Black-Box Adversarial Attack on Time Series Classification

Daizong Ding[1], Mi Zhang[1], Fuli Feng[2], Yuanmin Huang[1], Erling Jiang[1], Min Yang[1*]

[1] School of Computer Science, Fudan University, China
[2]University of Science and Technology of China
{17110240010@, mi_zhang@, yuanminhuang21@m., eljiang21@m., m_yang@}fudan.edu.cn
fulifeng93@gmail.com

Abstract

With the increasing use of deep neural networks (DNN) in time series classification (TSC), recent work reveals the threat of adversarial attack, where the adversary can construct adversarial examples to cause model mistakes. However, existing research on the adversarial attack of TSC typically adopts an unrealistic white-box setting with model details transparent to the adversary. In this work, we study a more rigorous black-box setting with attack detection applied, which restricts gradient access and requires the adversarial example to be also stealthy. Theoretical analyses reveal that the key lies in: estimating black-box gradient with diversity and non-convexity of TSC models resolved, and restricting the ℓ_0 norm of the perturbation to construct adversarial samples. Towards this end, we propose a new framework named *BlackTreeS*, which solves the hard optimization issue for adversarial example construction with two simple yet effective modules. In particular, we propose a tree search strategy to find influential positions in a sequence, and independently estimate the black-box gradients for these positions. Extensive experiments on three real-world TSC datasets and five DNN based models validate the effectiveness of BlackTreeS, e.g., it improves the attack success rate from 19.3% to 27.3%, and decreases the detection success rate from 90.9% to 6.8% for LSTM on the UWave dataset.

1 Introduction

Time series classification has become one of the central themes of modern data mining with the increase in temporal data availability. It aims to classify sequential data into different categories (Yang and Wu 2006; Esling and Agon 2012; Gupta et al. 2020), e.g., forecasting the direction of stock market movement (Zhan et al. 2018) or detecting whether an electronic health record is anomalous (Che et al. 2017). DNNs such as convolutional neural networks (CNN) (Cui, Chen, and Chen 2016), recurrent neural networks (RNN) (Smirnov and Nguifo 2018) and self-attention network (Zerveas et al. 2021) shows superior performance on TSC. This is because DNN can capture complex temporal patterns with the aid of its deep non-linear structure (Gamboa 2017; Wang, Yan, and Oates 2017). However, DNN is also vulnerable to *adversarial attacks* (Goodfellow, Shlens, and Szegedy 2014). The adversary can construct *adversarial examples* by adding small perturbations on a natural example, leading to wrong prediction and severe loss (e.g., approval on risky loan applications and evasion of network flow anomaly detection) (Cartella et al. 2021).

Existing work on investigating the adversarial attack on TSC models (Karim, Majumdar, and Darabi 2020) focuses on the construction of confusing adversarial examples. For instance, the work (Fawaz et al. 2019) proposes to create adversarial examples by the fast gradient sign method (FGSM). Despite achieving a high attack success rate, existing researches mainly have two limitations. On one hand, they typically adopt a *white-box setting*, which assumes the leakage of model details, including the model structure and parameters, to adversaries (Meng et al. 2019). This assumption is unrealistic in TSC applications where adversaries can only access model prediction (Gupta et al. 2020). On the other hand, the existing research largely emphasizes the attack success rate but ignores *attack stealthiness*, leading to easily identifiable adversarial examples (Belkhouja and Doppa 2020). For instance, a simple auto-encoder can filter out more than 95% adversarial examples (Wang et al. 2020).

In light of these limitations, we investigate a more rigorous *black-box setting* with two constraints on the construction of adversarial examples: (1) using only model predictions; and (2) evading the common attack detection. Despite the success of black-box attacks in other applications (Dong et al. 2019; Wei, Yan, and Li 2022), e.g., the substitute model and score-based approaches in image classification (Guo et al. 2019), they cannot satisfy the two constraints in TSC. The reasons are twofold: *(1) diversity and non-convexity of TSC models,* on one side, the target model can be a CNN, RNN or self-attention model (Vaswani et al. 2017) with largely different classification behaviors, making it difficult to train a substitute model that mimics the behaviors of the target one. On the other hand, through theoretical and empirical study, we find that the non-convexity of RNN and self-attention model often makes the score-based black-box gradient approximation inaccurate. *(2) The low-dimensional data manifold:* sequential data typically lies in a low dimensional manifold (Rodrigues, Congedo, and Jutten 2018), making it sensitive to minor feature changes. That is, a very small perturbation will push the sequence far away from the natural example manifold (see Fig. 1), resulting in the de-

*Corresponding authors: Mi Zhang and Min Yang

tection of adversarial examples. Therefore, it is non-trivial to keep adversarial examples stealthy.

In this work, we propose a novel framework called **BlackTreeS**, which consists of two new modules to tackle the challenges above, respectively. To construct effective adversarial examples, we devise an independent approximation module to accurately approximate the black-box gradient of each input position. The module overcomes the estimation errors of conventional gradient approximation methods caused by the non-convexity of TSC models. Furthermore, we theoretically reveal the relation between the ℓ_0 norm of perturbations and the natural example manifold and propose to minimize the number of manipulated positions in the sequence. To deal with the extra complexity brought by the ℓ_0 penalty, we propose a tree position search module to fetch influential positions. Owing to the logarithmic property, the module well caps the number of model prediction queries, which is essential in practice[1].

Through the experimental results on three real-world applications and five DNN based classifiers (one self-attention, two RNN and two CNN models), we validate the effectiveness and stealthiness of our attack. The BlackTreeS largely outperforms the existing black-box attack techniques that are directly applied to TSC w.r.t the attack success rate. Meanwhile, the stealthiness is also certificated where the detection success rate is reduced from 100.0% to 6.8% for RNN-based classifiers in some cases. In summary, the main contributions of this work are:

- We study a new black-box setting for the attack on TSC and proposed a new framework **BlackTreeS** to construct effective and stealthy adversarial examples.
- We design the independent approximation and tree position search modules to precisely estimate black-box gradients and query-efficiently optimize the ℓ_0 norm.
- We conduct extensive experiments on three applications and five DNN models of TSC, validating the effectiveness and stealthiness of our attack.

2 Technical Background

Time Series Classification. This work focuses on real-valued time series data due to its wide applications (Yang and Wu 2006; Esling and Agon 2012; Gupta et al. 2020) such as recognizing the abnormal network traffic flow (Hayes and Danezis 2016) and predicting extreme events in climate data (Ding et al. 2019). Suppose there are N labeled sequences $\mathcal{D} = \{X^{(i)}, y^{(i)}\}_{i=1}^{N}$, where $X^{(i)} \in \mathbb{R}^{T\times M}$ and $y^{(i)}$ denote the input sequence and label respectively. At each timestamp $t \in [1, T]$, $x_t^{(i)}$ describes the temporal feature, the daily climate data. The goal of TSC is to learn the mapping between the sequence $X^{(i)}$ and its label $y^{(i)}$, a classifier f_θ where θ denotes model parameters. Owing to the deep and non-linear structure, DNN can recognize the complex temporal pattern embedded in X (Gamboa 2017; Wang, Yan, and Oates 2017). There are three representative structures of DNN based TSC models: CNN (Cui, Chen, and Chen 2016), RNN (Smirnov and Nguifo 2018) and self-attention network (Zerveas et al. 2021), which are all selected as the target models of this work.

