Ge et al. Cybersecurity (2023) 6:43
https://doi.org/10.1186/s42400-023-00179-4

Cybersecurity

RESEARCH Open Access

Attack based on data: a novel perspective to attack sensitive points directly

Yuyao Ge[1], Zhongguo Yang[2*], Lizhe Chen[1], Yiming Wang[1] and Chengyang Li[3]

Abstract

Adversarial attack for time-series classification model is widely explored and many attack methods are proposed. But there is not a method of attack based on the data itself. In this paper, we innovatively proposed a black-box sparse attack method based on data location. Our method directly attack the sensitive points in the time-series data according to statistical features extract from the dataset. At first, we have validated the transferability of sensitive points among DNNs with different structures. Secondly, we use the statistical features extract from the dataset and the sensitive rate of each point as the training set to train the predictive model. Then, predicting the sensitive rate of test set by predictive model. Finally, perturbing according to the sensitive rate. The attack is limited by constraining the L0 norm to achieve one-point attack. We conduct experiments on several datasets to validate the effectiveness of this method.

Keywords Black-box adversarial attack, Time series classification, Data mining

Introduction

Time Series Classification (TSC) problems are encountered in various real world data mining tasks such as security (Tan et al. 2017; Tobiyama et al. 2016), transportation (Nguyen et al. 2018), health care (Abdelfattah et al. 2018; Ma et al. 2018; Fawaz et al. 2018), etc. Under the background of big data, deep learning technology has developed rapidly and has been widely used in various fields such as computer vision and autonomous vehicles. In recent years, researchers have begun to apply deep learning techniques to TSC problems (Ismail Fawaz et al. 2019; Fawaz et al. 2019).

However, due to the wide applicability of DNNs, they pose a threat that cannot be ignored, so researchers began to study the vulnerability of DNNs. Adversarial attacks in the field of TSC are capable of leading neural networks to misclassify adversarial time series data while remaining difficult for humans to detect (Moosavi-Dezfooli et al. 2016; Papernot et al. 2016a). There are several types of attack methods that have been developed to perturb the input data in order to mislead the model's output, such as Gradient-based attacks (Goodfellow et al. 2014; Szegedy et al. 2014), Optimization-based attacks (Carlini and Wagner 2017), Transferability attacks (Papernot et al. 2017), Decision-Based attacks (Brendel et al. 2017), etc. In recent years, thanks to the increasingly in-depth research conducted by scholars, more and more effective adversarial attack methods and theories have been proposed.

Among the many attack methods, one-pixel attack has received widespread attention as an extreme attack method. One-pixel attack was presented by Su et al. (2019) as a special black-box sparse attack, due to its particular attack pattern, which only perturbs a single pixel of an image to mislead the classifier, different from other attacks that may change hundreds of pixels (Wang et al. 2020). Notwithstanding its efficacy with small datasets, the one-pixel attack method encounters

*Correspondence:
Zhongguo Yang
yangzhongguo@ncut.edu.cn
[1] School of Information Science and Technology, North China University of Technology, Beijing, China
[2] Beijing Key Laboratory on Integration and Analysis of Large-Scale Stream Data, North China University of Technology, Beijing, China
[3] School of Computer Science, Peking University, Beijing, China

difficulties in larger datasets due to the significant computational demands imposed by the requirement for thousands of iterations, thus constraining its practical application.

Inspired by the attack pattern of the one-pixel attack, we propose a black-box sparse attack method based on data which can find the **sensitive locations** directly.

Compared with the one-pixel attack proposed by Su et al. (2019), the adversarial attack method proposed in this paper is not based on query but based on statistical features of the datasets, which can substantially decrease the number of queries and attack the sensitive points directly. We believe that our approach presents a novel perspective to the field of adversarial attacks. The difference between the method presented in this article and the traditional methods are illustrated in Fig. 1.

In brief, the main contributions of this paper are summarized as follows.

- This article proposes a black-box attack method based on data and directly attack sensitive points, similar to the one-pixel attack, which only needs to perturb a very small number of time points, or even only one point.
- We have conducted experiments under various parameters of the method proposed in this paper. The experiments prove that adversarial attacks can be based not only on classification models but also on the statistical features of the dataset.

Related works

The security problem of DNN has become a critical topic since Szegedy et al. (2014) proposed adversarial examples for image recognition. Many researchers proposed attack methods from different perspectives. In the field of black-box attack, the methods are mainly based on substitute models, genetic algorithms, and zeroth-order optimization. For instance, Papernot et al. (2016b) proposed that utilizes the transferability property of the substitute model to approximate the target model. Wei et al. (2021) manipulated the image attributes to perform the black-box attacks with a genetic algorithm. Szegedy et al. (2014) revealed the sensitivity to well-tuned artificial perturbation which can be crafted by several gradient-based algorithms using back-propagation for obtaining gradient information. Specifically, the gradient-based method was proposed by Goodfellow et al. (2015), the attackers perturb the image in the direction of the gradient of the loss function with respect to the input image thus reducing the classification accuracy. Lin et al. (2020) propose a black-box technique: Black-box Momentum Iterative Fast Gradient Sign Method (BMI-FGSM) to test the robustness of DNN models.

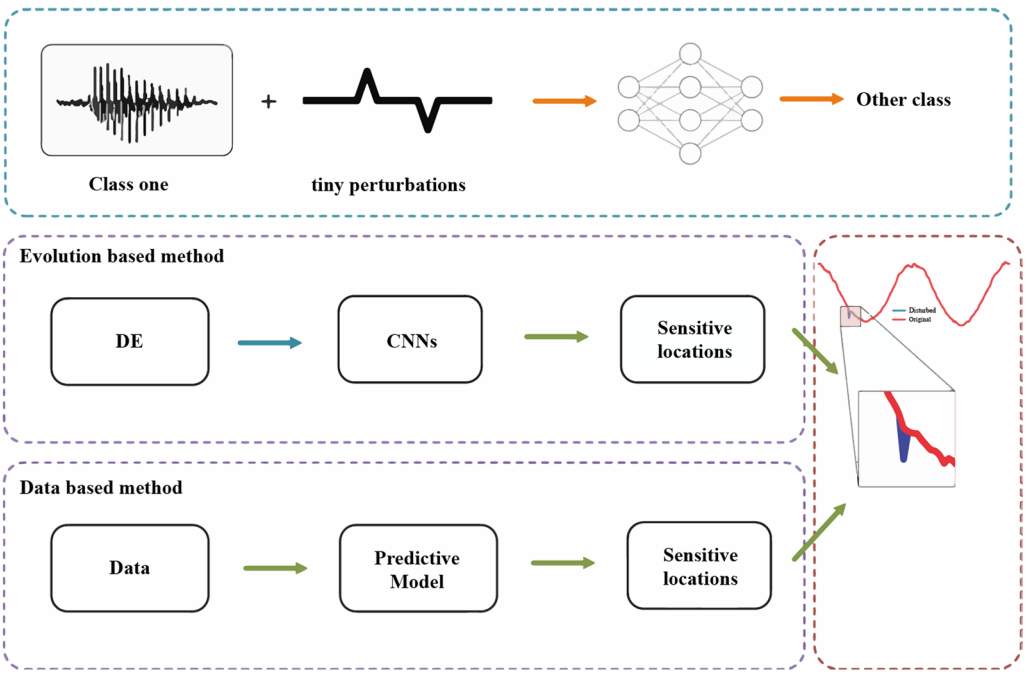


Fig. 1 Our method is based on data, by extracting the statistical features of the data to find the points which are suitable for the attack. Unlike genetic algorithms which require cyclic query, our method only requires a few times of visiting the target model or even a single visit

