Adversarial Attacks for Sequential Data Models

Alina Bogdanova [1] **Nikita Ligostaev** [1] **Matvey Skripkin** [1] **Anastasia Sozykina** [1]

Abstract

This paper presents a novel approach to adversarial attacks in the realm of time series models. Adversarial attacks, which have been widely studied in computer vision, are adapted for time series data. However, a significant challenge in this domain is the susceptibility of these attacks to detection by discriminators.

To address this challenge, we introduce an innovative adversarial attack method that incorporates regularization with a discriminator model. This regularization enhances the concealment of the attacks, making them less prone to detection.

Our results underscore the potential of adversarial attacks to compromise time series models while remaining undetected, underscoring the critical need for robust defense mechanisms in this domain. This research contributes to a better understanding of security concerns in machine learning applications.

Github repository: link to the project Github repository.

1. Introduction

Deep learning models have demonstrated remarkable performance in various domains, but they are not immune to vulnerabilities (Szegedy et al., 2014). Adversarial attacks, which were originally prominent in the realm of computer vision (Goodfellow et al., 2014), have attracted substantial interest for their capability to undermine the reliability of machine learning models. In contrast to the extensive research on adversarial attacks targeting image-domain models, there has been relatively limited attention given to adversarial attacks on time series models. However, it is crucial to recognize that these attacks have the potential to be highly impactful in sensitive applications such as medical diagnosis, industrial chemical processes, and more. In medical applications, for example, adversarial attacks on time series models could lead to misdiagnoses or inaccurate predictions, potentially endangering patient outcomes. Similarly, in industrial chemical analysis, adversarial perturbations on time series data could result in hazardous substances being undetected or incorrect analyses, thereby compromising the safety of the environment or human health.

This paper extends the scope of adversarial attacks into the domain of sequential data models, particularly emphasizing their application to time series data (Fawaz et al., 2019). In the context of time series data, a critical challenge surfaces: the susceptibility of these attacks to detection by specialized classifiers known as discriminators. Addressing this challenge, we introduce a novel adversarial attack approach that integrates a discriminator model's predictions, thereby enhancing concealment.

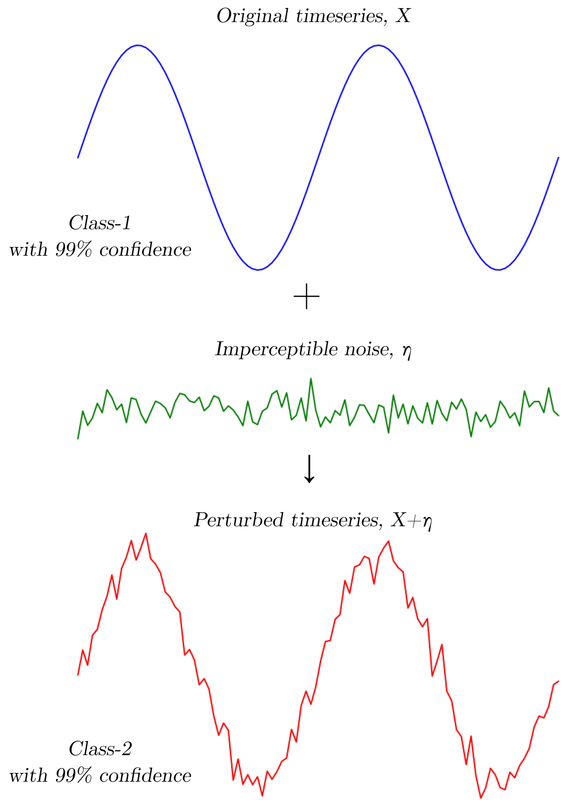


Figure 1. An example of adversarial perturbations. The first row shows the original time series X, classified as Class-1. The second row displays the corresponding perturbation η, which can, for example, be computed using the Iterative Fast Gradient Sign Method (IFGSM). The third row depicts the time series $X + \eta$, classified as Class-2. This figure is inspired by (Fawaz et al., 2019).