Adversarial Attack on TSC Models. The goal of the adversarial attack is to construct an adversarial example from a natural one that can cause a targeted classification $\tilde{y}$. As to perform adversarial attack on TSC (Rathore et al. 2020), given a natural example X, the target is to optimize a perturbation $\delta \in \mathbb{R}^{T\times M}$, which is typically formulated as:

$$\min_{\delta} \; L(\theta; X+\delta, \tilde{y}), \quad \text{s.t. } \|\delta\|_\infty \le \epsilon, \tag{1}$$

where $L(\theta; X+\delta, \tilde{y})$ is the loss function of the classifier, the cross-entropy loss between $f_\theta(X+\delta)$ and $\tilde{y}$; $\|\delta\|_\infty$ is the infinity norm of the perturbation, which restricts the maximum value of δ to be smaller than a threshold ϵ. Similar to the optimization of DNN parameters, the key to calculating the optimal perturbation lies in the gradient of the input $\nabla_X L(\theta; X, \tilde{y})$. For instance, the FGSM algorithm (Goodfellow, Shlens, and Szegedy 2014) directly takes the sign of $\nabla_X L(\theta; X, \tilde{y})$ to create the perturbation, while the PGD algorithm applies an iterative framework (Madry et al. 2017). Existing researches investigate the adversarial attack on TSC models under a white-box setting, where the gradient is assumed to be accessible to the adversary.

Black-Box Adversarial Attack. A more realistic black-box setting has been studied in other tasks such as image classification (Guo et al. 2019) and video recognition (Wei et al. 2020), where the adversary only knows the prediction $f_\theta(X)$. As a real-world example in TSC, the victim DNN could be an RNN on a lending platform that classifies the transaction history to judge loan application. An attacker could manipulate its recent records to obtain a loan approval by several queries. Under this setting, the challenge is the black-box estimation of the gradient $\nabla_X L(\theta; X, \tilde{y})$. Two typical approaches of gradient estimation are substitute model and score-based methods.

- **Substitute Model:** these approaches mimic the target model with a local substitute model $\hat{f}_\theta$, which is trained by regarding the prediction $f_\theta(X)$ as labels. They can use the substitute model to obtain the approximated gradients $\nabla_X L(\theta; X, \tilde{y})$, and transfer the created adversarial example $\tilde{X}$ to the black-box model (Papernot et al. 2017; Liu et al. 2016).
- **Score-Based Methods:** these approaches estimate the gradient by numerical approximation:

$$\nabla_X L(\theta; X, \tilde{y}) \approx \frac{1}{B}\sum_{b=1}^{B}[r(X+\boldsymbol{\eta}^{(b)}) - r(X)] \cdot [\boldsymbol{\eta}^{(b)}]^{-1}, \tag{2}$$

where $\boldsymbol{\eta}^{(b)} \in \mathbb{R}^{T\times M}$ are small perturbations, and the loss calculation are repeated B times. Different algorithms generate the perturbation with different strategies. For instance, the NES (Ilyas et al. 2018) samples $\boldsymbol{\eta}^{(b)}$ by $\beta \cdot \tilde{\eta}$, where $\tilde{\eta}$ is sampled from Gaussian distribution. The SPSA samples $\tilde{\eta}$ from Rademacher distribution (Uesato et al. 2018) and the AutoZOOM samples $\tilde{\eta}$ from unit Euclidean sphere (Chen et al. 2017; Tu et al. 2019).

[1]Practical TSC services typically adopt rate-limiting and may also charge by frequency.

3 Problem Analysis

3.1 Black-Box Gradient Estimation

For launching an effective adversarial attack, a seemingly reasonable solution for estimating black-box gradients in TSC is to directly apply the existing substitute model or score-based methods. We thus conduct a pilot study on the UWave dataset (see Sec. 5 for settings), where the performance of these methods is unsatisfactory. In particular, the attack success rate (ASR) of the substitute model is only 2.3%, which means the estimation of gradients is inaccurate. The main reason lies in the diversity of the classifier, which largely increases the difficulty of mimicking its behavior. Score-based methods achieve higher ASR than the substitute model in most cases. However, their performance suffers a significant drop when the target model is changed from CNN (ASR = 100%) to RNN (ASR = 18%) or self-attention model (ASR = 67%). To construct effective adversarial examples in TSC, we have to first answer the question: *why do current score-based methods fail for RNN and self-attention models?*

To shed light on the root reason for the problem, we focus on the non-convex optimization based on the black-box gradient estimation. Then we develop the following theorem:

Theorem 1. *Suppose the non-convex and Lipschitz continuous function $r(x) : \mathbb{R}^D \to \mathbb{R}$ is optimized with the gradients estimated by Eq. 2, and also suppose the maximal norm of the gradients is $\|\nabla_x r\|_*$. Then we have,*

$$r(x^{(I)}) - r(x^{(0)}) \leq \frac{\sqrt{6\alpha}}{8} \sum_{l=0}^{I-1} \left(\frac{D+4}{2} \sqrt{\frac{3\alpha}{2}} \|\nabla_x r\|_* - \|\nabla r(x^{(l)})\| \right), \tag{3}$$

where $x^{(l)}$ is the variable at l-th iteration, α is the step size, and I is the number of the iteration.

The detailed proof is provided in Appendix A. As we can see from the theorem, the upper bound of the optimization error is mainly determined by two factors: the $\|\nabla_x r\|_*$ and the dimension D. When the $r(x)$ is the adversarial attack goal given a trained neural network, we could leverage Theorem 1 to analyze the attack effectiveness of Eq. 2. Specifically, CNN often has a low $\|\nabla_x r\|_*$ since the widely used convolutional operation and ReLU activation function often behave in a linear pattern (Virmaux and Scaman 2018). However, RNN and self-attention models often encounter a larger $\|\nabla_x r\|_*$ due to the numerous non-linear operations such as the $\tanh$, softmax and the product term between variables (e.g., $x_1 \cdot x_2$) (Erichson et al. 2020), which often leads to the problem of gradient explosion (Pascanu, Mikolov, and Bengio 2013; Nguyen and Salazar 2019),

$$h_t = (1 - \sigma(W_z \cdot [h_{t-1}, x_t])) \odot h_{t-1} \qquad \text{(RNN)}$$
$$h_t = \text{Attention}(x_t; [x_1, \cdots, x_T]) \qquad \text{(Self-attention)}$$

Therefore, the commonly used black-box gradient estimation methods often lead to a larger upper bound when we leverage them to generate adversarial examples.

3.2 Low-Dimensional Manifold

The second difficulty is that the detection of adversarial examples is much easier in TSC as compared to other applications such as image classification. Simple defense strategies, e.g., leveraging the auto-encoder to compute the reconstruction error and regard sequences with large errors as adversarial examples (Wang et al. 2020), can successfully detect 95% adversarial examples in some TSC tasks. We postulate the reason is the distinct property of time series data and analyze it based on the data manifold (Dey 2006).

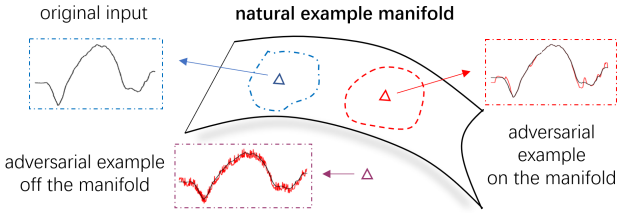

Figure 1: The demonstration of the natural example manifold. Better viewed in color.

Natural Example Manifold. Given a set of sequences $X \in \mathbb{R}^{T \times M}$, and let $D = T \times M$, the sequences with specific temporal patterns lie in a *natural example manifold* $\mathcal{M}^d$ with dimension of d, where $d \ll D$. This is because the values that consist of a sequence are not arbitrarily determined, or it will lead to noise-like data with no semantic meanings. Instead, they actually live on a d-dimensional subspace. The successful detection of adversarial examples is that they often run out of the natural example manifold (Wang et al. 2020). Formally, the percentage of the manifold $\mathcal{M}^d$ covered by the space of adversarial examples is proportional to $(\frac{2\pi}{D-d})^{d/2}$ (Khoury and Hadfield-Menell 2018)[2], where a smaller d leads to less coverage. Given that the dimension d in TSC is much smaller than other modals of data such as images or sentences (Rodrigues, Congedo, and Jutten 2018), which commonly possess more semantic information (Pless and Souvenir 2009), the adversarial examples are more likely to be far from the natural example manifold in this problem. Existing methods propose to leverage the auto-encoder to learn the natural example manifold, therefore the adversarial examples can be easily detected. In a word, it is essential to restrict adversarial examples within the natural example manifold to enhance their stealthiness.