Recently, new attack methods are proposed. Hu et al. (2022) propose Adversarial Texture (AdvTexture) which hides from person detectors by covering clothes. Yang et al. (2022) propose a gradient-free black-box method called TSadv to attack DNNs with local perturbations. He et al. (2022) introduce a generator architecture to alleviate the overfitting issue. Kahla et al. (2022) proposed Boundary-Repelling Model Inversion (BREP-MI) to invert private training data using only the target model's predicted labels. Luo et al. (2022) proposed a novel algorithm that attacks semantic similarity on feature representations.

one-pixel attack Su et al. (2019) showed how neural networks can be fooled by altering the value of just a single input pixel. As shown in Fig. 2, unlike most attack methods,one-pixel attack only needs to perturb one point. By constraining the **L0** norm, they enforced a limit on the number of pixels that were allowed to be perturbed. One-pixel attack using differential evolution (Storn and Price 1997) which is an optimization algorithm that can be used to search for the minimum or maximum of a function. The researchers encode the perturbation into an array (candidate solution), each candidate solution contains a fixed number of perturbations and each perturbation is a tuple holding five elements: x-y coordinates and RGB value of the perturbation.

$$x_i(g+1) = x_{r1}(g) + F(x_{r2}(g) - x_{r3}(g)), \quad r1 \neq r2 \neq r3, \tag{1}$$

Once generated, each candidate solution competes with their parents according to the index of the population and the winner survive for next iteration.

In summary, each black-box attack method has its own strengths and weaknesses, and the choice of attack method depends on the specific target model and the attacker's resources and goals.

Among all the methods mentioned above, none of them is data-based, while our approach is a huge innovation in the field of adversarial attacks.

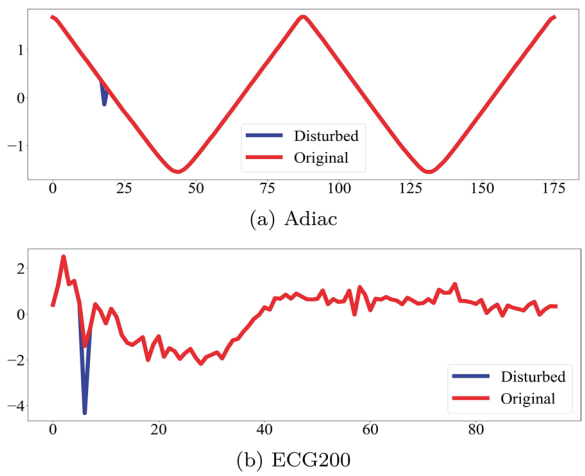


Fig. 2 One-pixel attacks on TSC successfully fooled ResNet architecture

Methodology

In this section, initially, we provide a description of the problem. Subsequently, we introduce the dataset and the main diagram illustration of the experiment. Finally, we describe the attack methodology in detail.

Problem description

We assume that the attacker has access to the target model in a black-box setting. The information, such as architecture, parameters is not accessible. Then, we assume part of the dataset (including both time series and labels) is public. Let f represent the target model.

For a time series it can be represented as $\mathbf{x} = (x_1, \ldots, x_n)$, each element is a numeric value at the corresponding time step. If the original time series $\mathbf{x}$ belongs to class *ori*, the output probability of f is, therefore, $f_{ori}(x)$. While the perturbation can be represented as $\delta(x)$. The magnitude of perturbation ϵ is controlled via a factor β. ϵ is defined as

$$\epsilon = \beta \cdot \left(\frac{1}{T} \sum_{i=1}^{T} (x_i^{max} - x_i^{min}) \right), \tag{2}$$

where x_i^{max} and x_i^{min} separately represent the maximum and minimum value of a time series of the total T items.

Our goal is to find the solution $\delta(x)$ for the following problem:

$$f_{ori}(\mathbf{x} + \delta(\mathbf{x})) \quad < \quad f_{other}(\mathbf{x} + \delta(\mathbf{x})) \tag{3}$$

Subject to $\delta(\mathbf{x})$ is limited in the shapelet interval l

$$\begin{aligned} \|\delta(\mathbf{x})\|_0 &= 1 \\ \|\delta(\mathbf{x})\|_\infty &\leq \epsilon, \end{aligned} \tag{4}$$

where $\|\delta(\mathbf{x})\|_0$ and $\|\delta(\mathbf{x})\|_\infty$ represent the L_0 and L_∞ norms of $\delta(\mathbf{x})$, respectively. That means we want to add perturbations to minimize the confidence score of class *ori*.

In order to visit the target model as little as possible, our method uses predictive model to predict the coordinates of sensitive locations in the test set. By using predictive model we can attack the sensitive locations directly.

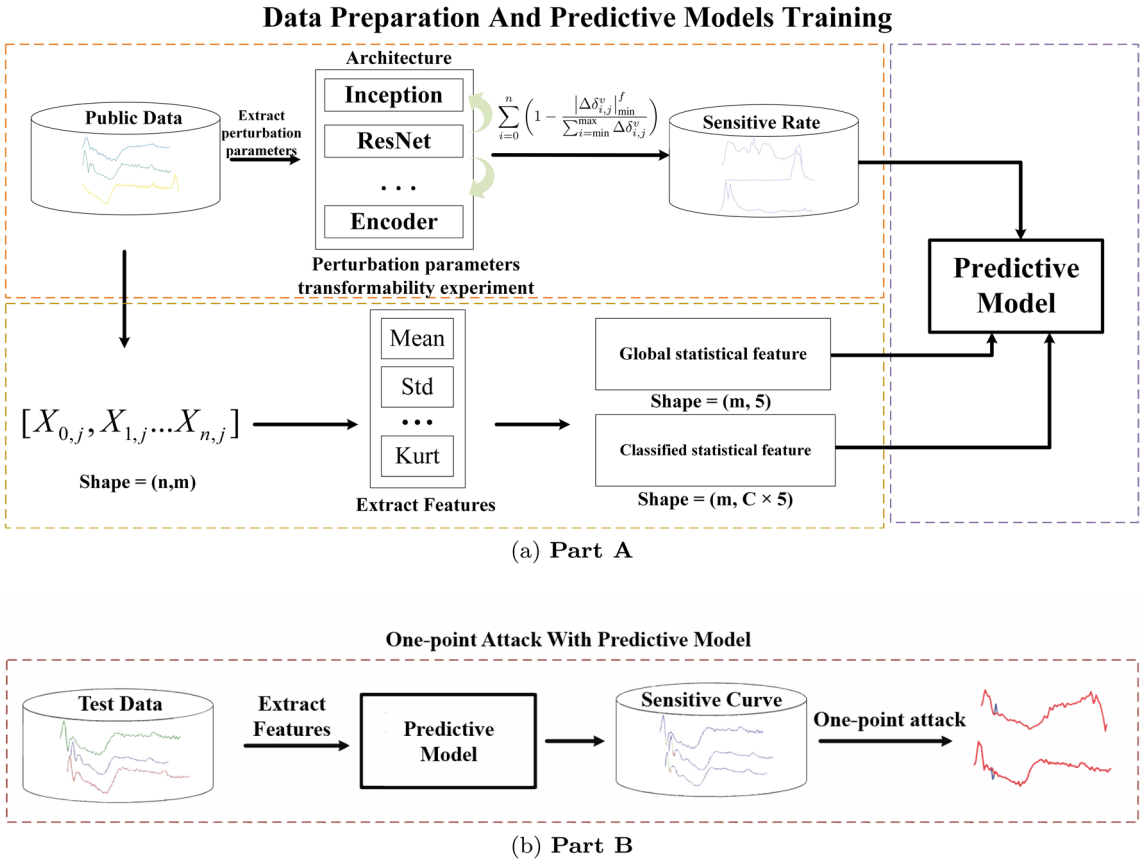


Fig. 3 In **Part A**, in the section of the perturbation parameters transformability experiment, the perturbation parameters generated by the ResNet architecture are input into the Inception and Encoder architectures. To be specific, we have done the test for each pair of architectures, the result show as Fig. 6. In the step of training predictive models, global statistical features and classified statistical features are used to train predictive models respectively. In **Part B**, **sensitive curve** is used as a visual representation of sensitive rate

The main diagram illustration

Before the formal experiment begins, we take the result of one-pixel attack based on differential evolution as the baseline for comparison.

