^*, calculate the gradient $\nabla_x\sum_{t=1}^{T}|Y_t - Y'_t|$;

3: Generate the adversarial perturbation according to the sign of the gradient, $\eta = \varepsilon \cdot \text{sign}\left(\nabla_x\sum_{t=1}^{T}|Y_t - Y'_t|\right)$;

4: Construct adversarial time series $\hat{\mathrm{X}}$ based on the perturbation, $\hat{\mathrm{X}} = \mathrm{X} + \eta$;

5: Return the adversarial time series $\hat{\mathrm{X}}$.

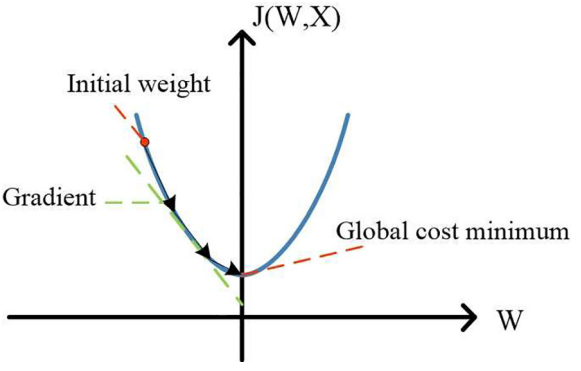
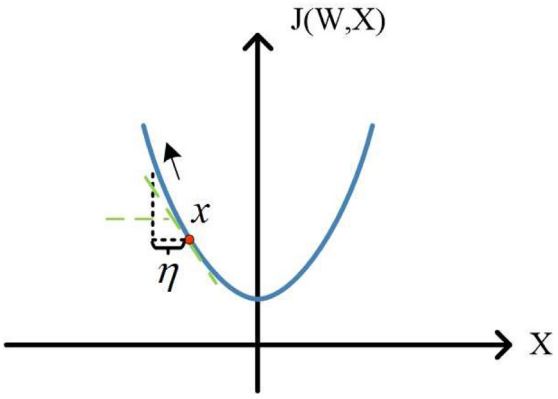


Fig. 5. Gradient descent for model solving.



(a) Getting the gradient of the cost function.

$$\begin{bmatrix} 0.9 & 0.8 \cdots & 0.8 \\ & \cdots\cdots & \\ 0.7 & 0.5 \cdots & 0.6 \\ 0.6 & 0.7 \cdots & 0.8 \end{bmatrix} + \varepsilon * \begin{bmatrix} -1 & 0 \cdots & 1 \\ & \cdots\cdots & \\ 1 & -1 \cdots & 0 \\ -1 & 0 \cdots & -1 \end{bmatrix}$$

(b) Illustration of generating adversarial time series.

Fig. 6. Gradient-based generation of adversarial time series.

4.4. Adversarial attacks with importance measurement

According to Algorithm 1, an adversarial time series can be obtained based on global perturbations to attack the time series prediction model. However, although the amount of maximum perturbation ε is small, the attacks in Algorithm 1 will change every point of the original data, which will not ensure the imperceptibility of the perturbations and the utility of the data [42]. Based on our previous study [44], herein, we assume that the data points in the time series data play different roles in the potential patterns, where some of them have a disproportionate influence on the regularity of the data, and the time series data can then be perturbed based on a limited number of data points. Meanwhile, the area over the perturbation curve (AOPC) method [32] was proposed to quantify the importance of the features to the model performance. Specifically, the feature sequence $O = (r_1, r_2, \ldots, r_L)$ is obtained by sorting the features in descending order of importance. Then, the feature r_k is perturbed sequentially with the Most Relevant First rule. The recursive formula is defined as follows:

$$\forall\, 1 \leqslant k \leqslant L : x_{MF}^{(k)} = g\left(x_{MF}^{(k-1)}, r_k\right) \tag{11}$$

where $x_{MF}^{(0)} = x$, and the function g represents the perturbation of the features r_k. The AOPC is defined as follows:

$$AOPC = \frac{1}{L+1}\left\langle \sum_{k=0}^{L} f\left(x_{MF}^{(0)}\right) - f\left(x_{MF}^{(k)}\right) \right\rangle_{p(x)} \tag{12}$$

where $f\left(x_{MF}^{(k)}\right)$ represents the prediction of the perturbed samples, and $\left\langle \cdot \right\rangle_{p(x)}$ denotes the average of all data. Thus, the perturbation of more important features will produce larger AOPC values.

Based on the above discussions, we argue that adversarial attacks should be conducted on the data points, which have much more influence on the model performance than the others. That is, the model can be fooled effectively by perturbing the important data points while reducing the difference between the perturbed time series and the original time series as much as possible. To find the important data points for adversarial attacks, this study proposes an attack method based on the importance measuring. The details of the method are shown in Algorithm 2. As shown in Algorithm 2, the distance between $\hat{y}_i$ and y_i is calculated to measure the importance of the data points in the time series. The larger the distance, the greater the contribution of $\hat{x}_i$ for the time-series prediction. Finally, based on the perturbation proportion P, the most important data points are selected for adversarial attacks.

Algorithm 2: Adversarial attack with importance measuring (AAIM)

Input: Original time series X, original time series label Y, adversarial time series $\hat{X} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_t, \ldots, \hat{x}_T]$, time series prediction model f, perturbation proportion P;

Output: Adversarial time series $\hat{X}'$;

1: Constructing $\hat{Y}'_t = f(\hat{x}_1, \ldots, \hat{x}_{t-1}, x_t, \hat{x}_{t+1}, \ldots, \hat{x}_T), t = (1, 2, \ldots, T)$ for each data point iteratively to estimate its importance, x_t denotes the original data point at time t;
2: **for** $t = 1$ to T **do**
3: Calculating the importance of the data point at time t, $importance_t = \|Y_t - \hat{Y}'_t\|$;
4: **end for**
5: Sorting the data points in descending order according to $importance_t$ and selecting the top P% data points;
6: Obtain $\hat{X}'$ by perturbing the top P% important time points of the original time series X according to $\hat{X}$;
7: Return adversarial time series $\hat{X}\prime$.