4 Our Approach

We propose a framework called Black-box Adversarial Attack by Tree Search (**BlackTreeS**) for the effectiveness and stealthiness of black-box adversarial attack on TSC.

4.1 ℓ_0 Normalization

We first consider how to create adversarial examples that lie on the natural example manifold. Formally, suppose $\mathcal{M} \subset \mathbb{R}^D$ is a d-dimensional manifold embedded in $\mathbb{R}^D$, $X \in \mathcal{M}$ is a sample lie on the manifold and δ is the perturbation on X. Then the goal is to make the adversarial example $\tilde{X} = X + \delta$ lie on $\mathcal{M}$. To this end, we pay attention to the *tangent space* at X, which is the set of vectors along the manifold starting from X. Suppose the basis of the tangent

[2]For details of the statement, please refer to Appendix B.

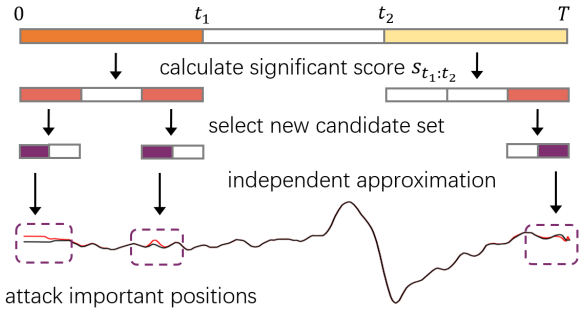

Figure 2: The tree position search.

space is denoted as $v_k \in \mathbb{R}^D$, $k = 1, \cdots, d$, that is, any vector in the tangent space could be represented by the linear combination of basis. To let $\tilde{X}$ lie on the manifold, we propose to minimize the following objective function,

$$\min_{\delta} \sum_{k=1}^{d} \|\delta\|_2 \cdot \cos\langle \delta, v_k \rangle \quad \text{s.t. } \|\delta\|_\infty \leq \epsilon. \tag{4}$$

where the $\cos\langle \delta, v_k \rangle$ represents the angle between δ and v_k. For a detailed analysis of Eq. 4 please refer to Appendix B. Intuitively, the objective function reveals that, when the angle between δ and the tangent space or the ℓ_2-norm of δ is large, the $X + \delta$ will run out of the manifold. This explains why we often leverage the ℓ_2-norm to calculate the reconstruction error in the auto-encoder (Wang et al. 2020).

Since each sample has a different basis vector v_k and the attack is conducted under the black-box setting, it is difficult to obtain v_k in our problem. To address the issue, one practical solution is to minimize the $\|\delta\|_2$ during the attack. As such, the generated adversarial example will be close to the manifold with a large probability. However, we find that $\|\delta\|_2$ should be extremely small to evade the detector in practice, e.g., on the UWave dataset, the $\|\delta\|_2$ should be smaller than 0.08. That means, if we manipulate all positions in the sequence, the averaged value of the perturbation at each position will be smaller than $1e^{-4}$. In such a condition, the attack effectiveness will largely drop, e.g., the attack success rate decreases from 48.7% to 18.5% on the UWave dataset after we add ℓ_2 constraints on Eq. 1.

In this work, we propose to minimize the ℓ_0-norm of δ to reduce the ℓ_2-norm indirectly, i.e., we only modify a small part of positions instead of all positions. For instance, if we only manipulate $1/50$ of the positions, the averaged magnitude of the perturbations could raise to 0.005, which could largely increase the attack effectiveness. Since the $\|\delta\|_\infty$ and the $\|\delta\|_2$ are small enough, the detector hardly could recognize the generated adversarial examples in most cases. We thus formulate the attack goal as,

$$\min_{\delta} \; L(\theta; X + \delta, \tilde{y}) + \lambda \|\delta\|_0, \quad \text{s.t. } \|\delta\|_\infty \leq \epsilon. \tag{5}$$

4.2 Independent Approximation

In order to better estimate the black-box gradients for non-convex cases, we should consider the *independent approximation*, which estimates the gradient for each dimension of the input independently (Lax and Terrell 2020). The independent approximation is widely used in non-convex optimization, e.g., the automatic differentiation (Abadi et al. 2016). To demonstrate the effectiveness of the independent approximation, we construct a naive case for better understanding. Suppose a non-convex objective function $r : \mathbb{R}^2 \to \mathbb{R}$ with the definition $f(x) = x_1 \cdot x_2$. Apparently, the gradient of the function is $\nabla_x r(x) = [x_2, x_1]$, while applying existing joint approximation approaches,

$$\nabla_x r(x) \approx \lim_{\eta \to 0} \frac{r(x + \eta \cdot e) - r(x)}{\eta} \cdot e = (x_2 - x_1) \cdot e, \tag{6}$$

where $e = [1, -1]$ is the vector in $\mathbb{R}^2$, which satisfies the requirement such as the unit sphere in AutoZOOM and the Rademacher distribution in SPSA. Although increasing the number of sampled e and η could reduce the estimation bias, the bias is still larger compared with linear cases. On the other side, if we set $e = [1, 0]$ for estimating the first dimension, i.e., the independent approximation, we could obtain the correct estimation of the gradients in this case. Back to our problem, we first approximate the partial derivative as follows, for feature x_{tm} of TSC in X,

$$\frac{\partial L(\theta; X, \tilde{y})}{\partial x_{tm}} \approx \frac{r(X + \eta \cdot \boldsymbol{e}_{tm}) - r(X)}{\eta}, \tag{7}$$

where $\boldsymbol{e}_{tm}$ is an one-hot vector with the only non-zero entry at index (t, m). As such, we could leverage the independent approximation to conduct the optimization in Eq. 5. Nevertheless, there still remains two problems:

- It is extremely difficult to optimize the loss mainly because the ℓ_0 penalty could break the chain rule of the derivation when we estimate the gradients.
- Besides, the independent approximation raises the concern of *limited queries*, i.e., we need to independently estimate the gradients for $T \times M$ times, which requires numerous queries and is unaffordable in black-box attack scenarios (Papernot et al. 2017; Bhagoji et al. 2018).

4.3 Tree Position Search

To address the issue above, we first develop an incremental strategy that could minimize the attack goal with ℓ_0 normalization. That is, at each iteration during the generation of $\tilde{X}$, we perform attacks to K more positions within a sequence. If the attack is not successful, we further increase K positions to attack during next iteration. Then the problem left is how to choose K positions at each iteration to efficiently perform the attack. A straightforward thought is to estimate the gradient at each position with the independent approximation and select those with K-largest gradients. Nevertheless, as we have discussed, it will raise the concern of limited queries. We thus focus on measuring the importance of x_{tm} in a black-box setting without brute-force search.

Towards this end, we develop a new tree position search algorithm, which iteratively narrows down the scope of important positions. As shown in Fig. 2, we first divide the whole sequence into several large regions and select regions with top-K significance scores as candidates. Then we divide each candidate region into smaller regions and calcu-

Algorithm 1: The proposed BlackTreeS.

Input: Classifier f_θ, sequence X and target label $\tilde{y}$.
Output: Adversarial example $\tilde{X}$.