The whole experimental process is shown in Fig. 3. Since it's not available for the architecture of the target model under the black-box attack state, we have to use the public part of the datasets to prepare the models with multiple architectures as substitute models.

To test the transformability of perturbation parameters between different CNNs, we designed the perturbation parameters transformability experiment.

There are six architecture using in this experiment, respectively MLP,CNN,ResNet,Inception,Encoder and FCN. The code for the above architectures is referenced in the Ismail Fawaz et al. (2019)'s open source project.

We obtained the sensitive rate of the typical datasets through the global search algorithm used on multiple substitute models.

Then, we extract the statistical feature of the public part of the typical datasets.

In the step of training predictive model, the extracted statistical feature and sensitive rate are used to train predictive models.

As shown in Fig. 4, after obtaining the predictive models, the sensitive rate of the test set are obtained by predictive models. Then, complete one-point attack by perturbing the sensitive points.

Data preparation and predictive models training

This section introduce the datasets mentioned in this paper and how to extract and use the training set to train the prediction model.

In order to fully demonstrate the effect of the experiment, this experiment selected several typical datasets from the number of classes, length, size of datasets and other aspects as demonstration examples(as shown in Table 1). These datasets are from the UCR/UEA archive (Chen et al. 2015).

There are two data necessary for the training of predictive models, one is the **sensitive rate** of each point,

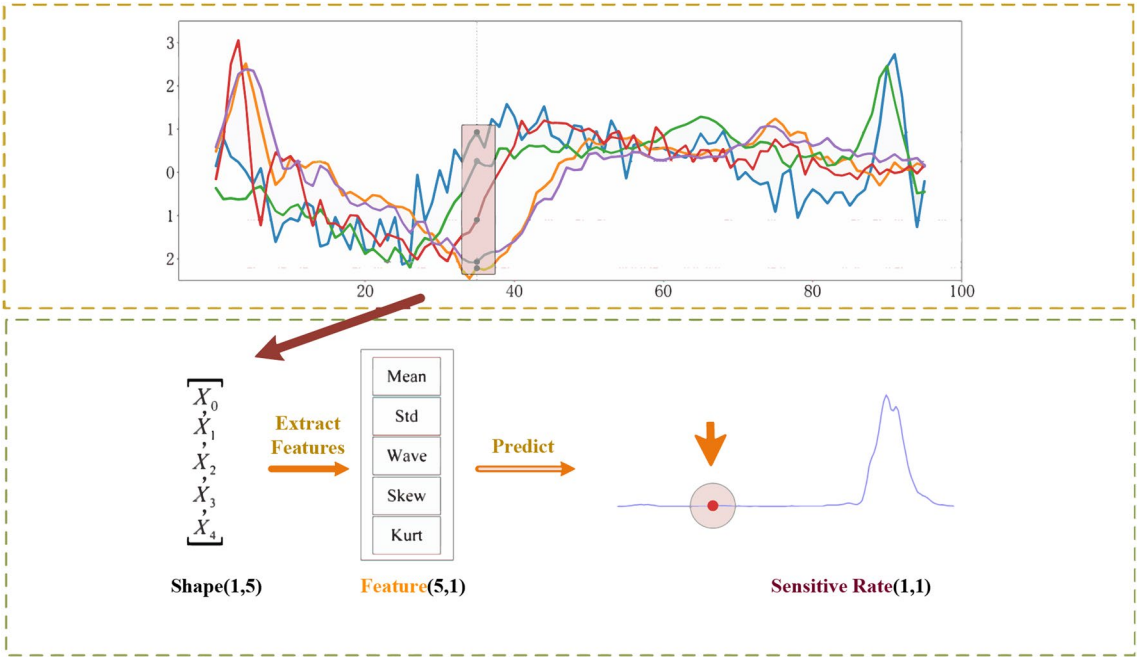


Fig. 4 Take the **35** th time point of ECG200 dataset as an example. The values at the 35th time point of each time series are spliced into a one-dimensional sequence, and the average value, standard deviation, number of peaks and valleys, skewness and kurtosis of the sequence are obtained through feature extraction. In the subsequent process, the sensitive rate of the **35** th time point will be predicted through these features

Table 1 Typical datasets from UCR time series classification archive (POC:PhalangesOutlinesCorrect)

Dataset	Classes	Length	Public	Unpublished
Adiac	37	176	390	391
Car	4	577	60	60
DistalPhalanxTW	6	80	400	139
ECG200	2	96	100	100
ECGFiveDays	2	136	23	861
FaceAll	14	131	560	1690
FISH	7	463	175	175
Meat	3	448	60	60
MedicalImages	10	99	381	760
POC	2	80	1800	858
Strawberry	2	235	613	370
SwedishLeaf	15	128	500	625

and the other is the **statistical feature**. We will introduce them in turn.

The method of one-point attack is to only perturb one point in a time series resulting misclassification of some well-performing models. The perturbation parameters are consist of the parameter records the coordinates of points and the perturbation that can lead to misclassification.

Due to the limitation of black-box state, the attacker is unaware of the model's architecture. Therefore, it is necessary to consider the impact of the substitute models' architectures on the results before the training of substitute model. To address this, we design an experiment to test the transformability of perturbation parameters. That is, testing the **attack success rate (ASR)** of perturbation parameters which extract from the model A on other models. Specifically, the ASR of perturbation parameters extract from the model A on model B can be represented as

$$\mathrm{ASR}_{\mathrm{B}}^{\mathrm{A}} = \frac{\sum_{i=0}^{n} \left[f^{B}_{\max^* \neq ori}(\mathbf{x} + \delta_A(\mathbf{x})) \right]}{n} \tag{5}$$

where $f^{B}_{max* \neq ori}$ represents the highest probability among all classes output by model B when the time series not belong original class. Otherwise, it output is 0. $\delta_A(\mathbf{x})$ represents the perturbation parameters generated by model A.