Our project revolves around the training of two distinct time series classification models, one of which is based on the TimesNet architecture. We advocate for the implementation of these attacked models using PyTorch, which facilitates gradient-based adversarial perturbations. These adversarial perturbations are generated through the Iterative Fast Gradient Sign Method (IFGSM) and DeepFool techniques.

Our experimental framework employs FordA dataset, which are suitable for binary time series classification, alongside other pertinent time series datasets. The results of our study underscore the potential of adversarial attacks in compromising the performance of time series models while evading detection. This highlights the imperative need for robust defense mechanisms in the realm of sequential data models. This research contributes to an enhanced understanding of security concerns in the practical application of machine learning.

Our main contributions are:

- Training TimesNet and LSTM-FCN for time-series classification;
- Implementation of the attacked model using PyTorch, making sure it's vulnerable to attacks;
- Performing IFGSM and DeepFool adversarial attacks on the model;
- Training discriminator models to evaluate the effectiveness and concealability of the attacks;
- Evaluating the attacks using metrics:
  Effectiveness: 1 - accuracy of the attacked model,
  Concealability: 1 - accuracy of the discriminator model.

2. Related work

Adversarial attacks have their roots in the field of computer vision. Broadly, these attacks aim to introduce minimal perturbations into input data to achieve a specific misclassification objective (Madry et al., 2017). In recent years, there has been a multitude of proposed approaches for computing adversarial perturbations. These approaches can be categorized as either white-box or black-box attacks. White-box attacks utilize information about a model's gradients, and some well-known examples include the Fast Gradient Sign Method (FGSM, (Goodfellow et al., 2015)), Projected Gradient Descent (PGD, (Madry et al., 2017)), Carlini and Wagner attacks (Carlini & Wagner, 2017), and DeepFool (Moosavi-Dezfooli et al., 2016). On the other hand, when access to the model's gradients is unavailable, adversarial perturbations can still be computed using a query-based approach (Ilyas et al., 2018; Andriushchenko et al., 2020; Yin et al., 2023; Zhang et al., 2022). Such a variety of approaches to fool neural networks have been explored in various areas of machine learning, including biomedical imaging (Shao et al., 2021), face detection and recognition (Dong et al., 2019), and speech recognition (Qin et al., 2019), raising concerns about the reliability of models deployed in these fields.

Deep learning based time series classification models have been shown to be vulnerable to various types of adversarial attacks, as demonstrated in a study on UCR time series datasets (Rathore et al., 2020). The transferability of these attacks and the generalization of defenses across different time series datasets have been subjects of interest. For example, (Karim et al., 2020) conducted research on the transferability of adversarial attacks on time series data, highlighting the importance of robust regularization methods. To improve the robustness of time series models, incorporating regularization strategies has become essential. One approach is the application of adversarial training with regularization introduced by (Zhang et al., 2023), which has shown potential in enhancing model security. Additionally, the concept of randomized smoothing has been extended to time series data by (Raghunathan et al., 2020), emphasizing the effectiveness of noise injection as a regularization technique to strengthen model robustness.

3. Methodology

In the context of adversarial attacks on time series data, a common challenge is that these attacks can often be detected by a specialized model called a ”discriminator”. To address this issue, we propose a novel approach, which involves a special type of adversarial attack method that incorporates regularization based on the predictions of the discriminator model. This approach is designed to make it more difficult for the discriminator to detect the adversarial perturbations. As a result, the adversarial attacks become less conspicuous and harder to detect by the discriminator. This strategy aims to enhance the stealthiness and effectiveness of adversarial attacks in the time series domain.

Consider a time series dataset consisting of a collection of input samples denoted as $X = \{x_1, x_2, \ldots, x_N\}$, where x_i represents an individual time series sample. Each time series x_i is associated with a ground truth label y_i, and there exists a discriminator model D trained to distinguish between genuine and adversarial time series samples. The goal is to generate adversarial time series samples, denoted as $X_{\text{adv}} = \{x_{\text{adv}_1}, x_{\text{adv}_2}, \ldots, x_{\text{adv}_N}\}$, such that they are misclassified by a target classifier model F, while minimizing detection by the discriminator D.