Set $\tilde{X} = X$.
repeat
  Find K most important positions by tree search.
  Calculate $\frac{\partial L(\theta;\tilde{X},\tilde{y})}{\partial x_{tm}}$ for important positions t.
  Form $G(\tilde{X})$ by concatenating $\frac{\partial L(\theta;\tilde{X},\tilde{y})}{\partial x_{tm}}$.
  Update the adversarial example:

$$\tilde{X} = \tilde{X} + \alpha \cdot \text{clip}(G(\tilde{X}), -\epsilon, \epsilon).$$

  Predict the label $\hat{y} = f_\theta(\tilde{X})$.
  if $\hat{y} = \tilde{y}$ or $\|G(X)\|_2 \leq \tau$ or $\|\tilde{X} - X\|_0 > \epsilon_0$ **then**
    Stop the attack.
until Convergence

late the significance scores for these smaller regions. After that, we form the new candidate set by selecting the top-K smaller regions. The search continues until the size of the candidate regions shrinks to one position, i.e., the leaf of the search tree. In this way, we select K positions that contribute most to the target label with queries in $O(KD \log_D T)$, which is much smaller than the brute-force search. To measure the significance of a region, we propose a *regional approximation* procedure, formally,

$$s_{t_1:t_2}(X) = \left| \frac{r(X + \eta \cdot \boldsymbol{e}_{t_1:t_2}) - r(X)}{\eta} \right|, \tag{8}$$

where t_1 and t_2 denote the range of the region; $\boldsymbol{e}_{t_1:t_2}$ is the vector that $\boldsymbol{e}_{tm} = 1$ for $t \in [t_1, t_2]$ and $\boldsymbol{e}_{tm} = 0$ for the others. The significance score reflects the relative importance of a region. For instance, if $s_{0:t}$ is larger than $s_{t:T}$, it indicates that the part within range $[0, t]$ is more important when the classifier predicts the sequence as the target label $\tilde{y}$.

Different from the joint approximation (i.e., Eq. 2), we only manipulate part of the input to approximate gradients of a local region. Although this measurement also suffers from inaccurate estimation, the accuracy could increase along with the top-down search. The main reason lies in Theorem 1. Specifically, a smaller region indicates a smaller D in the theorem, leading smaller estimation error. As such, we would obtain a better approximation. In other words, it is more of a coarse to fine process. We also validate the accuracy of region approximation in Sec. 5.

4.4 The Comprehensive Framework

With the two proposed modules, at the l-th iteration of the adversarial example generation, we first estimate the important positions of $\tilde{X}^{(l)}$. Then we apply the independent approximation to calculate the gradients of important positions and leverage them to update the adversarial example, i.e., $\tilde{X}^{(l+1)}$. Note that if there is no important position, i.e., the $s_{t_1:t_2}(\tilde{X}^{(l)})$ is small for all regions, or the $\|\tilde{X}^{(l)} - X\|_0$ is too large, or the $\tilde{X}^{(l)}$ could successfully achieve the attack goal, we will stop the update and output the adversarial example at current iteration. Algorithm 1 in Appendix C summarizes the overall framework. Compared with existing score-based black-box attacks,

- We propose to only manipulate important positions to generate adversarial examples by the proposed tree position search. Such design helps us to generate adversarial examples that lie on the natural example manifold.
- Instead of approximating the gradients by adding perturbations the whole inputs, we propose to independently estimate the gradients of each important position, which could increase the accuracy of the approximated gradients.

5 Experiment

Experimental Setting. We conduct the experiments on three time series classification datasets: Uwave, Climate and Eye. For the victim models, we implement five representative classifiers including two RNN models (LSTM and Bi-RNN), two CNN models (vanilla CNN and TCN) and one self-attention model (DynamicConv) as the target classifier. We compare the proposed BlackTreeS with six adversarial attack approaches. The first two methods are existing white-box adversarial attacks on TSC: the FGSM (Fawaz et al. 2019) and the PGD (Oregi et al. 2018). The others are the state-of-the-art black-box adversarial attack techniques for other applications: the substitute model, NES, SPSA and AutoZOOM. The effectiveness of the attack is measured by the attack success rate (ASR) (Rathore et al. 2020), i.e., the probability of forcing the target classifier to predict the expected label $\tilde{y}$. For the baseline defense strategies, we implement the auto-encoder based defense strategy proposed (Wang et al. 2020), which is proved to the state-of-the-art defense strategy compared with others such as adversarial training. The stealthiness of the attack is measured by the defense success rate (DSR), i.e., the probability of adversarial examples being detected by the defense strategy.

For all DNN based classifiers, the hidden size and the learning rate are set as 20 and 0.005 respectively. The optimizer of RNN is the RMSProp, while the optimizer of the CNN and self-attention model is the Adam (Diederik, Jimmy et al. 2015). For the BlackTreeS, the K is 20 and the maximal size of perturbed positions is 100. We adopt a quadtree to perform the tree search strategy. For ϵ, the default value is set as 0.3, which is widely used in previous adversarial attacks on TSC models (Oregi et al. 2018). For these parameters, we have also tried various values in the experiment. The popularity size in the NES, SPSA and AutoZOOM is set as 100. All the experiments are conducted on a machine with a 20-core CPU, 256GBs of memory and 5 NVIDIA RTX 2080Ti GPUs. For more details such as the description of the datasets please refer to Appendix C.

5.1 The Effectiveness of the Adversarial Attack

In this subsection, we mainly focus on the effectiveness of our proposed method. We present the main results of the Uwave dataset in Table 1. Firstly, we can see that the quality of the gradients largely influences the attack effectiveness. As we can see from the comparison between NES and

Attack Methods	BiRNN		LSTM		CNN		TCN		DynamicConv		Avg	
	ASR↑	DSR↓	ASR↑	DSR↓	ASR↑	DSR↓	ASR↑	DSR↓	ASR↑	DSR↓	ASR↑	DSR↓
FGSM	15.9%	25.0%	11.4%	37.5%	43.2%	95.5%	23.9%	90.9%	35.2%	94.3%	25.9%	68.6%
PGD	22.7%	20.5%	21.6%	27.3%	58.0%	95.5%	47.7%	72.7%	58.0%	83.0%	41.6%	59.8%
Substitute	2.3%	100.0%	6.8%	100.0%	33.0%	98.9%	5.7%	100.0%	6.8%	100.0%	10.9%	99.8%
NES	1.1%	100.0%	5.7%	100.0%	17.1%	100.0%	4.6%	100.0%	5.7%	100.0%	6.8%	100.0%
SPSA	14.8%	100.0%	19.3%	100.0%	98.9%	100.0%	40.9%	100.0%	51.1%	100.0%	45.0%	100.0%
AutoZOOM	18.2%	90.9%	19.3%	90.9%	**100.0%**	100.0%	56.8%	80.7%	67.1%	96.6%	52.3%	91.8%
BlackTreeS	**26.1%**	**9.1%**	**27.3%**	**6.8%**	**100.0%**	**5.7%**	**60.2%**	**21.6%**	**73.9%**	**28.4%**	**57.5%**	**14.3%**

Table 1: Main results of the UWave dataset.