4.5. Loss function

In this subsection, we discuss the choice of the loss function. In general, L_1-Loss and L_2-Loss are used as loss functions. It can be seen that for the noise points, L_2-Loss squares the error, and therefore, the calculated error value is sensitive to the noises in the dataset. By contrast, L_1-Loss is robust to the noises and is usually not affected by them. In this study, we analyze the robustness of the prediction models with the L_1 and L_2 loss functions to extensively explore the performance of the proposed adversarial attack methods.

$$L_1 = \sum_{i=0}^{n} |Y_i - f(X_i)| \tag{13}$$

$$L_2 = \sum_{i=0}^{n} (Y_i - f(X_i))^2 \tag{14}$$

where Y_i represents the target value and $f(X_i)$ indicates the prediction value of the models.

4.6. Time complexity analysis

The most time-consuming components of Algorithm 1 are the gradient computations in step 2. The gradient computation takes $O(T)$ for the time series of length T, and the complexity of the adversarial perturbation generation can be ignored. Therefore, the total time complexity of Algorithm 1 is $O(T)$. 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 and sorting them based on the importance scores from steps 2 to 5 and perturbing the most important time points in step 6. For the adversarial time series corresponding to all data points, the importance estimation of every data point for adversarial attacks costs $O(T)$. The complexity of the ordering operation in step 5 is $O(T \log T)$, and the perturbation operation in step 7 costs $O(N), N = P * T, N < T$. Thus, the Algorithm 2 costs nearly $O(T) + O(T \log T) + O(N)$.

5. Experiments

In this section, extensive experiments are conducted on four state-of-art deep time series prediction models, and four evaluation metrics and three real-world industrial datasets are used to verify the performance of the proposed adversarial attack methods.

5.1. Experiment setup

5.1.1. Datasets

This study uses three publicly available time-series datasets for a performance evaluation. The datasets are divided into a training set, a validation set, and a test set at scales of 0.6, 0.2, and 0.2, respectively.

(1) Electricity [37]: The data points in the dataset were collected all day once every 15 min and represent 96 measurements. This dataset contains household electricity consumption data collected by 321 meters from 2012 to 2014, in which each column represents one client. Here, the data are preprocessed to obtain the records in terms of kilowatts per hour.

(2) Solar [48]: These data are intended for use by energy professionals. The dataset contains records of solar power production in 2006 and was collected once every 5 min. The experiment in this paper uses data collected from 137 photovoltaic power plants in the state of Alabama, USA.
(3) Household_power_consumption [16]: These data are the electricity consumption data of a French household from December 2006 to November 2010, containing nine attributes. This study only uses the attributes global_active_power and global_reactive_power. The electricity consumption data were collected once per minute, with missing values. We replace the missing values with zeros for the experiments.

5.1.2. Time series prediction methods

To demonstrate that our adversarial attack methods are applicable to various time series prediction models and the generated adversarial time series can be transferred over different models, we selected four state-of-art time-series prediction methods, i.e., RNN, CNN, LSTNet, and MHANet, for evaluation, the details of which are presented in Section 3.4.

5.1.3. Metrics

As the evaluation metrics, this study uses the root relative squared error (RSE), relative absolute error (RAE), and empirical correlation coefficient (CORR) for a performance evaluation. Moreover, for simplicity, the mean squared error (MSE) is adopted to measure the influence of a single-point perturbation on the model performance directly. The metrics are the most popular accuracy measures of the differences between the values predicted by a model and the actual values observed [23,34]. The RSE, RAE, and MSE are within the range of $[0, +\infty]$. When the predicted value is exactly consistent with the true value, they are equal to zero, which is the perfect model. The larger the errors, the greater the values. The values of CORR range between -1.0 and 1.0. A correlation of -1.0 shows a perfect negative correlation, whereas a correlation of 1.0 shows a perfect positive correlation. A correlation of 0.0 shows no linear relationship between the movement of the two variables. The goal of adversarial attacks is to fool the prediction models to produce inaccurate results, meaning that effective attack methods should result in large error values and a correlation value of close to zero. Specifically, the evaluation metrics are defined as follows.

(1) For the root RSE,

$$RSE = \frac{\sqrt{\sum_{(i,t)\in\Omega_{\text{Test}}} \left(Y_{it} - Y'_{it}\right)^2}}{\sqrt{\sum_{(i,t)\in\Omega_{\text{Test}}} \left(Y_{it} - \text{mean}(Y)\right)^2}}. \tag{15}$$

(2) For the RAE,

$$RAE = \frac{\sum_{(i,t)\in\Omega_{\text{Test}}} |Y_{it} - Y'_{it}|}{\sum_{(i,t)\in\Omega_{\text{Test}}} |Y_{it} - \text{mean}(Y)|}. \tag{16}$$

(3) For the CORR,

$$CORR = \frac{1}{n}\sum_{i=1}^{n} \frac{\sum_t (Y_{it} - \text{mean}(Y_i))\left(Y'_{it} - \text{mean}(Y'_i)\right)}{\sqrt{\sum_t (Y_{it} - \text{mean}(Y_i))^2 \left(Y'_{it} - \text{mean}(Y'_i)\right)^2}}, \tag{17}$$

where Y, Y' are the true and the predicted values, respectively.
(4) For the MSE,

$$MSE = \frac{\sum_{(i,t)\in\Omega_{\text{Test}}} \left(Y_{it} - Y'_{it}\right)^2}{n}, \tag{18}$$

where n is the number of data points.

Adversarial attack methods are designed to deliberately fool the prediction models while minimizing the perturbation. To quantify the amount of adversarial perturbations, this study adopts a Frobenius norm to quantify the distance between the adversarial time series and the original time series.

(1) Frobenius norm (F-Norm): the Frobenius norm is formulated as

$$\text{F} - \text{Norm} = \|\hat{\text{X}} - \text{X}\|_F \tag{19}$$

where $\hat{X}$ denotes the adversarial time series and X represents the original time series.