The problem can be defined as follows:

Given:
$X = \{x_1, x_2, \ldots, x_N\}$: Original time series dataset with ground truth labels;
F: The target classifier model;
D: The discriminator model;
ϵ: A perturbation budget (maximum allowable perturbation).

Objective:
Find $X_{\text{adv}} = \{x_{\text{adv}_1}, x_{\text{adv}_2}, \ldots, x_{\text{adv}_N}\}$ such that:

$$\begin{aligned} &\text{Minimize detection by discriminator } D; \\ &\text{Maximize misclassification by target classifier } F; \\ &\text{Subject to} \quad \|x_{\text{adv}_i} - x_i\|_\infty \leq \epsilon, \quad \forall i \in [1, N], \end{aligned}$$

where $\|x_{\text{adv}_i} - x_i\|_\infty$ represents the maximum perturbation allowed for each time series sample.

Solution:
Find a perturbation $x_{\text{adv}_i} - x_i$ for each time series sample x_i that satisfies the constraints while achieving the desired misclassification by F and minimizing detection by D.

The problem aims to enhance the stealthiness and effectiveness of adversarial attacks on time series data by finding perturbations that lead to misclassification while being difficult to detect by the discriminator. Solving this problem involves optimizing the perturbations X_{adv} such that they achieve the desired trade-off between misclassification and stealthiness within the perturbation budget ϵ.

3.1. Time series classifiers

For conducting experiments we considered two models intended for time series classification, namely TimesNet (Wu et al., 2022) and Long Short Term Memory Fully Convolutional Network (LSTM-FCN) (Karim et al., 2019).

3.1.1. TimesNet

The authors of the article (Wu et al., 2022) introduce an innovative approach to the analysis of time series data by addressing the complex temporal variations present in real-world datasets. The core challenge lies in the existence of multiple periodicities, where various time scales overlap and interact, making the modeling of these variations intricate. The authors identify two distinct types of temporal variations: intraperiod-variation, which characterizes short-term patterns within a period, and interperiod-variation, which reflects long-term trends across different periods. To overcome the limitations of representing these variations in a 1D time series, the authors propose a transformation into a 2D space. By reshaping the 1D time series into a 2D tensor, where each column represents time points within a period, and each row corresponds to time points at the same phase across different periods, the authors effectively unify intraperiod- and interperiod-variations. This transformation provides a clearer representation of complex temporal patterns and enables a modular architecture for temporal variation modeling. The proposed model, TimesNet, incorporates TimesBlock to discover multiple periodicities and captures temporal 2D-variations using a parameter-efficient inception block. The TimesNet model demonstrates its versatility by achieving state-of-the-art performance in various time series analysis tasks, including forecasting, imputation, classification, and anomaly detection.

3.1.2. LSTM-FCN for Time Series Classification

In the paper (Karim et al., 2019), the authors propose transforming existing univariate time series classification models, specifically the the Long Short Term Memory Fully Convolutional Network (LSTM-FCN) and Attention LSTM-FCN (ALSTM-FCN), into multivariate time series classification models. They achieve this by augmenting the fully convolutional block with a squeeze-and-excitation block, resulting in improved accuracy. This enhancement helps to capture important temporal dependencies and patterns in multivariate time series data. The proposed models outperform most state-of-the-art models and are efficient for complex multivariate time series tasks such as activity or action recognition. Additionally, these models are highly efficient during testing and can be deployed on memory-constrained systems.

3.2. Adversarial attacks

In our research two prominent techniques to evaluate the robustness of time series model will be used: the Iterative Fast Gradient Sign Method (IFGSM) and DeepFool.