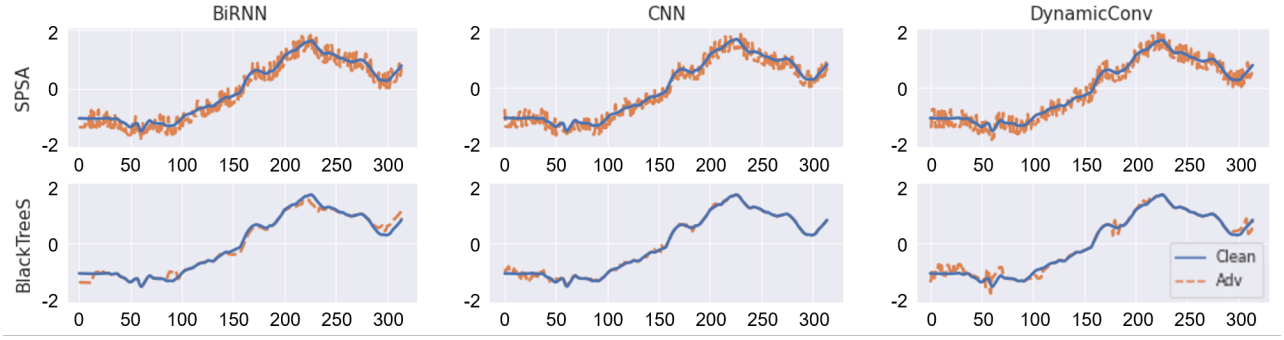

Figure 3: The visualization of SPSA and BlackTreeS attacks on BiRNN, CNN and DynamicConv on the first channel of a sample in the UWave dataset.

FGSM, although they all adopt the sign operation on the gradients of the input, the gradients in the NES are estimated under the black-box setting, which consequently reflects in NES's much lower ASR than FGSM. Therefore, to perform an effective black-box attack on TSC, we should improve the quality of the estimated gradients.

Secondly, as the comparison between existing black-box adversarial attack techniques shows, the substitute model based black-box attack fails to perform an effective attack on various target classifiers. For instance, the ASR is 2.3% for Bi-RNN, and the averaged ASR is only 10.9% for the UWave dataset. As we have discussed in Sec. 3.1, the substitute model is unable to mimic the behavior of the black-box TSC model due to the diversity. On the other side, the score-based methods often show better performance over the substitute model. For instance, the averaged ASR of the AutoZOOM could reach 52.3% on the UWave. Despite the improvement of SPSA and AutoZOOM when the target classifier is CNN, e.g., the ASRs are 98.9% and 100% respectively, their performance become poor when facing with RNN-based models, e.g., the ASRs are 14.8% and 18.2% for Bi-RNN and LSTM respectively. Recall the analysis in Sec. 3.1, the failure mainly comes from the non-linear operations in the classifier, making it difficult to approximate the correct black-box gradients. Compared with existing black-box adversarial attack techniques, our proposed BlackTreeS shows superior performance on all datasets. Furthermore, we find that the ASRs on RNN-based classifiers are greatly improved. For instance, we improve the ASRs of Bi-RNN from 18.2% to 26.1% on UWave, which states that our independent approximation could obtain better gradients when there exist non-convex terms in the classifier.

5.2 The Stealthiness of the Adversarial Attack

We validate the stealthiness of the proposed BlackTreeS by the DSR in Table 1 as well. As the results demonstrates, the BlackTreeS shows the lowest DSR over nearly all models even if it has the highest ASR. For instance, for the Uwave dataset, the averaged DSR is only 14.3%, while the best result of current black-box attack techniques is 91.8%. Such improvement mainly comes from the ℓ_0 penalty during the attack, where we only select important positions to create the adversarial sequences. Moreover, we find that the DSRs of TCN and DynamicConv are often higher than RNN-based approaches for the BlackTreeS. We infer the main reason is that these two approaches pay attention to more positions in the sequence (Zerveas et al. 2021). As a consequence, our approach has to attack more positions to achieve higher ASR. Nevertheless, the DSR is still much lower than existing approaches. To further study the perturbations generated by BlackTreeS, we visualize the adversarial examples generated by our method and the SPSA. From Fig. 3, we find that the BlackTreeS is able to find important positions within a data sequence regarding the classifier. For instance, for the Bi-RNN, our method tries to attack the positions at the head and the tail. For CNN and DynamicConv, our method pays attention to middle positions. In addition, the BlackTreeS also manipulates fewer positions compared with other exist-

Approximation	BiRNN		CNN		DynamicConv	
	ACC	RMSE	ACC	RMSE	ACC	RMSE
Independent	96.1%	0.060	92.2%	0.388	96.4%	0.091
Region	87.8%	-	81.1%	-	84.1%	-
Joint	76.4%	0.138	75.4%	0.891	75.3%	0.414

Table 2: The accuracy of selecting the top 50 most significant gradient positions, and the averaged RMSEs between approximated and ground truth gradients on the UWave dataset.

Attack Methods	UWave		Climate		Eye	
	ASR	DSR	ASR	DSR	ASR	DSR
BlackTreeS	57.5%	14.3%	70.3%	21.0%	93.8%	22.0%
BlackTreeS-RP	27.5%	49.8%	50.3%	55.0%	82.3%	53.5%

Table 3: Comparison of average ASR and DSR between our proposed BlackTreeS with tree position search and with random position selection (suffixed with RP) for attacking.

ing attacks, which helps it better evade potential detection.

5.3 Ablation Study

We further explore the effect of each component in our method. First, to demonstrate the effectiveness of our proposed independent approximation and the region approximation (BlackTreeS), we conduct a case study on the UWave dataset. The results are shown in Table 2. The results show that the estimation error (RMSE) is much lower for independent approximation compared with the joint approximation (SPSA). Besides, the region approximation is also relatively accurate in selecting top-K positions, e.g., the ACC is over 80% for all three models. Second, to evaluate the effectiveness of our proposed tree position search algorithm, we substitute the original position search module with random position selection and compare their performance difference on all three datasets. We keep the BlackTreeS with random position selection attack the same number of positions as the original one, and they share the same gradient approximation procedure. The average ASR and DSR across all models are shown in Table 3, where BlackTreeS-RP denotes the random position selection variant. The results demonstrate that our proposed BlackTreeS consistently achieves higher ASRs and lower DSRs than BlackTreeS-RP among all three datasets, which indicates the effectiveness of the tree position search. For more results such as the influence of hyper-parameters and query count please refer to Appendix C.

6 Related Work

Time series classification (TSC) is a crucial task in modern data mining, which aims to classify sequential data into different categories (Yang and Wu 2006; Esling and Agon 2012; Gupta et al. 2020). Specifically, multivariate time series classification has wide applications such as stock trend prediction (Ding et al. 2019), network flow recognition (Hayes and Danezis 2016) and medical data analysis (Che et al. 2017). Recently, deep neural network (DNN) shows superior performance on this task (Gamboa 2017; Wang, Yan, and Oates 2017). For instance, CNN is proposed to capture the local temporal pattern by the convolution operation (Cui, Chen, and Chen 2016), RNN is proposed to model the temporal dependency (Smirnov and Nguifo 2018), and self-attention model is proposed to find similar positions in the input (Zerveas et al. 2021). More details can be found in the survey (Gupta et al. 2020). Despite their effectiveness, recent studies have found that DNNs are vulnerable to adversarial attacks. Several attacks and defenses have been proposed accordingly. For instance, (Fawaz et al. 2019) and (Oregi et al. 2018) propose to leverage the FGSM and PGD respectively to create adversarial examples for TSC models (Fawaz et al. 2019; Oregi et al. 2018), while (Wang et al. 2020) and (Belkhouja and Doppa 2020) propose to detect adversarial examples by the auto-encoder model and the adversarial training framework respectively (Wang et al. 2020; Belkhouja and Doppa 2020). In a word, current adversarial attacks on TSC models strongly rely on accurate gradients ensured by the white-box attack setting, which makes them less practical for real-world scenarios. Besides, the adversarial examples generated by existing attack methods could often be easily detected by the defense strategy.