Sensitive rate

To quantitatively evaluate the vulnerability of a point, we define a degree of difficulty of causing the classifier to produce wrong results after a point is attacked as

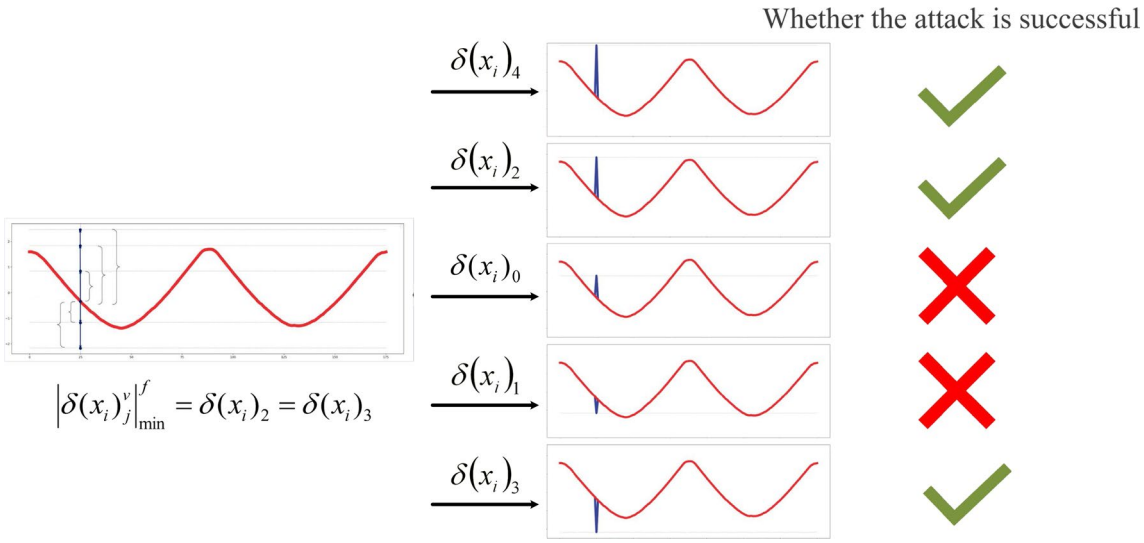


Fig. 5 The figure of using global search algorithm on Adiac dataset ($exp = 1.5, eps = 1$). Only when the perturbation is greater than or equal to $|\delta(\mathbf{x_i})_j^v|_{\min}^f$ can the model misjudge, the minimum perturbation above is a $\delta(\mathbf{x})_2$ or $\delta(\mathbf{x})_3$

sensitive rate. The points with high sensitive rate are called **sensitive points** which means it's sensitive to attack.

The sensitive rate of a point is related to the value of dividing the minimum perturbation that causes models to misjudge ($|\delta(\mathbf{x_i})_j^v|_{\min}^f$) by the sum of the perturbation applied at that point ($\sum_{i=\min}^{\max} \delta(\mathbf{x_i})_j^v$), as shown in equation 6. The higher the sensitive rate of a point, the easier it is to be attacked, and vice versa.

$$\text{Sensitive Rate}_j = \sum_{i=0}^{n} \left(1 - \frac{|\delta(\mathbf{x_i})_j^v|_{\min}^f}{\sum_{i=\min}^{\max} \delta(\mathbf{x_i})_j^v}\right) \tag{6}$$

Perturbation parameters are obtained by global search algorithm which continuously perturbs the value of time series data and record the perturbation parameters when perturbing successfully, as shown in Fig. 5. For a time series $\mathbf{x}$, its solution space of global search algorithm can be represented as

$$\begin{aligned} &(\text{coordinate,value}) \in \{(x,y) \mid x \in [0,n], y \in [x_{exp}^{min}, x_{exp}^{max}]\}, \\ &x_{exp}^{min} = (x^{min} - x^{mid}) \times exp + x^{mid}, \\ &x_{exp}^{max} = (x^{max} - x^{mid}) \times exp + x^{mid} \end{aligned} \tag{7}$$

For the purpose of obtaining more perturbation parameters, we expand the solution space of the global search algorithm in terms of parameter setting, which can improve the generalization ability of the attack. Specifically, we increase the parameter **exp** from **1** to **1.5**.

In our method, there are two statistical features used to train the predictive model. One is global statistical feature, and the other is classified statistical feature.

Global statistical feature

For a coordinate, the values of all time series data in this dataset at this coordinate are spliced into a row of sequences. Calculate the average value, standard deviation, number of peaks and valleys, skewness and kurtosis of this group of sequences, and splice these five values as the statistical feature of this point. This statistical feature is called the global statistical feature of this dataset. By this method, we can get the global statistical features of each point of this dataset.

$$\begin{aligned} feature_i^{global} &= [\phi(E_i(X))], \\ \phi(x) &= [mean(x), std(x), wave(x), skew(x), kurt(x)], \\ E_i(X) &= [X_{0,i}, X_{1,i}, \ldots, X_{n,i}] \end{aligned} \tag{8}$$

where $feature_i^{global}$ represents the statistical feature of **i** th point, *mean* is the average value of the input sequence, *std* is the standard deviation of the input sequence, *wave* is the number of peaks and valleys of the input sequence, *skew* is the skewness of the input sequence, *kurt* is the kurtosis of the input sequence, $E_i(X)$ is a sequence consisting of the values of the **i** th point of each time series.

Classified statistical feature

Separate the time series in the dataset by class, calculate the statistical feature sequence of a point in the time series of each class according to the method of calculating the

global statistical feature, then splice the statistical feature sequence of each class to form a classified statistical feature matrix with width $\mathbf{C} \times \mathbf{5}$, height $\mathbf{n}$, where $\mathbf{C}$ is the number of classes and n is the number of time series.

$$\begin{aligned} feature_i^{classes} &= [\phi(E_i(X^0)), \ldots, \phi(E_i(X^m))], \\ \phi(x) &= [mean(x), std(x), wave(x), skew(x), kurt(x)], \\ f(X^t) = t, E_i(X^t) &= [X_{0,i}^t, X_{1,i}^t, \ldots, X_{n,i}^t], t \in [0, m] \end{aligned} \tag{9}$$

$feature_i^{classes}$ represents the classified statistical feature of the $\mathbf{i}$ th point, f is the classifier, t is the category of time series, m is the number of categories, X^t is the time series of the t category.

One-point attack with predict model

Algorithm 1 One-point attack with predictive model

Input: Testing Set $\mathbf{X}, \mathbf{Y}$,Coordinates of sensitive points $\mathbf{SensitivePoints}$
Output: Accuracy on the original dataset $\mathbf{AccuracyOri}$, Accuracy after attack $\mathbf{AccuracyOPA}$

Initialize matrix which used to record series that never predicted errors(**type:bool**):$\mathbf{AccCorr}$
Input $\mathbf{X}, \mathbf{Y}$ into the trained model ,return the accuracy after attack:$\mathbf{AccuracyOri}$
for arguments in SensitivePoints: do
  $\mathbf{j}$, $\mathbf{RATE} \leftarrow \mathbf{arguments}$
  $\mathbf{XTMP} \leftarrow \mathbf{X.copy}()$
  for $\mathbf{i}, \mathbf{X_i}$ **in** $\mathbf{enumerate(XTMP)}$: **do**
    Determine whether to increase or decrease by judging the location of the attack point : $\mathbf{Direct} \leftarrow \mathbf{GetDirect}(\mathbf{X_i}, \mathbf{j})$
    if Direct is up then
      $\mathbf{XTMP_{i,j}} \leftarrow \mathbf{X_{i,j}} + \delta(\mathbf{x})$
    end if
    if Direct is down then
      $\mathbf{XTMP_{i,j}} \leftarrow \mathbf{X_{i,j}} - \delta(\mathbf{x})$
    end if
  end for
  Input $\mathbf{XTMP}, \mathbf{Y}$ into the trained model, return the accuracy after attack and prediction matrix(**type:bool**):$\mathbf{CorrPreds}$
  $\mathbf{AccCorr} = \mathbf{AccCorr}$ & $\mathbf{CorrPreds}$
end for
$\mathbf{AccuracyOPA} = \mathbf{AccCorr.sum}() \div \mathbf{len}(\mathbf{X})$

Input the features extracted from the testing set into the trained predictive model to predict the sensitive rate of each point in the testing set. One-point attack can be completed by perturbing sensitive points with high sensitive rate.