5.1.4. Parameter setting

The parameter ε in Algorithm 1 represents the total amount of perturbations in the adversarial time series. If the parameter is set to a large value, the adversarial perturbation will degrade the performance of the time-series prediction models significantly. However, because of the large amount of adversarial perturbations, they are easily identified and removed as noises, even if the goal of fooling the prediction models is achieved. Therefore, the purpose of adversarial attack methods is always to attack the target models with imperceptible perturbations. Thus, the value of ε should not be too large. Herein, to evaluate the performance of the proposed adversarial attack methods, we set ε to 0, 0.05, 0.1, 0.15, and 0.2, respectively.

5.2. Effectiveness, applicability, and transferability

.

(1) Effectiveness: Tables 1 and 2 show the effectiveness of the adversarial attack method ATSG (Algorithm 1) against LSTNet with L_1-Loss and L_2-Loss, respectively. When ε is equal to zero, we can see that LSTNet predicts the original time series quite well (with a low error and high correlation coefficient). As ε gradually increases, we can see that the error of the predicted values increases, and the correlation coefficient decreases. That is, the performance of LSTNet for a time-series prediction deteriorates as the perturbations increase. Moreover, Tables 1 and 2 show that there is little performance difference between LSTNet models using L_1-Loss and L_2-Loss. Meanwhile, the L_1 loss function is generally more robust to the anomaly in the real-time series data [22]. Therefore, the subsequent experiments in this paper use L_1-Loss for an evaluation of the adversarial attack methods.

(2) Applicability: Herein, the proposed adversarial attack method ATSG is conducted on the time-series prediction methods RNN, CNN, MHANet, and LSTNet, respectively, to evaluate its applicability. Fig. 7 shows the results of the time-series prediction methods with various amount of perturbations ε. In general, to all four state-of-art prediction models, the error values of the prediction methods all increase as the amount of perturbations rises, which demonstrates that the proposed ATSG method is suitable for the CNN, RNN, MHANet, and LSTNet. Moreover, Table 3 shows the CORR values between the prediction results based on the adversarial time series and the results based on true time series on the four state-of-art prediction models. When the perturbation magnitude ε equals zero, the time-series data are not perturbed and the CORR values between the prediction results and the true values are large. When the magnitude of the perturbation ε increases, the data are perturbed and the correlation decreases. In addition to the perturbation amount ε, the F-Norm is used to quantify the distance between the adversarial time series and the original time series. As shown in Fig. 8, as the F-Norm increases, the error values of the prediction models increase and the correlation between the predicted values and the true values weakens. According to the above experiment results, we can conclude that ATSG is effective for all four prediction models, thereby achieving a good applicability.

(3) Transferability: A transferable attack generates an adversarial time series for a specific time-series prediction model but can also fool other prediction models. Here, on the Electricity, Solar and Household dataset, we confirm the transferability of the proposed adversarial attack method ATSG among the four time-series prediction models. Specifically, to evaluate the performance of ATSG for transferable attacks, Table 4(a) presents the adversarial attack results when ε equals 0.0 and 0.1. Next, Table 4(b) shows the results of transferable attacks, in which adversarial time series are generated for one of the models and then used as the input of the other models. By considering the experimental results in Table 4(a) and (b) comprehensively, we can see that the adversarial attacks can transfer between the models. That is, even if the specific target model is unknown, the adversarial time series generated against other prediction models are also effective on the target prediction model. To observe the effect of transferable attacks more directly, we present the relevant experiment results in Fig. 9. Fig. 9(a)–(d) present the performances of the prediction models, RNN, CNN, LSTNet, and MHANet, under various attack mechanisms. For example, in Fig. 9(a), the performance of the RNN prediction model under transferable attacks is illustrated. On the Electricity, Solar and Household dataset, the bars with a blue color denote the values of RSE and RAE using RNN_0.0 as

Table 1
Adversarial attack against LSTNet with L_1-Loss.

Datasets	Metrics	ε				
		0	0.05	0.1	0.15	0.20
Electricity	RSE	0.102	0.809	0.850	0.882	0.958
	RAE	0.058	0.4039	0.446	0.491	0.556
	CORR	0.871	0.003	−0.008	0.002	0.006
Solar	RSE	0.431	1.766	2.072	2.269	2.429
	RAE	0.244	1.874	2.208	2.432	2.640
	CORR	0.909	0.008	−0.003	−0.006	0.010
Household	RSE	0.496	0.806	0.904	1.015	1.130
	RAE	0.321	0.570	0.641	0.733	0.821
	CORR	0.623	0.003	−0.012	−0.032	0.005

Table 2
Adversarial attack against LSTNet with L_2-Loss.

Datasets	Metrics	ε				
		0	0.05	0.1	0.15	0.20
Electricity	RSE	0.101	0.806	0.874	0.942	1.057
	RAE	0.059	0.406	0.466	0.541	0.642
	CORR	0.875	0.001	−0.008	0.001	0.002
Solar	RSE	0.271	1.435	1.604	1.756	1.882
	RAE	0.168	1.471	1.701	1.883	2.032
	CORR	0.964	0.003	−0.009	−0.009	0.025
Household	RSE	0.414	0.812	0.899	1.051	1.174
	RAE	0.276	0.562	0.630	0.754	0.861
	CORR	0.669	−0.031	−0.007	−0.004	0.003