Under the black-box setting, attackers can only obtain the input and output of the model instead of the whole model (Ilyas et al. 2018; Narodytska and Kasiviswanathan 2017; Guo et al. 2019). Existing work on black-box adversarial attacks mainly leverages two kinds of approaches: substitute model and score-based approaches. The first split of approaches aims to mimic the target model by several queries and transfer the generated adversarial examples to the target model (Ilyas et al. 2018). The second split of approaches aims to estimate the gradients of inputs with numerical approximations such as NES (Wierstra et al. 2014; Ilyas et al. 2017), SPSA (Spall et al. 1992) and AutoZoom (Chen et al. 2017). Since these approaches do not require the details of the target model, they could be applied in various real-world applications. To further study the vulnerability of TSC models in real-world applications, we need to investigate a stealthy black-box adversarial attack for this task.

7 Conclusion

In this work, we are the first to reveal the threat of effective and stealthy black-box adversarial attacks on DNN based time series classification. We highlight the difference between TSC and other applications during the black-box adversarial attack. To deal with the challenges of the low-dimensional manifolds and non-convex classifiers, we propose a novel framework called BlackTreeS. Our study sheds light on the threat of adversarial attacks when we apply the DNN based TSC models in real-world scenarios. In the future, we consider to extend our work to more kinds of sequential data such as discrete value based sequences and sentences. Second, we may further study the failure of joint approximation with more theoretical analysis. Lastly, we tend to leverage the BlackTreeS to discover potential threats of adversarial attacks in current commercial DNN based TSC services.

A Black-box Non-Convex Optimization

Convex optimization is widely used for analyzing the convergence of the learning (Diederik, Jimmy et al. 2015), the main assumptions such as the L-strongly convex could be satisfied when we analyze the convolutional neural network (Diederik, Jimmy et al. 2015). The main reason lies in that the convolutional operation and ReLU activation function often act in a linear pattern (Virmaux and Scaman 2018). However, as we have discussed, owing to the numerous non-linear operations and product terms in RNN and self-attention models, they often show strong non-convexity. Therefore, in this work, we pay attention to the analysis based on non-convex optimization, specifically,

Theorem. *3.1. Suppose a non-convex and Lipschitz continuous function $r(x) : \mathbb{R}^D \to \mathbb{R}$ is optimized with the gradients in Eq. 3. Also suppose the maximal norm of the gradients is $\|\nabla_x r\|_*$. Then we have,*

$$r(x^{(I)}) - r(x^{(0)}) \leq \frac{\sqrt{6\alpha}}{8} \sum_{l=0}^{I-1} \left(\frac{D+4}{2} \sqrt{\frac{3\alpha}{2}} \|\nabla_x r\|_* - \|\nabla r(x^{(l)})\| \right)$$

where $x^{(l)}$ is the variable at l^{th} iteration in the SGD, I is the step of the iteration and α is the learning rate.

Proof. We first define the Gaussian approximation of a function, formally,

$$f_\alpha(X) = \frac{1}{\kappa} \int f(x + \alpha u) e^{-\frac{1}{2}\|u\|^2} du \tag{9}$$

where α is the approximated coefficient, i.e., the learning rate used in the optimization. Then we suppose that,

$$\|\nabla r(x_1) - \nabla r(x_2)\| \leq \|\nabla_x r\|_* \|x_1 - x_2\| \tag{10}$$

for any x_1 and x_2. As pointed out by previous researches (Nesterov and Spokoiny 2017), the following condition holds if we adopt the black-box optimization such as the one used in SPSA,

$$\mathbb{E}_\eta[r_\alpha(x^{(I)})] \leq f_\alpha - \frac{\|\nabla f_\alpha(x^{(l)})\|^2}{8\tilde{D}\|\nabla_x r\|_*} + \frac{3\alpha^2 \|\nabla_x r\|_* \tilde{D}}{32}, \tag{11}$$

where $\tilde{D} = D + 4$. By summing up through all l and taking expectation on η,

$$\begin{aligned}
&\mathbb{E}_\eta[r_\eta(x^{(I+1)}) - r_\eta(x^{(0)})] \\
\leq &\frac{3\alpha^2 I \|\nabla_x r\|_*^2 (D+4)^2 - 8\sum_{l=0}^{I-1} \|\nabla r(x^{(l)})\|^2}{32(D+4)\|\nabla_x r\|_*} \\
\leq &\frac{\sum_{l=0}^{I-1} \left(\frac{D+4}{2} \sqrt{\frac{3\alpha}{2}} \|\nabla_x r\|_* + \|\nabla r(x^{(l)})\| \right)}{4(D+4)\|\nabla_x r\|_*} \\
&\cdot \left(\frac{D+4}{2} \sqrt{\frac{3\alpha}{2}} \|\nabla_x r\|_* - \|\nabla r(x^{(l)})\| \right) \\
\leq &\frac{1}{4(D+4)\|\nabla_x r\|_*} \sum_{l=0}^{I-1} \sqrt{\frac{3\alpha}{2}} (D+4) \|\nabla_x r\|_* \\
&\cdot \left(\frac{D+4}{2} \sqrt{\frac{3\alpha}{2}} \|\nabla_x r\|_* - \|\nabla r(x^{(l)})\| \right) \\
\leq &\frac{\sqrt{6\alpha}}{8} \sum_{l=0}^{I-1} \left(\frac{D+4}{2} \sqrt{\frac{3\alpha}{2}} \|\nabla_x r\|_* - \|\nabla r(x^{(l)})\| \right)
\end{aligned} \tag{12}$$

□

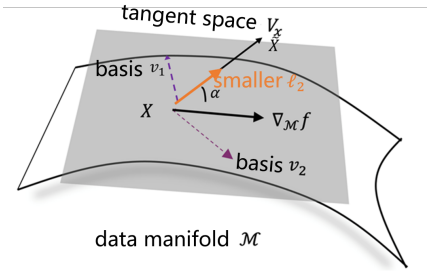

Figure 4: The tangent space and tangent vector around X.

Considering that $\frac{D+4}{2}\sqrt{\frac{3\alpha}{2}} > 1$ for most cases and $\|\nabla_x r\|_* > \|\nabla r(x^{(l)})\|$, the major factor that determines the upper bound are two-folds: (1) the dimension D and (2) the maximal norm of the gradient.

B The Manifold of Natural Examples

We present more details for the statement in Sec. 3.2. As we have discussed, previous work (Khoury and Hadfield-Menell 2018) proves the following theorem,

Lemma 1. *Let $\mathcal{M} \subset \mathbb{R}^D$ be a d-dimensional manifold embedded in $\mathbb{R}^D$ with finite volume. Let $X \subset \mathcal{M}$ be a finite set of points sampled from $\mathcal{M}$. Suppose that ϵ is smaller than the reach of $\mathcal{M}$'s medial axis defined in (Dey 2006), we have*

$$\frac{vol(X^\epsilon \cap \mathcal{M}^\epsilon)}{vol\mathcal{M}^\epsilon} \leq \frac{\pi^{\frac{d}{2}} \Gamma(\frac{D-d}{2}+1)}{\Gamma(\frac{D}{2}+1)} \cdot \frac{\epsilon^d}{vol\mathcal{M}}, \tag{13}$$

where vol is the volume of the set and

$$\begin{aligned}
\mathcal{M}^\epsilon &= \{x \in \mathbb{R}^D : \inf_{z \in \mathcal{M}} \|x - z\|_2 \leq \epsilon\} \\
X^\epsilon &= \{x \in \mathbb{R}^D : \inf_{z \in \mathcal{X}} \|x - z\|_2 \leq \epsilon\}.
\end{aligned} \tag{14}$$

Owing to the approximation,

$$\frac{\Gamma(\frac{D-d}{2}+1)}{\Gamma(\frac{D}{2}+1)} \approx \left(\frac{2}{D-d}\right)^{\frac{d}{2}}. \tag{15}$$

This theorem states that, if we add perturbations on the training point to construct X^ϵ, the ratio between the volume of the $X^\epsilon \cap \mathcal{M}^\epsilon$ and the volume of the manifold $\mathcal{M}^\epsilon$ is proportional to $\left(\frac{2}{D-d}\right)^{\frac{d}{2}}$. Since $\left(\frac{2}{D-d}\right)^{\frac{d}{2}}$ is a increasing function, $X^\epsilon \cap \mathcal{M}^\epsilon$ would be much smaller than $\mathcal{M}^\epsilon$ if d were small, which means that the adversarial examples would easily be far from the natural example manifold even if the ℓ_2 norm is smaller than ϵ.