3.2.1. IFGSM for binary classifier

The Fast Gradient Sign Method (FGSM) is a technique in adversarial machine learning. It creates adversarial examples by calculating the gradient of the loss with respect to the input data, focusing on the gradient's direction, and adding a small perturbation to the input. The goal is to deceive the model with minimal changes to the input while highlighting the model's vulnerability to adversarial attacks. Iterative Fast Gradient Sign Method (IFGSM) is an iterative version of FGSM. It applies FGSM multiple times in succession, each time recalculating the gradient and taking a step. These multiple steps (iterations) gradually refine the perturbation. IFGSM can be configured to run for a specific number of iterations, making it more flexible. This iterative approach can create more robust and stronger perturbations compared to a single-step FGSM attack. It can find more complex perturbations that are harder for models to defend against, but it is computationally more expensive. Pseudo-code of

IFGSM attack presented in Algorithm 1.

Algorithm 1 Iterative Fast Gradient Sign Method (IFGSM) for Binary Classifier (Kurakin et al., 2018)

```
Require: Time series $X$, classifier $f$, maximum iterations $T$, step size $\alpha$.
Ensure: Perturbation $\hat{r}$.
Initialize $X_0 \leftarrow X$, $i \leftarrow 0$.
while $i < T$ do
    Compute gradient: $\nabla J(X_i, Y)$ where $J$ is the loss function and $Y$ is the target class.
    Perturb time series: $X_{i+1} \leftarrow X_i + \alpha \cdot \text{sign}(\nabla J(X_i, Y))$.
    Clip perturbation: $X_{i+1} \leftarrow \text{clip}(X_{i+1}, X - \epsilon, X + \epsilon)$, where $\epsilon$ is a bound on the perturbation.
    $i \leftarrow i + 1$
end while
return: $\hat{r} = X_T - X$.
```

3.2.2. DeepFool for binary classifier

The general idea of the DeepFool adversarial attack for binary classification (Moosavi-Dezfooli et al., 2016) is to iteratively compute the smallest perturbation to an input that causes it to be misclassified from one class to the other. To compute the perturbation, DeepFool locally approximates the decision boundary near the current input. It assumes that in the small neighborhood of the input, the model's decision boundary is approximately linear. The attack then calculates the minimal perturbation needed to cross this linear decision boundary and move the input closer to the desired target class. This perturbation is determined by analyzing the gradients of the model's prediction with respect to the input features. The attack updates the input by adding this minimal perturbation to it, thus moving the input closer to the target class. These steps are repeated iteratively until the model's prediction changes from the original class to the target class. By linearizing the decision boundary in the local region around the input, DeepFool effectively finds the direction in which the input should be perturbed to achieve misclassification. Pseudo-code of IFGSM attack presented in Algorithm 2.

Presented algorithm can often converge to a point on the decision boundary. In order to reach the other side, the final perturbation vector is multiplied by a constant $1 + \epsilon$, $\epsilon 1$. Authors of original paper have used $\epsilon = 0.02$, but we tested different values.

In our implementation of this attack, we use the labels -1 and 1 for classes and the difference of output 1 and output 0 as the function f. Positive value of this function corresponds to class 1 and negative value corresponds to class -1.

Algorithm 2 DeepFool for Binary Classifier (Moosavi-Dezfooli et al., 2016)

```
Require: Time series $X$, classifier $f$.
Ensure: Perturbation $\hat{r}$.
Initialize $X_0 \leftarrow X$, $i \leftarrow 0$.
while $\text{sign}(f(X_i)) = \text{sign}(f(X_0))$ do
    $\| \nabla f(X_i) \|^2$
    $X_{i+1} \leftarrow X_i + r_i$
    $i \leftarrow i + 1$
end while
return: $\hat{r} = \Pi r_i$.
```

3.3. Discriminators

To distinguish attacked data from real data, we use classifiers with the same architecture as for the original task, namely TimesNet and Multivariate LSTM-FCNs. For training, we use datasets made up of original data and data attacked by different attacks.