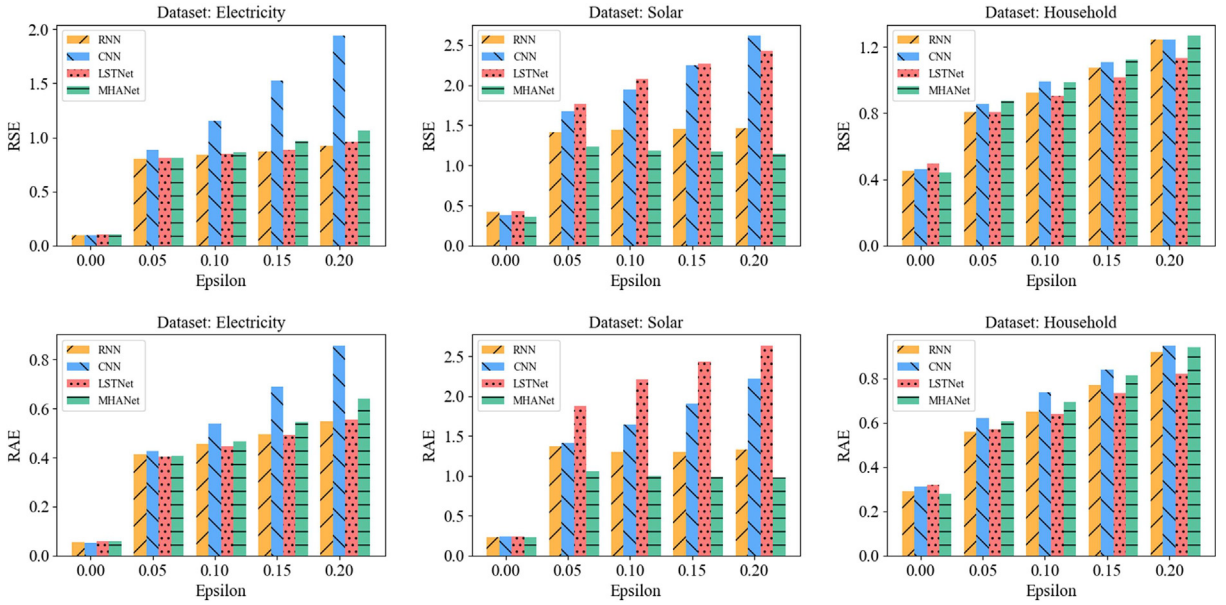


Fig. 7. The errors of time-series prediction models for adversarial time series generated with various perturbation amounts.

Table 3
The correlation coefficient (CORR) between the true values and the predicted values from adversarial attacks.

Datasets	Predictors	ε				
		0	0.05	0.1	0.15	0.20
Electricity	RNN	0.900	0.007	0.004	0.005	0.005
	CNN	0.911	0.005	−0.004	−0.002	−0.005
	LSTNet	0.871	0.003	−0.008	0.002	0.006
	MHANet	0.873	0.007	0.004	0.001	0.001
Solar	RNN	0.912	0.013	0.005	0.002	0.0035
	CNN	0.927	−0.019	0.003	0.014	−0.001
	LSTNet	0.909	0.008	−0.002	−0.006	0.010
	MHANet	0.933	−0.007	0.002	−0.013	−0.004
Household	RNN	0.683	0.009	0.003	0.003	−0.038
	CNN	0.633	−0.013	−0.005	−0.002	−0.011
	LSTNet	0.623	0.003	−0.012	−0.031	0.005
	MHANet	0.688	−0.014	−0.020	−0.005	0.000

the attack mechanism. That is, the adversarial time series are generated for the RNN model when ε equals 0.0 and there are no adversarial perturbations. The bars with red denote the values of the RSE and RAE using RNN_0.1 as the attack mechanism. That is, the adversarial time series are generated for the RNN model when ε equals 0.1, i.e., when an adversarial attack occurs. Similarly, the bars with green, yellow, and purple denote the values of RSE and RAE using CNN_0.1, LSTNet_0.1, and

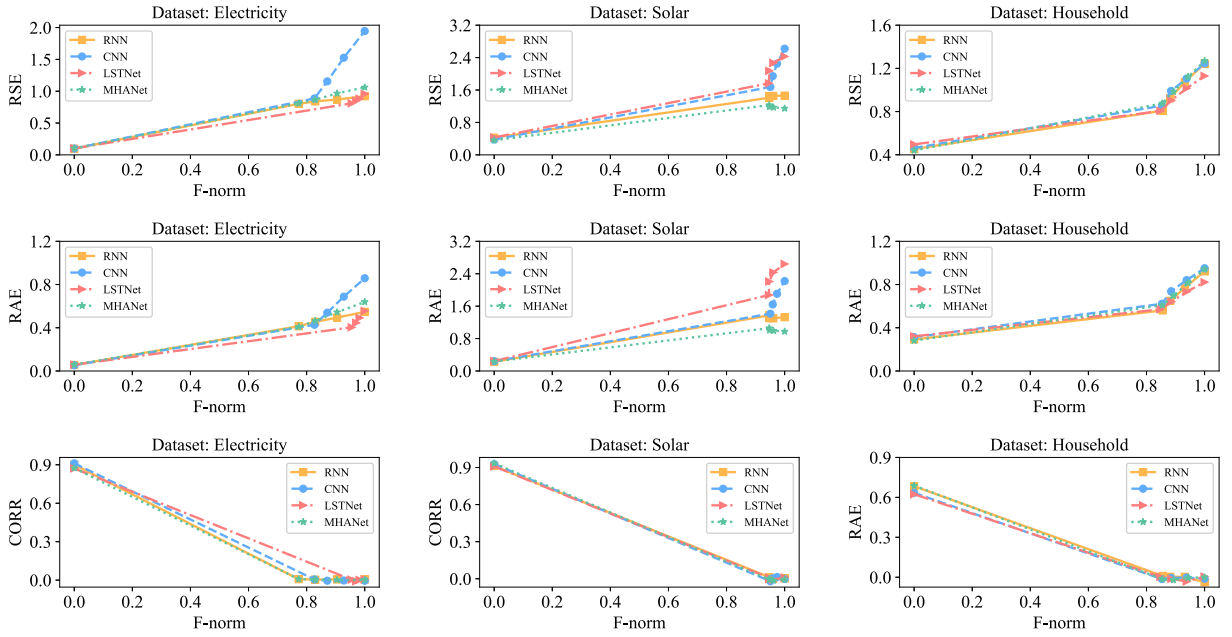


Fig. 8. Effectiveness of adversarial attacks with various perturbation distances.

MHANet_0.1 as attack mechanisms, respectively. That is, the adversarial time series are generated for the CNN, LSTNet, and MHANet when ε equals 0.1. As can be seen from Fig. 9, for all prediction models, although the error values of the models under transferable attacks are lower than those under adversarial attacks, they are generally much higher than those under no adversarial perturbations. This indicates that the adversarial perturbations against the other prediction models can also affect the intrinsic patterns of the input data and thus deteriorate the performance of the target model.

5.3. Imperceptibility of importance measurement based attacks

To generate imperceptible adversarial examples and slightly perturb the time-series data, AAIM (Algorithm 2) was proposed to craft the adversarial time series generated by Algorithm 1. Specifically, we estimate the importance of the data points in the adversarial time series by measuring their influence 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, achieving a better attack effect with an extremely small perturbation cost. The following experiments are based on the adversarial time series generated by Algorithm 1 with ε = 0.1.