C The Tangent Vector of Manifold

We further present the claim of ℓ_0 normalization in Sec. 4.2 as follows. Let $\mathcal{M} \subset \mathbb{R}^D$ be a d-dimensional manifold embedded in $\mathbb{R}^D$ with finite volume, and a smooth real-valued function defined on the manifold $f : \mathcal{M} \to \mathbb{R}$, e.g., the loss function of the targeted attack. For a variable $X \in \mathcal{M}$, the $\nabla_{\mathcal{M}} f(X)$ is the gradient along the manifold at point X,

which is different from the gradient in the Euclidean space $\nabla_{\mathbb{R}^D} f(X)$.

To generate adversarial examples $\tilde{X}$ that lie on the manifold, we need to optimize the $\tilde{X}$ by the gradients $\nabla_{\mathcal{M}} f(X)$, in other words, the $\tilde{X} - X$ should be similar to $\nabla_{\mathcal{M}} f(X)$. However, the main challenges are: (1) the manifold $\mathcal{M}$ is extremely difficult to be described in a closed form (Dey, Ranjan, and Wang 2010), and (2) the function f is unknown to the attacker under the black-box setting. To address the issues, we introduce the concept of *tangent space*. Formally, suppose $X \in \mathcal{M}$ and a linear mapping V_X, i.e., a vector in $\mathbb{R}^D$, satisfy the condition $V_p(f \cdot g) = f(X) \cdot [V_p g] + g(X) \cdot [V_p f]$ for any smooth function $f, g : \mathcal{M} \to \mathbb{R}$. The set of all $V_X \in \mathbb{R}^D$, i.e., the tangent vectors, that satisfy the condition is called the tangent space at X. Intuitively, the tangent vector indicates the direction along the manifold at point X, e.g., the gradient $\nabla_{\mathcal{M}} f(X)$. Furthermore, according to the rank theorem, since the dimension of $\mathcal{M}$ is d, then the basis of tangent space could be represented by $v_k \in \mathbb{R}^D$ for $k = 1, \cdots, d$. That is, any tangent vector is a linear combination of the v_k. As such, in order to force $\tilde{X} - X$ to be close to the tangent space, we could minimize the following objective function $\min_\delta \sum_{k=1}^{d} \|\delta\|_2 \cdot \cos\langle \vec{\delta}, v_k \rangle$ (Li et al. 2020), where $\|\delta\|_\infty \leq \epsilon$.

On one side, the objective function describes the angle between the vector δ and the tangent space, where the vector is in the tangent space if the angle is 0. On the other side, when the angle is not 0, the objective function describes the length of the projection of δ on the tangent space, as shown in Fig. 4. As such, larger values represent that the vector will be far away from the manifold. Several approaches are proposed to estimate the basis vectors v_k, e.g., dimensional reduction methods (Papaioannou et al. 2021) or functional approximation (Qi et al. 2018), which require the full dataset to characterize the data manifold. However, in the black-box attack scenarios, it is difficult to obtain the training data and the victim model, making it difficult to estimate the basis vectors for each target sample.

In this work, we present a simple way that could conduct the optimization, i.e., minimizing the ℓ_2 penalty term, which could minimize the projected length between the vector δ and the tangent space. On the other side, the Eq. 10 also explains why current defense strategies leverage the ℓ_2 distance between the reconstructed input and the original input to detect adversarial examples (Wang et al. 2020). However, we find that the attack effectiveness is limited if we restrict the ℓ_2 to be small values. We infer the main reason is that the averaged perturbations will be extremely small in this case. To address the issue, we propose to minimize the ℓ_0 norm, as such, the perturbations on each position will be sufficient to cause the wrong prediction. Theoretically, the model could learn such property if we only leverage the ℓ_2 penalty, i.e., only attacking important positions to improve the attack effectiveness, however, due to the inaccurate gradient estimation under the black-box setting and the local-optima caused by the optimization method, it is difficult to automatically find such attack strategy. In a word, the ℓ_0 penalty will help the model achieve better effectiveness while the stealthiness is still held in our problem.

Dataset	T	D	Category	Train Set	Test Set
UWave	315	3	8	352	88
Climate	200	1	3	320	80
Eye	200	14	2	320	80

Table 4: Dataset description.

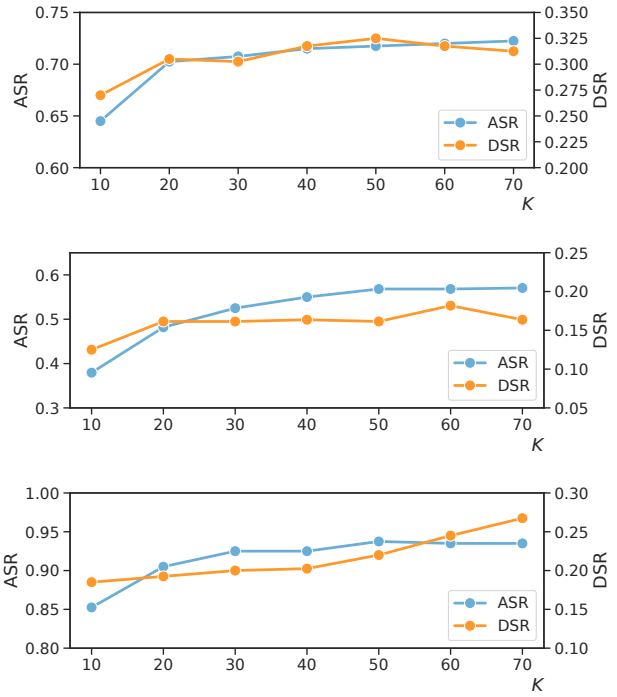

Figure 5: The average ASR and DSR across all five classification models of BlackTreeS under different K on Climate (top), UWave (middle) and Eye (bottom).

D More Empirical Results

We study the influence of hyper-parameters in our method, the number of important positions K. We tried different settings on the three datasets and present the results in Figure 5. As we can see from the results, our method is not sensitive to the choice of K. For instance, on the Eye dataset, the DSR is still below 27% when $K = 70$, and the ASR is over 65% even if $K = 10$. Owing to the dynamic tree position search, when the number of positions is not enough to perform successful attacks on the classifier, our model would turn to the sub-important positions in the next tree position search. Therefore, smaller K does not influence the ASR a lot when $K > 10$. On the other side, our method would stop when there do not exist important positions over the threshold τ, which reflects that a larger K does not cause a larger DSR in our framework. Therefore our attack strategy is not sensitive to the choice of hyper-parameters in most cases.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments that helped improve the quality of the paper. This work was supported in part by the National Key Research and Development Program (2021YFB3101200), National Natural Science Foundation of China (61972099, U1736208, U1836210, U1836213, 62172104, 62172105, 61902374, 62102093, 62102091, 62272437), Natural Science Foundation of Shanghai (19ZR1404800). Min Yang is a faculty of Shanghai Institute of Intelligent Electronics & Systems, Shanghai Collaborative Innovation Center of Intelligent Visual Computing and Engineering Research Center of Cyber Security Auditing and Monitoring, Ministry of Education, China. Mi Zhang and Min Yang are the corresponding authors.

References

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Belkhouja, T.; and Doppa, J. R. 2020. Analyzing Deep Learning for Time-Series Data Through Adversarial Lens in Mobile and IoT Applications. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.