3.4. Regularized attacks

Since the adversarial attack on time series is easily detected by a trained discriminator, we can use adversarial training with pretrained discriminators to make the attack less noticeable. To reduce the visibility of the attack, we change the input data in such a way as to simultaneously classify the attacked data incorrectly and increase the loss of the discriminator. Since at each step in the attack algorithms we change the input data approximately in the direction of the gradient of the classifier model's loss we can simultaneously do a gradient ascent for the mean of discriminators losses. At each step, we change the input towards the gradient of sum of the classifier's loss and the mean of discriminators losses multiplied by the regularization parameter. Thus we increase the concealment of the attack by saving its effectiveness.

3.5. Proposed pipeline

The overall idea of this project is to increase concealment of attacks with applying regularization while leaving effectiveness as high as possible. To measure the concealment of attack we use discriminator pretrained on the original and attacked with unregulated attack data. Also we can visually compare input data after initial attacks. Consequently, we outline a shared experimental pipeline, which is visually represented in Figure 2. The steps of the set of experiments:

- train 2 classification models (TimesNet and LSTM-FCN),
- test 2 different adversarial attacks (IFGSM and DeepFool) on them and prepare datasets with attacked and original data,

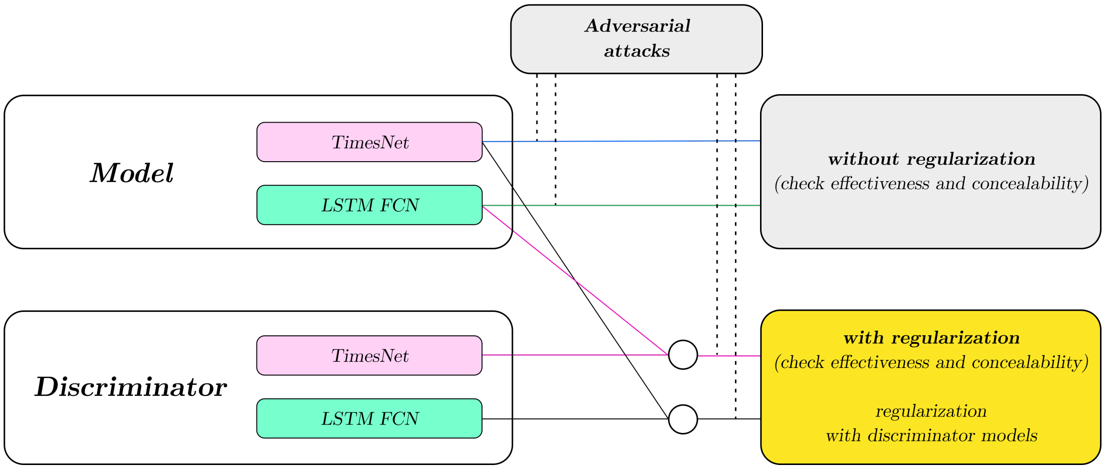


Figure 2. Pipeline of the conducted experiments.

- train a lot of discriminators with different hyperparameters on 4 collected datasets (discriminators have the same architecture as classifiers, for example TimesNet discriminator for TimesNet classificator+DeepFool atack, but different hyperparameters, for example, size of hidden dimension in feed forward neural network),
- for each dataset choose one of the discriminators trained on it and use it to measure the concealability of unregulated attack,
- for each of 4 attacks regularize it with trained discriminators,
- measure the effectiveness and concealability of regularized attacks.

To begin, we established four variations of model-attack pairs, for example, TimesNet model paired with DeepFool discriminator. Subsequently, we investigate each of these pairs both with and without the application of regularizing discriminators. Further, within the no regularization and regularization sections of each experiment, we conduct a comprehensive exploration of various hyperparameters.

4. Experiments description

4.1. Dataset description

4.1.1. FordA

The FordA dataset, sourced from the UCR archive, comprises 4921 instances in total. Each instance represents a time series measurement of engine noise recorded by a motor sensor, it has a sequence length of 500. The primary objective is to automatically identify the presence of a specific engine issue. This problem is a balanced binary classification task, where the dataset is divided into two classes.

The dataset is publicly available here (link to download FordA dataset).

Experiments were conducted on FordA dataset. However, we prepared 2 more datasets for future work:

- Chinatown dataset
  The dataset is publicly available here (Link to Time Series Classification Website);
- Wine dataset
  The dataset is publicly available here (Link to UCR time series datasets for classification).