Fig. 10 shows the performance of the importance-measuring-based adversarial attack algorithm AAIM on three datasets. The horizontal coordinate represents the percentage of perturbations (0–100%). Here, 0% represents the prediction of the model based on the original time series, and 100% represents the prediction based on the adversarial time series generated by Algorithm 1. The vertical coordinates indicate the evaluation metrics RSE, RAE, and CORR, respectively. Fig. 10 shows that the top 5% importance measuring-based adversarial perturbations have almost the same effect as the 100% perturbation of

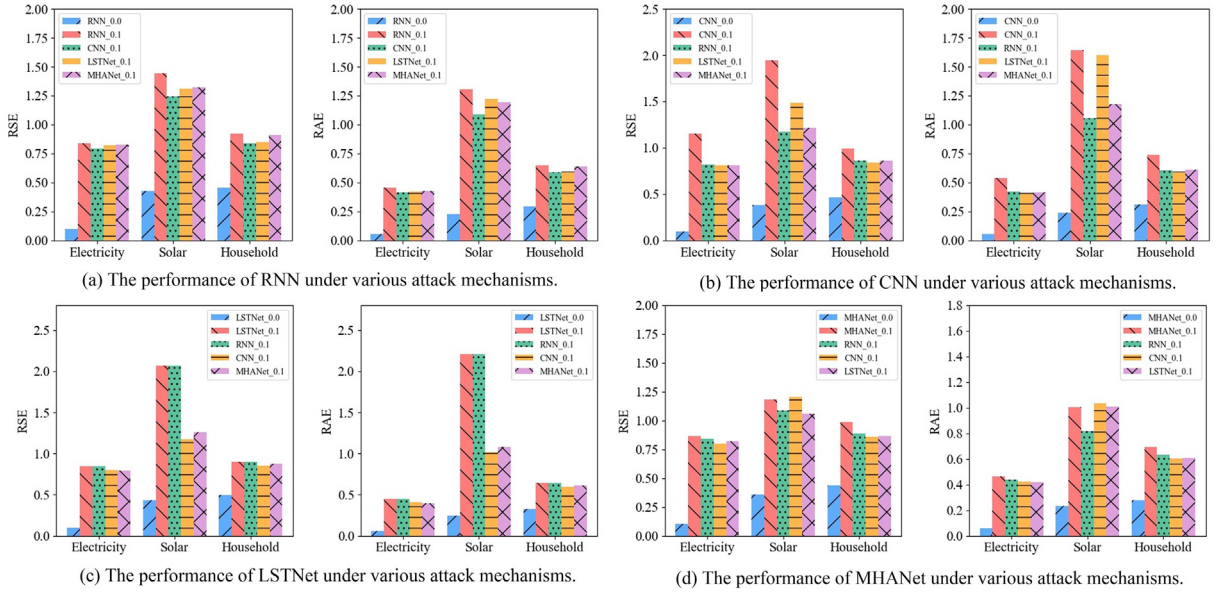
Table 4a
Transferability validation of adversarial attack method.

(a) Adversarial attack with ATSG

Datasets	Metrics	Adversarial attacks							
		epsilon = 0.0				epsilon = 0.1			
		RNN	CNN	LSTNet	MHANet	RNN	CNN	LSTNet	MHANet
Electricity	RSE	0.097	0.099	0.102	0.102	0.840	1.154	0.850	0.866
	RAE	0.055	0.053	0.058	0.059	0.455	0.540	0.446	0.464
Solar	RSE	0.425	0.378	0.431	0.360	1.445	1.943	2.072	1.185
	RAE	0.228	0.240	0.244	0.234	1.303	1.642	2.208	1.003
Household	RSE	0.453	0.463	0.496	0.440	0.924	0.991	0.904	0.986
	RAE	0.292	0.313	0.321	0.281	0.649	0.738	0.641	0.696

Table 4b
Transfer attack with ATSG.

Datasets	Metrics	(b) Transfer attack with ATSG. Transfer attacks, epsilon = 0.1											
		From RNN			From CNN			From LSTNet			From MHANet		
		CNN	LSTNet	MHANet	RNN	LSTNet	MHANet	RNN	CNN	MHANet	RNN	CNN	LSTNet
Electricity	RSE	0.817	0.799	0.842	0.795	0.797	0.801	0.824	0.810	0.819	0.825	0.811	0.795
	RAE	0.419	0.403	0.437	0.418	0.414	0.423	0.423	0.412	0.421	0.425	0.415	0.395
Solar	RSE	1.170	1.244	1.087	1.245	1.176	1.204	1.311	1.487	1.062	1.324	1.215	1.260
	RAE	1.058	1.209	0.820	1.086	1.019	1.033	1.220	1.602	1.007	1.192	1.177	1.079
Household	RSE	0.863	0.889	0.889	0.839	0.852	0.859	0.852	0.839	0.864	0.911	0.863	0.880
	RAE	0.607	0.625	0.635	0.586	0.594	0.604	0.594	0.592	0.607	0.641	0.609	0.616



(a) The performance of RNN under various attack mechanisms.

(b) The performance of CNN under various attack mechanisms.

(c) The performance of LSTNet under various attack mechanisms.

(d) The performance of MHANet under various attack mechanisms.

Fig. 9. Transferability validation by comparing adversarial attacks against the target model.

the original time series. Therefore, the importance-measuring-based adversarial attack algorithm, i.e., AAIM, significantly reduces the perturbation cost and only a small amount of perturbations are adequate for an adversarial attack.

Fig. 11 illustrates the impact of the important adversarial data points on the prediction error. The important adversarial data points are obtained by calculating the importance of the perturbations in the adversarial time series. Here, for the MSE evaluation metric, $n = 1$ is used to measure the influence of adversarial perturbations on the LSTNet results directly. The horizontal coordinates indicate the adversarial perturbations ranked according to their importance. To evaluate the adversarial attacks extensively, three sub-sequences are selected randomly from each dataset. According to Fig. 11, we can find that the more important adversarial perturbation points force LSTNet to produce more erroneous results, that is, a larger value of the MSE. Meanwhile, we can observe that only a small number of perturbation points have a significant impact on the model prediction.