Bhagoji, A. N.; He, W.; Li, B.; and Song, D. 2018. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 154–169.

Cartella, F.; Anunciacao, O.; Funabiki, Y.; Yamaguchi, D.; Akishita, T.; and Elshocht, O. 2021. Adversarial attacks for tabular data: Application to fraud detection and imbalanced data. *arXiv preprint arXiv:2101.08030*.

Che, Z.; Cheng, Y.; Zhai, S.; Sun, Z.; and Liu, Y. 2017. Boosting deep learning risk prediction with generative adversarial networks for electronic health records. In *2017 IEEE International Conference on Data Mining (ICDM)*, 787–792. IEEE.

Chen, P.-Y.; Zhang, H.; Sharma, Y.; Yi, J.; and Hsieh, C.-J. 2017. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 15–26.

Cui, Z.; Chen, W.; and Chen, Y. 2016. Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995*.

Dey, T. K. 2006. *Curve and surface reconstruction: algorithms with mathematical analysis*, volume 23. Cambridge University Press.

Dey, T. K.; Ranjan, P.; and Wang, Y. 2010. Convergence, stability, and discrete approximation of Laplace spectra. SIAM.

Diederik, K.; Jimmy, B.; et al. 2015. Adam: A method for stochastic optimization. *ICLR*.

Ding, D.; Zhang, M.; Pan, X.; Yang, M.; and He, X. 2019. Modeling extreme events in time series prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1114–1122.

Dong, Y.; Su, H.; Wu, B.; Li, Z.; Liu, W.; Zhang, T.; and Zhu, J. 2019. Efficient decision-based black-box adversarial attacks on face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7714–7722.

Erichson, N. B.; Azencot, O.; Queiruga, A.; Hodgkinson, L.; and Mahoney, M. W. 2020. Lipschitz Recurrent Neural Networks. In *International Conference on Learning Representations*.

Esling, P.; and Agon, C. 2012. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1): 1–34.

Fawaz, H. I.; Forestier, G.; Weber, J.; Idoumghar, L.; and Muller, P.-A. 2019. Adversarial attacks on deep neural networks for time series classification. In *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Gamboa, J. C. B. 2017. Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Guo, C.; Gardner, J.; You, Y.; Wilson, A. G.; and Weinberger, K. 2019. Simple black-box adversarial attacks. In *International Conference on Machine Learning*, 2484–2493. PMLR.

Gupta, A.; Gupta, H. P.; Biswas, B.; and Dutta, T. 2020. Approaches and applications of early classification of time series: A review. *IEEE Transactions on Artificial Intelligence*, 1(1): 47–61.

Hayes, J.; and Danezis, G. 2016. Website fingerprinting at scale. *NDSS*.

Ilyas, A.; Engstrom, L.; Athalye, A.; and Lin, J. 2017. Query-Efficient Black-box Adversarial Examples. *ArXiv*, abs/1712.07113.

Ilyas, A.; Engstrom, L.; Athalye, A.; and Lin, J. 2018. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, 2137–2146. PMLR.

Karim, F.; Majumdar, S.; and Darabi, H. 2020. Adversarial attacks on time series. *IEEE transactions on pattern analysis and machine intelligence*, 43(10): 3309–3320.

Khoury, M.; and Hadfield-Menell, D. 2018. On the geometry of adversarial examples. *arXiv preprint arXiv:1811.00525*.

Lax, P. D.; and Terrell, M. S. 2020. *Calculus with applications*. Springer.

Li, Y.; Cheng, S.; Su, H.; and Zhu, J. 2020. Defense against adversarial attacks via controlling gradient leaking on embedded manifolds. In *16th European Conference on Computer Vision*, volume 12373, 753–769. Springer.

Liu, Y.; Chen, X.; Liu, C.; and Song, D. 2016. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Meng, L.; Lin, C.-T.; Jung, T.-P.; and Wu, D. 2019. White-box target attack for EEG-based BCI regression problems. In *International conference on neural information processing*, 476–488. Springer.

Narodytska, N.; and Kasiviswanathan, S. P. 2017. Simple Black-Box Adversarial Attacks on Deep Neural Networks. In *CVPR Workshops*, volume 2, 2.

Nesterov, Y.; and Spokoiny, V. 2017. Random gradient-free minimization of convex functions. *FoCM*.

Nguyen, T. Q.; and Salazar, J. 2019. Transformers without tears: Improving the normalization of self-attention. *arXiv preprint arXiv:1910.05895*.

Oregi, I.; Ser, J. D.; Perez, A.; and Lozano, J. A. 2018. Adversarial sample crafting for time series classification with elastic similarity measures. In *International Symposium on Intelligent and Distributed Computing*, 26–39. Springer.

Papaioannou, P.; Talmon, R.; di Serafino, D.; Kevrekidis, I.; and Siettos, C. 2021. Time Series Forecasting Using Manifold Learning. *arXiv preprint arXiv:2110.03625*.

Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z. B.; and Swami, A. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 506–519.

Pascanu, R.; Mikolov, T.; and Bengio, Y. 2013. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, 1310–1318. PMLR.

Pless, R.; and Souvenir, R. 2009. A survey of manifold learning for images. *IPSJ Transactions on Computer Vision and Applications*, 1: 83–94.

Qi, G.-J.; Zhang, L.; Hu, H.; Edraki, M.; Wang, J.; and Hua, X.-S. 2018. Global versus localized generative adversarial nets. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, 1517–1525. IEEE Computer Society.

Rathore, P.; Basak, A.; Nistala, S. H.; and Runkana, V. 2020. Untargeted, targeted and universal adversarial attacks and defenses on time series. In *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Rodrigues, P. L. C.; Congedo, M.; and Jutten, C. 2018. Multivariate time-series analysis via manifold learning. In *2018 IEEE Statistical Signal Processing Workshop (SSP)*, 573–577. IEEE.

Smirnov, D.; and Nguifo, E. M. 2018. Time series classification with recurrent neural networks. *AALTD*.

Spall, J. C.; et al. 1992. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3): 332–341.

Tu, C.-C.; Ting, P.; Chen, P.-Y.; Liu, S.; Zhang, H.; Yi, J.; Hsieh, C.-J.; and Cheng, S.-M. 2019. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 742–749.

Uesato, J.; O'donoghue, B.; Kohli, P.; and Oord, A. 2018. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, 5025–5034. PMLR.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Virmaux, A.; and Scaman, K. 2018. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31.

Wang, W.; Tang, P.; Xiong, L.; and Jiang, X. 2020. Radar: Recurrent autoencoder based detector for adversarial examples on temporal ehr. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 105–121. Springer.

Wang, Z.; Yan, W.; and Oates, T. 2017. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, 1578–1585. IEEE.

Wei, X.; Yan, H.; and Li, B. 2022. Sparse black-box video attack with reinforcement learning. *International Journal of Computer Vision*, 130(6): 1459–1473.

Wei, Z.; Chen, J.; Wei, X.; Jiang, L.; Chua, T.-S.; Zhou, F.; and Jiang, Y.-G. 2020. Heuristic black-box adversarial attacks on video recognition models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12338–12345.

Wierstra, D.; Schaul, T.; Glasmachers, T.; Sun, Y.; Peters, J.; and Schmidhuber, J. 2014. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1): 949–980.

Yang, Q.; and Wu, X. 2006. 10 CHALLENGING PROBLEMS IN DATA MINING RESEARCH. *IJITDM*.

Zerveas, G.; Jayaraman, S.; Patel, D.; Bhamidipaty, A.; and Eickhoff, C. 2021. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2114–2124.

Zhan, X.; Li, Y.; Li, R.; Gu, X.; Habimana, O.; and Wang, H. 2018. Stock price prediction using time convolution long short-term memory network. In *International Conference on Knowledge Science, Engineering and Management*, 461–468. Springer.