In detail, the predicted results of the predictive model are recorded in a array of **key − value pairs** named **SensitivePoints**. Each key-value pair consists of a index and a sensitive rate. The key-value pairs are arranged from largest to smallest according to the sensitive rate.

According to the sequence of key-value pairs, all time series in the testing set are modified at a single point according to the current key-value pair. If the point where the index is located is on the wave crest, subtract a perturbation. If the point where the index is located is in wave valley, add a perturbation. The reason for this is in accordance with the Frequency Principle (F-Principle).

The Frequency Principle (F-Principle) was proposed by Xu et al. (2019), which reveals the information that deep neural networks (DNNs) attend to during the classification process. The F-Principle suggests that DNNs often fit target functions from low to high frequencies.

Owing to trained DNNs focus on the high frequency information of data, in order to mislead DNNs, we try to modify the frequency information of the original time series data by perturbing the wave crests and wave valleys.

The value of perturbation is positive, which depends on the span of the dataset, and is generally 30% of the difference between the maximum value of the original dataset and the minimum value.

Due to the key-value pairs are sorted according to the sensitive rate, the higher the key-value pair is, the more vulnerable the point is, and the lower the key-value pair is, the more robust the point pair is.

In order to ensure the efficiency of one-point attack, the experiment in this article only uses the first five key-value pairs for one-point attack.

Experiment results and analyse

Perturbation parameters transformbility experiment

The result of perturbation parameters transformability experiment is shown in Fig. 6, where the y-axis represents the architectures that generating perturbation parameters, and the x-axis represents the architectures for testing. The values of the heatmap represent the ASR of the perturbation parameters generated by the y-axis architectures attack on the x-axis architectures.

From Fig. 6, in most cases, the perturbation parameters generated by different architectures is transformability. However, the ASR of attacking the ResNet architecture using perturbation parameters generated by other architectures are very low, which shows that only a small part of the perturbation parameters generated by the other five architectures can make the ResNet architecture misclassification. In contrast, the ASR in the first row of the heatmap are generally high, which indicates that most perturbation parameters generated by the ResNet architecture can lead to the misclassification of the other architectures. This shows from the side that the robustness of ResNet model is significantly higher than the other five models.

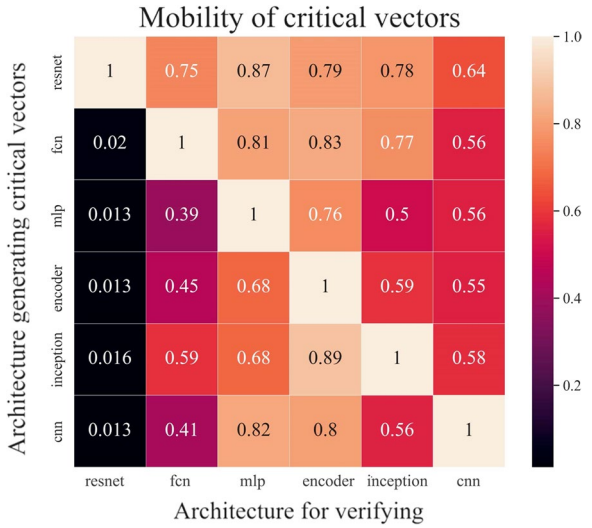


Fig. 6 The result of perturbation parameters transformability experiment. Specifically, the second line represents the ASR that the perturbation parameters generated by the FCN architecture used to attack the ResNet, MLP, Encoder, Inception, CNN architectures. In particular, the ASR of attacking the ResNet architecture using the perturbation parameters generated by other architectures is very low

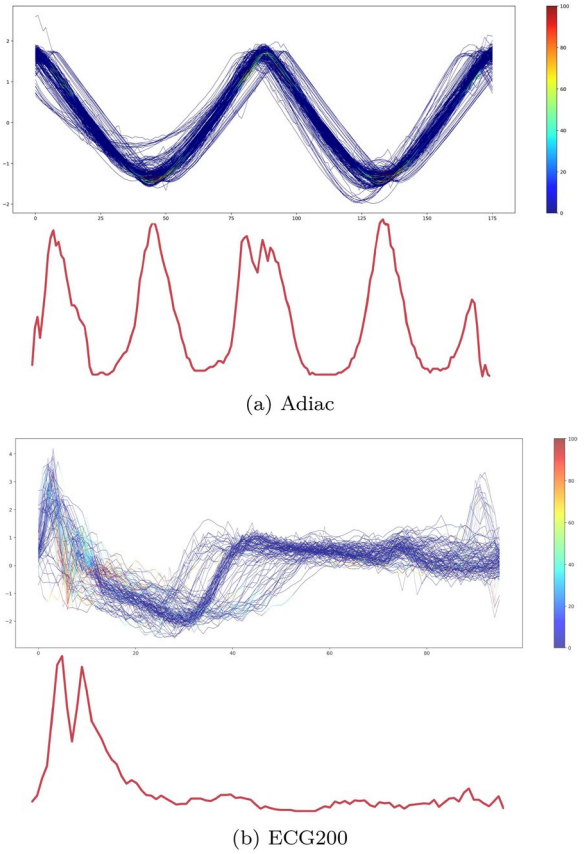


Fig. 7 Aligned figure of class activation diagram and sensitive curve. The coordinates with high sensitive rate in the time series data are usually located at wave crests and wave valleys

The analyse of data preparation

Figure 7 shows the aligned figures of class activation diagram and sensitive curve. The Fig. 8 shows the class activation diagram of each architecture in a certain class on the Adiac dataset. The Fig. 9 shows the sensitive curve of some datasets (both figures are obtained through ResNet architecture). For CAM, the highlighted part of the class activation diagram is the vulnerable part. The data plotted in Fig. 7,8,9 is generated from perturbed data produced by the global search algorithm.

Through the analysis of Fig. 8, we can further understand the reason of the special phenomenon in Fig. 6.

Only the highlighted positions of the ResNet architecture are very concentrated and distributed in a point shape, while the highlighted positions of the activation diagrams of other architectures are distributed in a strip shape. This shows that to increase the ASR of the ResNet architecture, it is necessary to focus on attacking its highlighted location. Because the highlighted area of ResNet is very small and concentrated, which leads to better robustness of ResNet compared with the other five architectures. Besides, the global search algorithm proves that the number of successful attacks on ResNet architecture is the lowest among all architectures.

Figure 7 shows the coordinates with high sensitive rate in the time series data are usually located at the peaks or troughs. If at a certain location in a dataset, the more common wave crests and wave valleys appear, the more sensitive to perturbation. This phenomenon can be understood as if the wave form of this position vulnerable they are similar in most time series, then a perturbation is added to this position, which results in the value form of this position exceeding the value range under normal conditions and leads to misjudgment of classifier. What's more, it can also be explained by the theory of F-Principle. For DNNS, as iteration steps of training increase, the high-frequencies are fitted, the locations of the wave crests and wave valleys are usually the location with high-frequency information.