4.2. Data processing

Data processing procedure included train/test split, editing labels for binary classification representation [0, 1] and random permutation in data. Sample from FordA dataset showen on the Figure 3.

4.3. Strength of adversarial attacks

During our experiments, we made some interesting observations. Firstly, when using the Iterative Fast Gradient Sign Method (IFGSM), we found that clipping the perturbations does not have a significant impact on the data. However, as the value of ϵ increases, more perturbations are introduced into the data, leading to a decrease in the quality of the model (Fig. 4). Consequently, the quality of the model decreases due to the greater impact of these perturbations on its performance.

On the other hand, when using DeepFool, we noticed that increasing the value of ϵ does not necessarily make the adversarial attack more potent (Fig. 5, 6). Unlike IFGSM, the

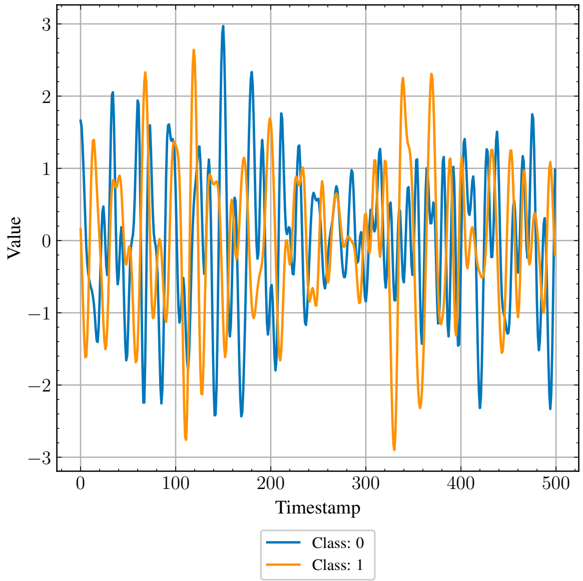


Figure 3. Sample of initial data from processed FordA dataset. Orange time series represents 1 class, dark blue time series - 0 class.

quality of the model does not decrease with the increasing number of ϵ.

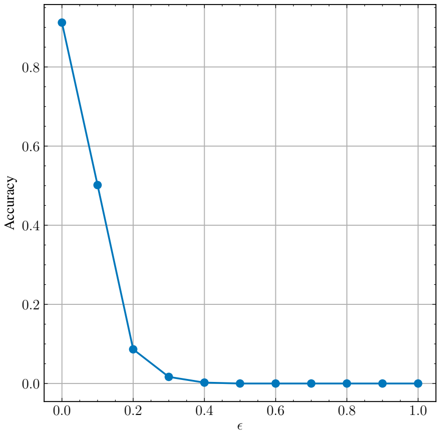


Figure 4. The relationship between accuracy and the epsilon parameter (ϵ) in TimesNet model with applied IFGSM adversarial attack.

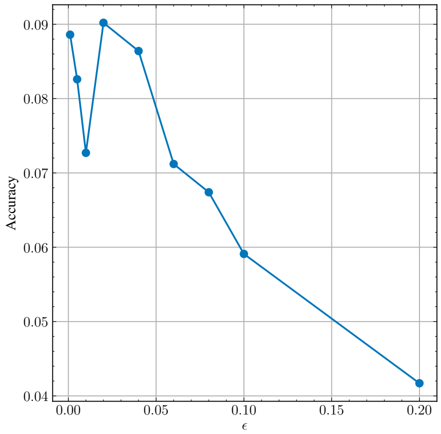


Figure 5. The relationship between accuracy and the epsilon parameter (ϵ) in TimesNet model with applied DeepFool adversarial attack.