To disclose the superiority of the proposed method AAIM (Algorithm 2), we carried out further experiments by adopting the BIM [21], a representative method in the field of adversarial attacks, as a comparison method. The attack effect and the amount of adversarial perturbations of the methods are tested on the real-world time series datasets, the results of which are shown in Table 5. The column of F-Norm denotes the sum of the adversarial perturbations applied on the data points of the time series. The last three rows indicate that AAIM adopts the top 1%, top 3%, and top 5% importance-measuring-based perturbations for adversarial attacks. The experimental results show that the proposed method AAIM greatly reduces the required amount of perturbations (the number of perturbation points for adversarial attacks is reduced by 90%, and the total perturbation value of all perturbation points is reduced by 70%), while maintaining the effect of the attack against the time-series prediction model. Moreover, Fig. 12 shows a comparison between the original time series and the adversarial time series visually. In Fig. 12(a), the gray line indicates the original time series, the blue line represents the adversarial time series

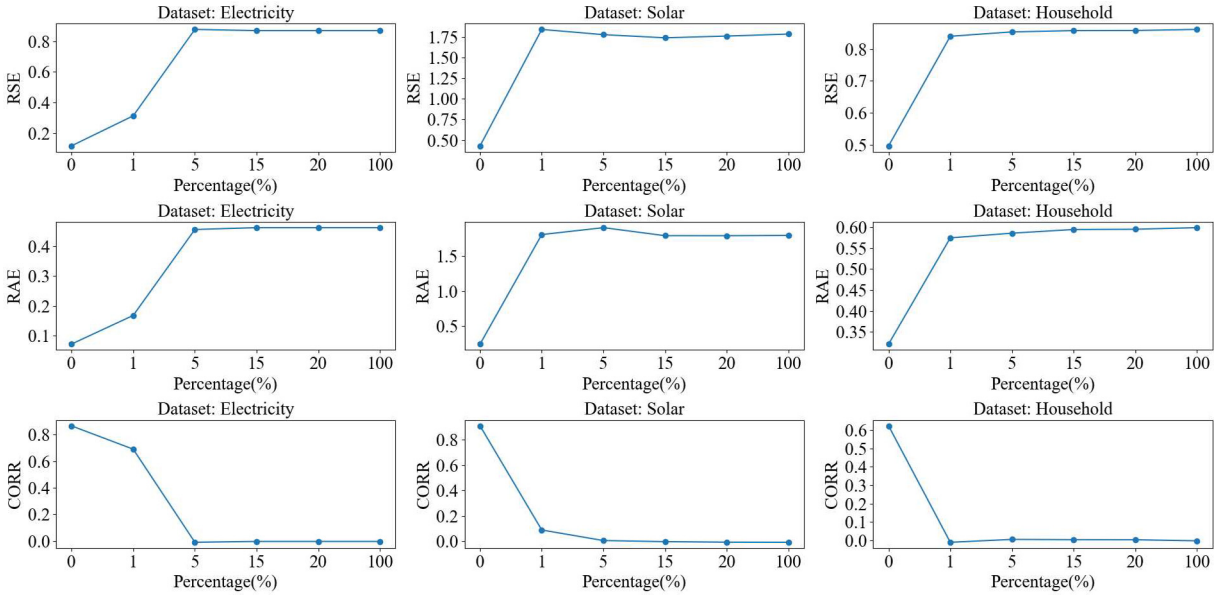


Fig. 10. Performance of importance measurement based attacks against LSTNet under various perturbation percentages.

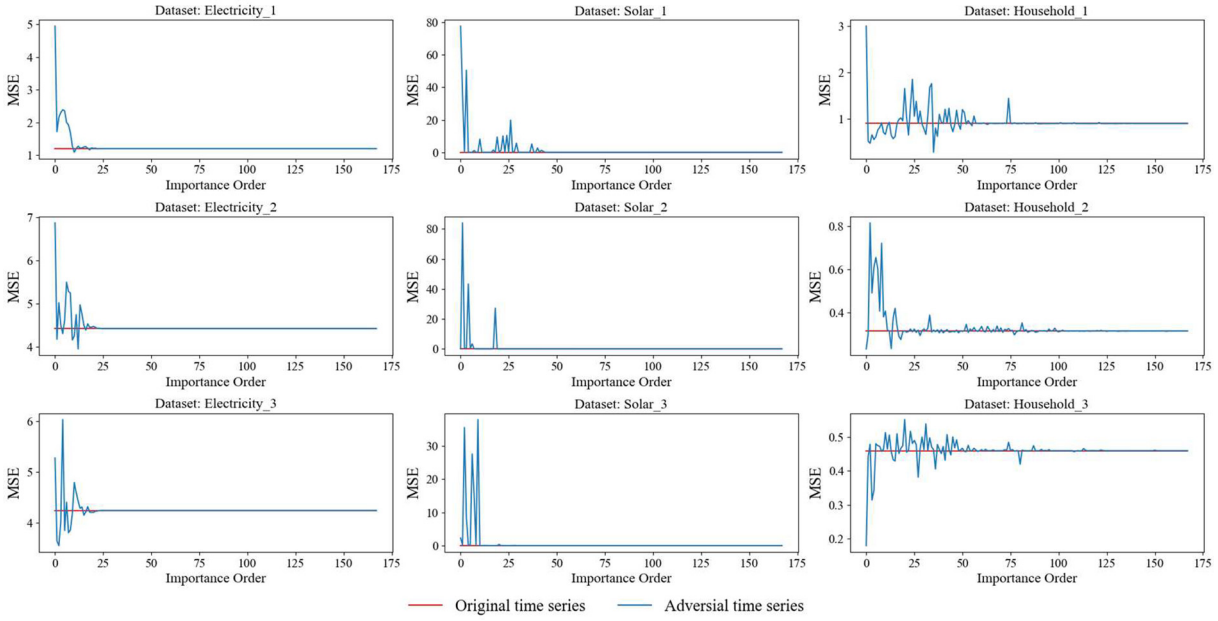
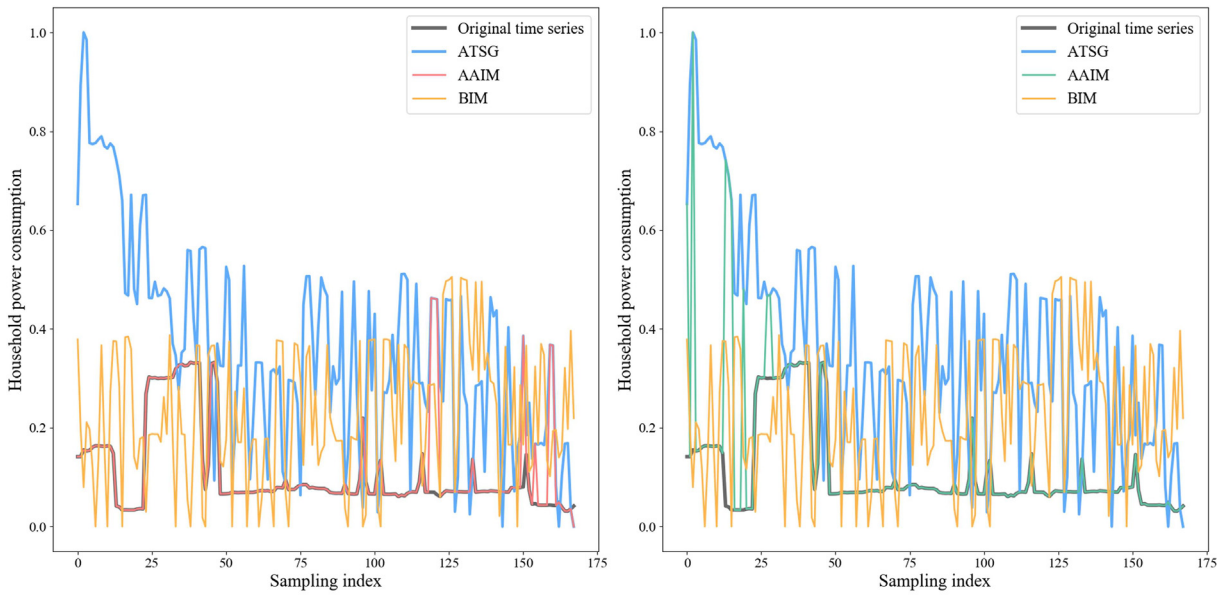