Table 2 shows the proportion of the perturbation parameters whose values are within the range of the original time series data values in the perturbation parameters obtained using the global search algorithm ($exp = 1.5$). The table shows that the perturb in the original value range only accounts for a small part of the total perturbation parameters, in the field of adversarial attack, properly expanding the value range of perturbation can effectively improve the ASR.

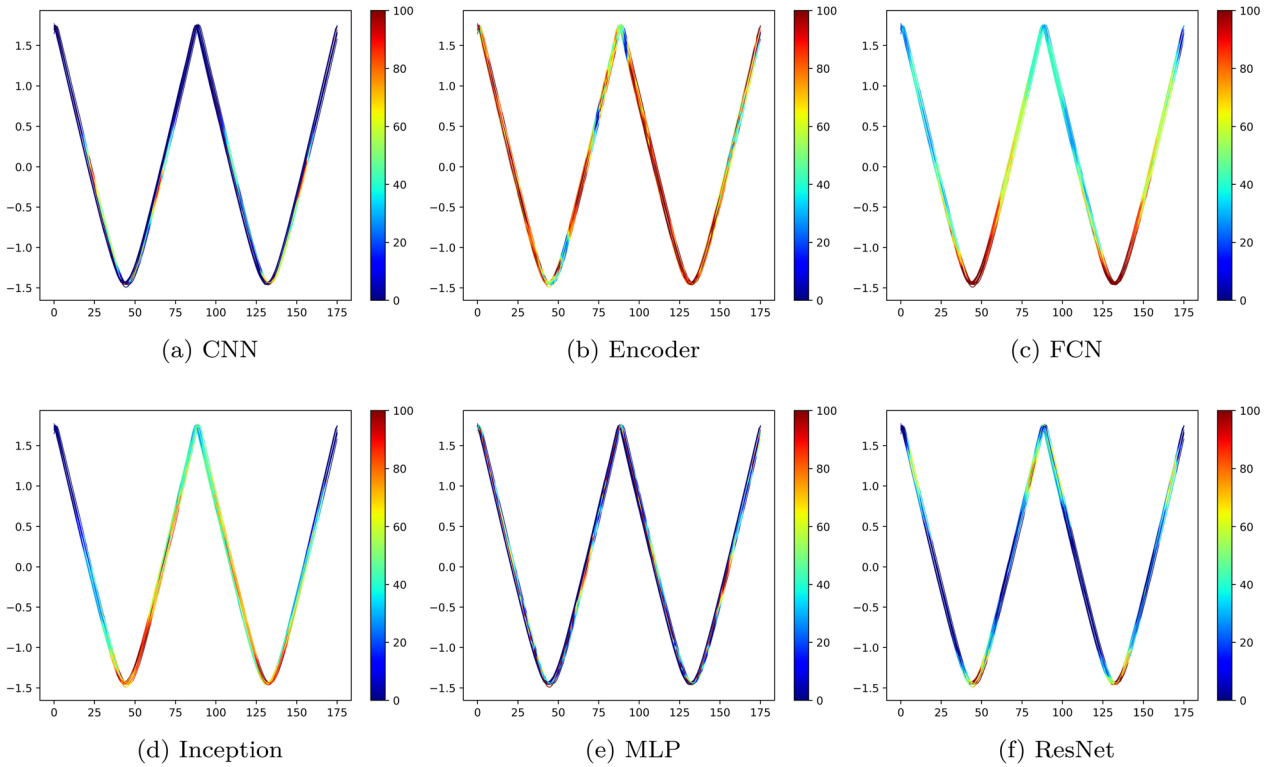


Fig. 8 CAM of a class of Adiac dataset on each architecture. Only the highlighted positions of the ResNet architecture are very concentrated and distributed in a point shape, while the highlighted positions of the activation diagrams of other architectures are distributed in a strip shape. (Take Adiac dataset as example)

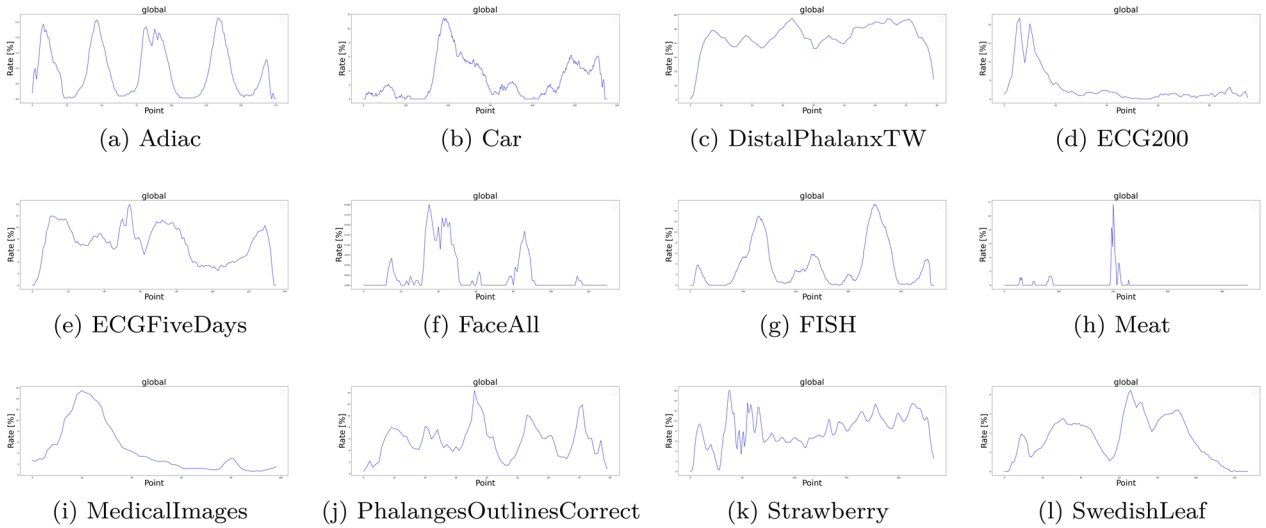
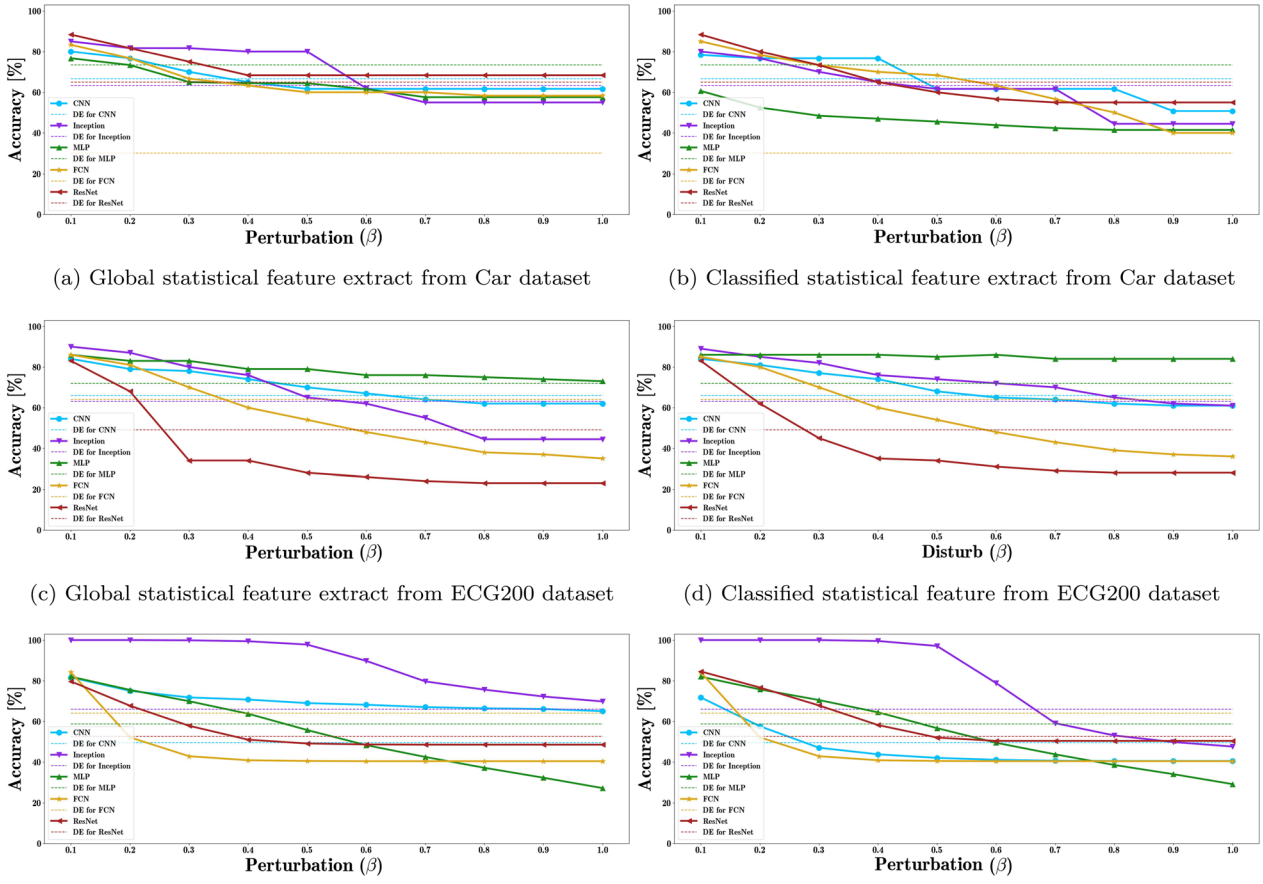