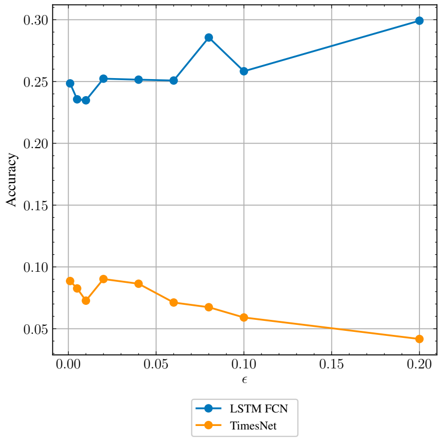


Figure 6. The relationship between accuracy and the epsilon parameter (ϵ) in TimesNet and LSTM FCN models with applied DeepFool adversarial attack.

4.4. Training description and evaluation

It is important to highlight that our training procedure encompasses not only the model's training but also the generation of perturbed data for training discriminators. To achieve this, we save both the initial (unperturbed) and perturbed data while assessing adversarial attacks on the model. Subsequently, we perform a straightforward data processing step by assigning class labels: we label the perturbed data as class 1 and the original data as class 0. This enables us to train our discriminator models effectively.

In order, to save text readability we put tables and figures of experiment results in Appendix. Series of experiments on IFGSM and DeepFool adversarial attack without/with regularization presented in Tables 1,2 and 4, 5, respectively. According conducted experiments, both adversarial attacks increase concealability with a slight loss of effectiveness.

Series of experiments on IFGSM/DeepFool adversarial attack with regularization using different number of discriminators presented in Table 3 and 6.

Figure 7 represents attacks results with different epsilons.

And examples from each series of experiments are presented in the Figures 8, 9, 10, 11, 12, 13.

4.5. Computing infrastructure

Preliminary work including data processing, training and testing time series classification models, adversarial attacks implementation were conducted in Google Colaboratory with NVIDIA Tesla T4 16Gb GPU.
Following work including pipeline for training and testing models/discriminators on different adversarial attacks were conducted on clusters with NVIDIA RTX A4000 16Gb and NVIDIA V100 16Gb GPUs.

4.6. Code availability

The code is written in PyTorch framework and is publicly available at the project Github repository.

5. Conclusion and future work

This research work includes the development of a new method of adversarial attack specifically designed for time series classification. The incorporation of regularization in the adversarial attack improves its concealability while maintaining high effectiveness. Furthermore, the performance of different attack methods was evaluated on various time series models. It was found that DeepFool performs better when used with the TimesNet architecture compared to LSTM-FCN, regardless of whether regularization is applied or not. On the other hand, IFGSM demonstrates better performance on the TimesNet architecture without regularization and on LSTM-FCN with regularization.

To validate the effectiveness of the proposed method, extensive simulations were conducted using FordA time series datasets. The results of these simulations confirm the effectiveness of the proposed approach in compromising time series models. However, it is important to note that achieving good quality results requires an extensive search of hyperparameters, emphasizing the importance of thorough experimentation and parameter tuning. Overall, this research contributes to the knowledge and understanding of adversarial attacks on time series models.

Future work could focus on improving the robustness and effectiveness of the proposed attack method, exploring different types of attacks and defense mechanisms, and investigating the impact of these attacks on specific applications within the time series domain.

References

Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square attack: a query-efficient black-box adversarial attack via random search. In *European conference on computer vision*, pp. 484–501. Springer, 2020.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. Ieee, 2017.

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

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

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

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Ilyas, A., Engstrom, L., Athalye, A., and Lin, J. Black-box adversarial attacks with limited queries and information. In *International conference on machine learning*, pp. 2137–2146. PMLR, 2018.

Karim, F., Majumdar, S., Darabi, H., and Harford, S. Multivariate lstm-fcns for time series classification. *Neural networks*, 116:237–245, 2019.

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

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pp. 99–112. Chapman and Hall/CRC, 2018.

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

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. 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, 2016.

Qin, Y., Carlini, N., Cottrell, G., Goodfellow, I., and Raffel, C. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *International conference on machine learning*, pp. 5231–5240. PMLR, 2019.

Raghunathan, A., Steinhardt, J., and Liang, P. Certified defenses against adversarial examples, 2020.