Fig. 11. The impact of important adversarial data points on prediction error.

Table 5
Comparison of adversarial attack methods.

Methods	F-Norm	RSE	RAE	CORR
BIM	217.6959	0.8686	0.6201	0.008
ATSG	218.7650	0.9039	0.6412	−0.0122
AAIM(Percentage = 1%)	28.8447	0.8500	0.5836	−0.0234
AAIM(Percentage = 3%)	47.0495	0.8515	0.5848	−0.0145
AAIM(Percentage = 5%)	58.38871	0.8539	0.5855	0.0045



(a) Distribution of the most important perturbation points.

(b) Distribution of the least important perturbation points.

Fig. 12. Comparison between the original time series and the adversarial time series (using Household as an example).

generated by ATSG, the red line indicates the crafted adversarial time series generated by AAIM, and the yellow line indicates the adversarial time series generated by BIM. We can see that the red line is extremely close to the gray line. That is, for AAIM, only a small amount of perturbations are needed to attack a time-series prediction model. By contrast, Fig. 12(b) shows the least important adversarial data points of the adversarial time series generated by the different methods. Combined with the results of Fig. 11, we can conclude that many perturbations are ineffective for time-series adversarial attacks. Therefore, we can conclude that the AAIM method achieves a satisfactory performance for adversarial attacks and can significantly reduce the amount of adversarial perturbations.

6. Conclusions and discussion

This study focuses on the problem of adversarial attacks on time-series prediction models. To the best of our knowledge, there have been few studies on adversarial attacks for the learning model of time-series data. Moreover, the existing studies have mainly focused on the migration and application of traditional adversarial attacks methods on the time-series data analysis tasks, particularly the attack methods proposed for image processing. In general, there are two deficiencies of the existing studies: (1) In fact, to the human eye, time-series data are more sensitive to adversarial perturbations than the image data. Hence, the assumption that a perturbation is imperceptible to the human eye for image applications is still invalid for a time-series data analysis. Therefore, there should be more strict requirements on the amount of adversarial perturbations for a time-series data analysis. (2) Several existing adversarial attacks that are effective against the learning tasks of time-series data do not target state-of-art deep prediction models, and thus cannot reflect their robustness. To solve the above problems, this paper proposes the ATSG adversarial attack method based on the model gradient information for the representative deep time-series prediction methods. As the advantage of gradient-based adversarial attack methods, they can quickly generate adversarial examples that make the learning model produce incorrect results. However, this type of method cannot control the optimal amount of adversarial perturbations. Therefore, on the basis of ATSG, the AAIM method is proposed herein, which significantly reduces the amount of perturbations while ensuring the effect of the attack.

In this paper, the experimental results prove the reasonability of the assumption that the data points of a time series have a disproportionate impact on the learning models. Thus, in the future, the design of sophisticated adversarial attack methods with subtle perturbations is expected. Moreover, the AAIM method proposed in this paper needs to calculate the importance of the data points successively. As future research, a way to develop advanced importance-measuring methods for adversarial attacks is worth considering.

CRediT authorship contribution statement

Tao Wu: Conceptualization, Methodology. **Xuechun Wang:** Formal analysis, Validation, Writing - original draft. **Shaojie Qiao:** Formal analysis, Validation, Writing - review & editing. **Xingping Xian:** Formal analysis, Writing - review & editing. **Yanbing Liu:** Writing - review & editing. **Liang Zhang:** Formal analysis, Validation, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is partially supported by the Natural Science Foundation of Chongqing under Grant No. cstc2020jcyj-msxmX0804; the National Natural Science Foundation of China under Grant Nos. 61802039, 62106030, 61772091, and 61802035; the Postdoctoral Science Foundation of Chongqing under Grant No. cstc2021jcyj-bsh0176; the National Key R&D Program of China under Grant Nos. 2018YFB0904900 and 2018YFB0904905; and the Sichuan Science and Technology Program under Grant Nos. 2021JDJQ0021 and 2020YJ0481.