Fig. 9 Sensitive curves of several datasets were plotted, which showed that the vulnerability levels of these datasets varied at different time points. Attacking at locations with high sensitive rates can result in higher attack success rates

Table 2 The proportion (%) of the perturbation parameters whose values are within the range of the original time series data values in the perturbation parameters obtained using the global search algorithm (*exp* = 1.5)

Architecture	Adiac	Car	ECG200	ECGFivedays	Meat	FaceAll
CNN	0.21	0.17	0.33	0.46	0.45	0.34
Encoder	0.15	0.06	0.15	0.12	0.41	0.06
FCN	0.08	0.24	0.31	0.23	0.27	0.02
Inception	0.11	0.17	0.20	0.10	0.38	0.02
MLP	0.23	0.48	0.44	0.36	0.53	0.11
ResNet	0.05	0.00	0.32	0.13	0.66	0.00



(a) Global statistical feature extract from Car dataset

(b) Classified statistical feature extract from Car dataset

(c) Global statistical feature extract from ECG200 dataset

(d) Classified statistical feature from ECG200 dataset

(e) Global statistical feature from ECGFiveDyas dataset

(f) Classified statistical feature from ECGFiveDyas dataset

Fig. 10 Accuracy of each architecture under different factors (β) for perturbing. The accuracy of architectures decreases as the β increases. (The dashed line represents the accuracy of the architectures after being attacked by one-point attack based on evolutionary algorithm ($\beta = 1$))

The result of one-point attack

Figure 10 shows that the accuracy of each architecture under different perturbation magnitudes. It can be observed from the figure that the accuracy of architectures decreases as the magnitude of the perturbation increases. The accuracy of the CNNs under the one-point attack based data is basically lower than that of the one-point attack with DE (The perturbation amplitude of one-point attack based on evolutionary algorithm is 1).

Table 3 shows that the size of the dataset and the quantity of perturbed points that impact the effect of the attack. For instance, when the parameter proportion is set to 0.3, then there are 30% samples of the original dataset used to obtain key-value pairs. When the parameter

Table 3 This table illustrates the accuracy of deep learning models under varying sizes of datasets and quantities of perturbed points

Dataset name	Accuracy	Proportion	Number of perturbed points				
			2	3	5	7	10
Adiac	83.12	1.0	59.56	61.95	34.34	42.34	32.25
		0.8	59.9	62.05	35.27	43.18	32.48
		0.5	68.57	66.79	35.79	43.94	35.25
		0.3	68.8	67.26	45.78	43.99	39.9
Car	93.33	1.0	83.34	84.26	73.87	79.48	88.5
		0.8	92.28	85.07	82.19	80.27	89.11
		0.5	92.73	92.75	85.59	80.49	89.92
		0.3	93.33	93.33	90	90	90
ECGFiveDays	96.17	1.0	89.61	79.37	82.37	73.46	62.12
		0.8	90.19	83.73	85.68	78.8	69.04
		0.5	90.48	89.66	85.94	79.17	73.12
		0.3	92.92	89.9	86.06	83.39	81.65
FaceAll	85.50	1.0	81.74	77.09	74.95	78.73	81.6
		0.8	82.4	80.16	82.46	78.75	81.65
		0.5	83.21	81.06	82.68	78.89	81.89
		0.3	84.2	83.79	83.31	83.02	82.43
FISH	97.71	1.0	97.64	79.51	93.9	84.57	89.05
		0.8	97.88	80.22	94.43	90.24	89.82
		0.5	98.02	88.97	96.64	90.63	89.93
		0.3	98.29	98.29	97.14	93.71	90.86
Meat	98.33	1.0	77.73	79.95	73.74	67.07	58.38
		0.8	78.42	80.83	74.56	67.87	63.68
		0.5	82.95	81.24	74.63	74.87	65.98
		0.3	91.67	81.67	75	75	75
MedicalImages	76.18	1.0	64.29	66.22	59.95	54.67	50.25
		0.8	65.23	70.87	60.48	55.38	50.99
		0.5	75.05	71.82	60.67	56.14	54.85
		0.3	75.39	72.5	65.26	61.05	56.97
PhalangesOutlinesCorrect	85.66	1.0	77.41	68.75	49.22	46.01	46.97
		0.8	78.35	69.01	58.8	46.05	47.89
		0.5	80.92	69.76	59.56	51.5	48.42
		0.3	81.35	72.96	60.02	51.98	48.95
Strawberry	96.25	1.0	86.07	81.26	77.76	77.09	78.84
		0.8	93.52	91.15	81.25	78.03	79.49
		0.5	94.22	91.8	85.91	82.77	80.22
		0.3	94.78	91.84	88.09	83.2	80.91
SwedishLeaf	95.36	1.0	94.53	88.69	82.86	79.27	77.05
		0.8	94.6	89.16	92.11	89.25	77.68
		0.5	94.72	89.61	92.58	89.42	82.53
		0.3	94.72	94.4	92.96	90.08	86.72

"accuracy" refers the accuracy of the dataset before perturbed, "proportion" refers to the proportion of the dataset available for use during attacks, "number of perturbed points" refers the quantity of points that have been perturbed.(β is set to 0.1)

proportion is fixed, the accuracy decreases as the number of perturbation points increases. It is intuitive that when the number of perturbation points increases the success rate of the attack should rightfully increase, which also means that the accuracy of the model on the counter sample decreases. When the number of perturbed points is fixed, the accuracy decreases as parameter proportion decreases. This phenomenon illustrate that the statistical

features obatined from the larger size of dataset have more information of the original dataset which is more beneficial for the attacker.

Discussion and conclusion

We propose a new black-box attack based on data and attack according to the high-frequency component of time series.

Compared to the evolutionary algorithm-based attack method, we are able to generalise to a wider range of application scenarios. Our approach has a higher ASR under the same perturbation magnitude constraints. The method of one-point attack can be used in extreme environments with minimal modification of the original data. This attack has the following features compared to other black-box attacks:

- Based on data.
- Fewer submissions.
- Attack the sensitive points directly

However, our approach does not achieve good performance on some datasets, such as ECG200 and DistalPhalanxTW, which may be because my method only focuses on statistical features, while the information related to sensitivity in these datasets is also contained in the frequency domain and time domain. In future experiments, we plan to increase the number and types of features. We believe that increasing the number and types offeatures can improve the accuracy of predictions and enable the development of more data-based adversarial attack methods.

Moreover, with the progress of explainable artificial intelligence, we hope we can rely on the work of previous researchers, proposing more theoretical work content. Additionally, it would be interesting to investigate the use of more sophisticated substitute models, such as generative models or deep neural networks, to improve the transferability of the attack across different target models.

In addition to theoretical research, we also hope to find practical applications for this achievement in the real world. The attack could be extended to consider other types of time series data beyond those studied in this work, such as images, audio or video. Overall, we believe that our work lays the foundation for further research into the data based black-box attack method.

Author contributions

YG: Designing computer programs; implementation of the computer code and supporting algorithms; Testing of existing code components. Writing - Original Draft. ZY: Ideas; formulation or evolution of overarching research goals and aims; Acquisition of the financial support for the project leading to this publication; Design of methodology;Supervision; Review and Editing. LZ: Data Curation; Data analysis. YW: Visualization; Translate. CL: Review and Editing. All authors read and approved the final manuscript.

Funding

This work is supported by the Key Program of National Natural Science Foundation of China (No. 61832004) and International Cooperation and Exchange Program of National Natural Science Foundation of China (Grant no. 62061136006).

Declarations

Competing interests

All authors disclosed no relevant relationships.

Received: 15 March 2023 Accepted: 5 July 2023
Published online: 05 November 2023

References

Abdelfattah SM, Abdelrahman GM, Wang M (2018) Augmenting the size of eeg datasets using generative adversarial networks. In: 2018 International joint conference on neural networks (IJCNN), pp 1–6. https://doi.org/10.1109/IJCNN.2018.8489727

Brendel W, Rauber J, Bethge M (2017) Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. arXiv preprint arXiv:1712.04248

Carlini N, Wagner D (2017) Towards evaluating the robustness of neural networks. In: 2017 IEEE symposium on security and privacy (SP), pp 39–57. IEEE

Chen Y, Keogh E, Hu B, Begum N, Bagnall A, Mueen A, Batista G (2015) The UCR time series classification archive. www.cs.ucr.edu/eamonn/time_series_data/

Fawaz HI, Forestier G, Weber J, Idoumghar L, Muller P-A (2018) Evaluating surgical skills from kinematic data using convolutional neural networks. CoRR, abs/1806.02750 arxiv:1806.02750

Fawaz HI, Forestier G, Weber J, Idoumghar L, Muller P-A (2019) Adversarial attacks on deep neural networks for time series classification. In: 2019 international joint conference on neural networks (IJCNN), pp 1–8. IEEE

Goodfellow I, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples

Goodfellow IJ, Shlens J, Szegedy C (2015) Explaining and harnessing adversarial examples

He Z, Wang W, Dong J, Tan T (2022) Transferable sparse adversarial attack. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 14963–14972

Hu Z, Huang S, Zhu X, Sun F, Zhang B, Hu X (2022) Adversarial texture for fooling person detectors in the physical world. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 13307–13316

Fawaz HI, Forestier G, Weber J, Idoumghar L, Muller P-A (2019) Deep learning for time series classification: a review. Data Min Knowl Disc 33(4):917–963

Kahla M, Chen S, Just HA, Jia R (2022) Label-only model inversion attacks via boundary repulsion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 15045–15053

Lin JX, Lei LY, Xiangyu Z (2020) Black-box adversarial sample generation based on differential evolution. J Syst Softw 170:110767

Linardatos P, Papastefanopoulos V, Kotsiantis S (2021) Explainable AI: a review of machine learning interpretability methods. Entropy. https://doi.org/10.3390/e23010018

Luo C, Lin Q, Xie W, Wu B, Xie J, Shen L (2022) Frequency-driven imperceptible adversarial attack on semantic similarity. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 15315–15324

Ma T, Xiao C, Wang F (2018) Health-ATM: a deep architecture for multifaceted patient health record representation and risk prediction, pp 261–269. https://doi.org/10.1137/1.9781611975321.30

Moosavi-Dezfooli S-M, Fawzi A, Frossard P (2016) Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2574–2582

Nguyen H, Kieu LM, Wen T, Cai C (2018) Deep learning methods in transportation domain: a review. IET Intell Transp Syst
Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A (2016a) The limitations of deep learning in adversarial settings. In: 2016 IEEE European symposium on security and privacy (EuroS &P), pp 372–387. IEEE
Papernot N, McDaniel PD, Goodfellow IJ (2016b) Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. CoRR. arxiv:abs/1605.07277
Papernot N, McDaniel P, Goodfellow I, Jha S, Celik ZB, Ananthram S (2017) Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia conference on computer and communications security, pp 506–519
Storn R, Price K (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341
Su J, Vargas DV, Sakurai K (2019) One pixel attack for fooling deep neural networks. IEEE Trans Evolut Comput 23(5):828–841. https://doi.org/10.1109/TEVC.2019.2890858
Christian S, Wojciech Z, Ilya S, Joan B, Dumitru E, Ian G, Rob F (2014) Intriguing properties of neural networks. Input-output mapping;Linear combinations;Prediction errors;Semantic information;Specific nature;State-of-the-art performance;Visual recognition, Banff, AB, Canada
Tan CW, Webb GI, Petitjean F(2017) Indexing and classifying gigabytes of time series under time warping. In: Proceedings of the 2017 SIAM international conference on data mining, pp 282–290. SIAM
Tobiyama S, Yamaguchi Y, Shimada H, Ikuse T, Yagi T (2016) Malware detection with deep neural network using process behavior. In: 2016 IEEE 40th annual computer software and applications conference (COMPSAC), volume 2, pp 577–582. IEEE
Wang W, Sun J, Wang G (2020) Visualizing one pixel attack using adversarial maps, pp 924–929, Shanghai, China. https://doi.org/10.1109/CAC51589.2020.9327603
Wei X, Guo Y, Li B (2021) Black-box adversarial attacks by manipulating image attributes. Inf Sci 550:285–296. https://doi.org/10.1016/j.ins.2020.10.028
Xu Z-QJ, Zhang Y, Luo T, Xiao Y, Ma Z (2019) Frequency principle: Fourier analysis sheds light on deep neural networks. CoRR, arxiv:abs/1901.06523
Yang W, Yuan J, Wang X, Zhao P (2022) Tsadv: black-box adversarial attack on time series with local perturbations. Eng Appl Artif Intell 114. ISSN 09521976. https://doi.org/10.1016/j.engappai.2022.105218

Publisher's Note