References

[1] 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..
[2] Xindi Cai, Nian Zhang, Ganesh K. Venayagamoorthy, Donald C. Wunsch II, Time series prediction with recurrent neural networks trained by a hybrid pso–ea algorithm, Neurocomputing 70 (13–15) (2007) 2342–2353.
[3] 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.
[4] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, 2014..
[5] Javid Ebrahimi, Anyi Rao, Daniel Lowd, Dejing Dou, Hotflip: white-box adversarial examples for text classification. arXiv preprint arXiv:1712.06751, 2017..
[6] Philippe Esling, Carlos Agon, Time-series data mining, ACM Comput. Surveys (CSUR) 45 (1) (2012) 1–34.
[7] 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 classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1625–1634..
[8] 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..
[9] 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..
[10] Tak-chung Fu, A review on time series data mining, Eng. Appl. Artif. Intell. 24 (1) (2011) 164–181.
[11] Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, Stuart Russell, Adversarial policies: attacking deep reinforcement learning. arXiv preprint arXiv:1905.10615, 2019..
[12] Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy, Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014..
[13] 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.
[14] Xiaoxiang Guo, Yutong Sun, Jingli Ren, Low dimensional mid-term chaotic time series prediction by delay parameterized method, Inf. Sci. 516 (2020) 1–19.
[15] Samuel Harford, Fazle Karim, Houshang Darabi, Adversarial attacks on multivariate time series. arXiv preprint arXiv:2004.00410, 2020..
[16] Georges Hebrail, Individual household electric power consumption data set. https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+ consumption..
[17] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780.
[18] Teng Huang, Yongfeng Chen, Bingjian Yao, Bifen Yang, Xianmin Wang, Ya Li, Adversarial attacks on deep-learning-based radar range profile target recognition, Inf. Sci. 531 (2020) 159–176.
[19] Fazle Karim, Somshubra Majumdar, Houshang Darabi, Adversarial attacks on time series, IEEE Trans. Pattern Anal. Mach. Intell., Early Access (2020).
[20] Irena Koprinska, Dengsong Wu, Zheng Wang, Convolutional neural networks for energy time series forecasting, in: Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), 2018, IEEE, pp. 1–8.
[21] Alexey Kurakin, Ian Goodfellow, Samy Bengio, Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533, 2016..
[22] 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.
[23] 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.
[24] Bryan Lim, Stefan Zohren, Time series forecasting with deep learning: a survey, Philos. Trans. Roy. Soc. A Math., Phys. Eng. Sci. 379 (2194) (2021) 1–14.
[25] Yang Liu, Wei Wang, Noradin Ghadimi, Electricity load forecasting by an improved forecast engine for building level consumers, Energy 139 (2017) 18–30.
[26] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Pascal Frossard, Deepfool: a simple and accurate method to fool deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2574–2582.
[27] Nicolas Papernot, Patrick McDaniel, Ananthram Swami, Richard Harang, Crafting adversarial input sequences for recurrent neural networks, in: Proceedings of the MILCOM 2016–2016 IEEE Military Communications Conference, 2016, IEEE, pp. 49–54.
[28] Shaojie Qiao, Nan Han, Yunjun Gao, Rong-Hua Li, Jianbin Huang, Jun Guo, Louis Alberto Gutierrez, Xindong Wu, A fast parallel community discovery model on complex networks through approximate optimization, IEEE Trans. Knowl. Data Eng. 30 (9) (2018) 1638–1651..
[29] 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..
[30] Xiangdong Ran, Zhiguang Shan, Yufei Fang, Chuang Lin, An lstm-based method with attention mechanism for travel time prediction, Sensors 19 (4) (2019) 861.
[31] 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..
[32] 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.
[33] 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.

[34] Shun-Yao Shih, Fan-Keng Sun, Hung-yi Lee, Temporal pattern attention for multivariate time series forecasting, Mach. Learn. 108 (8) (2019) 1421–1441.
[35] 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.
[36] 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..
[37] Artur Trindade, Electricityload. https://archive.ics.uci.edu/ml/datasets/Electricity LoadDiagrams20112014..
[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, Attention is all you need. In Advances in Neural Information Processing Systems, 2017, pp. 5998–6008..
[39] Huaiyu Wan, Shengnan Guo, Kang Yin, Xiaohui Liang, Youfang Lin, Cts-lstm: Lstm-based neural networks for correlatedtime series prediction, Knowl.-Based Syst. 191 (105239) (2020).
[40] Xiao-Lei Wang, Optimal attack strategy against fault detectors for linear cyber-physical systems, Inf. Sci. 581 (2021) 390–402.
[41] Yiteng Wu, Wei Liu, Hu. Xinbang, Xuqiao Yu, Parameter discrepancy hypothesis: adversarial attack for graph data, Inf. Sci. 577 (2021) 234–244.
[42] Xingping Xian, Tao Wu, Yanbing Liu, Wei Wang, Chao Wang, Guangxia Xu, Yonggang Xiao, Towards link inference attack against network structure perturbation, Knowl.-Based Syst. 218 (2021) 106674.
[43] 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.
[44] Xingping Xian, Tao Wu, Shaojie Qiao, Xi-Zhao Wang, Wei Wang, Yanbing Liu, Netsre: link predictability measuring and regulating, Knowl.-Based Syst. 196 (2020) 105800.
[45] 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.
[46] Zhongguo Yang, Han Li, Mingzhu Zhang, Jingbin Wang, Chen Liu, A method for resisting adversarial attack on time series classification model in iot system, in: Proceedings of the International Conference on Web Information Systems and Applications, 2020, pp. 559–566.
[47] 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.
[48] Yingchen Zhang, Solar power data for integration studies. https://www.nrel.gov/grid/solar-power-data.html..
[49] Zibin Zheng, Yatao Yang, Xiangdong Niu, Hong-Ning Dai, Yuren Zhou, Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids, IEEE Trans. Ind. Inf. 14 (4) (2017) 1606–1615.