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

Shao, M., Zhang, G., Zuo, W., and Meng, D. Target attack on biomedical image segmentation model based on multi-scale gradients. *Information sciences*, 554:33–46, 2021.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Yin, F., Zhang, Y., Wu, B., Feng, Y., Zhang, J., Fan, Y., and Yang, Y. Generalizable black-box adversarial attack with meta learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Zhang, J., Li, B., Xu, J., Wu, S., Ding, S., Zhang, L., and Wu, C. Towards efficient data free black-box adversarial attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15115–15125, 2022.

Zhang, Z., Li, W., Bao, R., Harimoto, K., Wu, Y., and Sun, X. Asat: Adaptively scaled adversarial training in time series. *Neurocomputing*, 522:11–23, 2023.

Table 1. Series of experiments on IFGSM adversarial attack without regularization.

Model	Discriminator	ϵ	max_iter	Concealability	Effectiveness
TimesNet	TimesNet	0.9	10	0.008	1.0000
TimesNet	TimesNet	0.8	10	0.0174	1.0000
TimesNet	TimesNet	0.7	10	0.1030	1.0000
TimesNet	TimesNet	0.6	10	0.4197	1.0000
TimesNet	TimesNet	0.5	10	0.8750	1.0000
LSTM FCN	LSTM FCN	0.9	10	0.000	0.9871
LSTM FCN	LSTM FCN	0.8	10	0.000	0.9864
LSTM FCN	LSTM FCN	0.7	10	0.0182	0.9856
LSTM FCN	LSTM FCN	0.6	10	0.1515	0.9848
LSTM FCN	LSTM FCN	0.5	10	0.3212	0.9818

Table 2. Series of experiments on IFGSM adversarial attack with regularization.

Model	Discriminator	ϵ	max_iter	λ	Concealability	Effectiveness
TimesNet	TimesNet	0.9	10	0.005	0.7402	0.7409
TimesNet	TimesNet	0.8	10	0.005	0.8780	**0.7591**
TimesNet	TimesNet	0.7	10	0.005	0.9432	0.7530
TimesNet	TimesNet	0.6	10	0.005	0.9909	0.7455
TimesNet	TimesNet	0.5	10	0.005	**0.9992**	0.7182
LSTM FCN	LSTM FCN	0.9	10	0.01	0.7750	0.9061
LSTM FCN	LSTM FCN	0.8	10	0.01	0.7962	0.9061
LSTM FCN	LSTM FCN	0.7	10	0.01	0.8008	0.9197
LSTM FCN	LSTM FCN	0.6	10	0.01	0.8061	0.9288
LSTM FCN	LSTM FCN	0.5	10	0.01	**0.8447**	**0.9326**

Table 3. Series of experiments on IFGSM adversarial attack with regularization using different number of discriminators.

Model	Discriminator	ϵ	number of discriminators	λ	Concealability	Effectiveness
TimesNet	TimesNet	0.9	5	0.005	0.7492	0.7553
TimesNet	TimesNet	0.8	5	0.005	**0.8386**	**0.7576**
TimesNet	TimesNet	0.9	4	0.005	0.7515	0.7492
TimesNet	TimesNet	0.8	4	0.005	0.8439	0.7508
TimesNet	TimesNet	0.9	3	0.005	0.7394	0.7409
TimesNet	TimesNet	0.8	3	0.005	0.8235	0.7402
TimesNet	TimesNet	0.9	2	0.005	0.7720	0.7583
TimesNet	TimesNet	0.8	2	0.005	**0.8508**	**0.7583**
TimesNet	TimesNet	0.9	1	0.005	0.7894	0.7568
TimesNet	TimesNet	0.8	1	0.005	**0.8780**	**0.7591**
LSTM FCN	LSTM FCN	0.9	3	0.01	0.7750	0.9061
LSTM FCN	LSTM FCN	0.9	2	0.03	**0.8841**	**0.9227**
LSTM FCN	LSTM FCN	0.9	1	0.05	0.8417	0.8894

Table 4. Series of experiments on DeepFool adversarial attack without regularization.

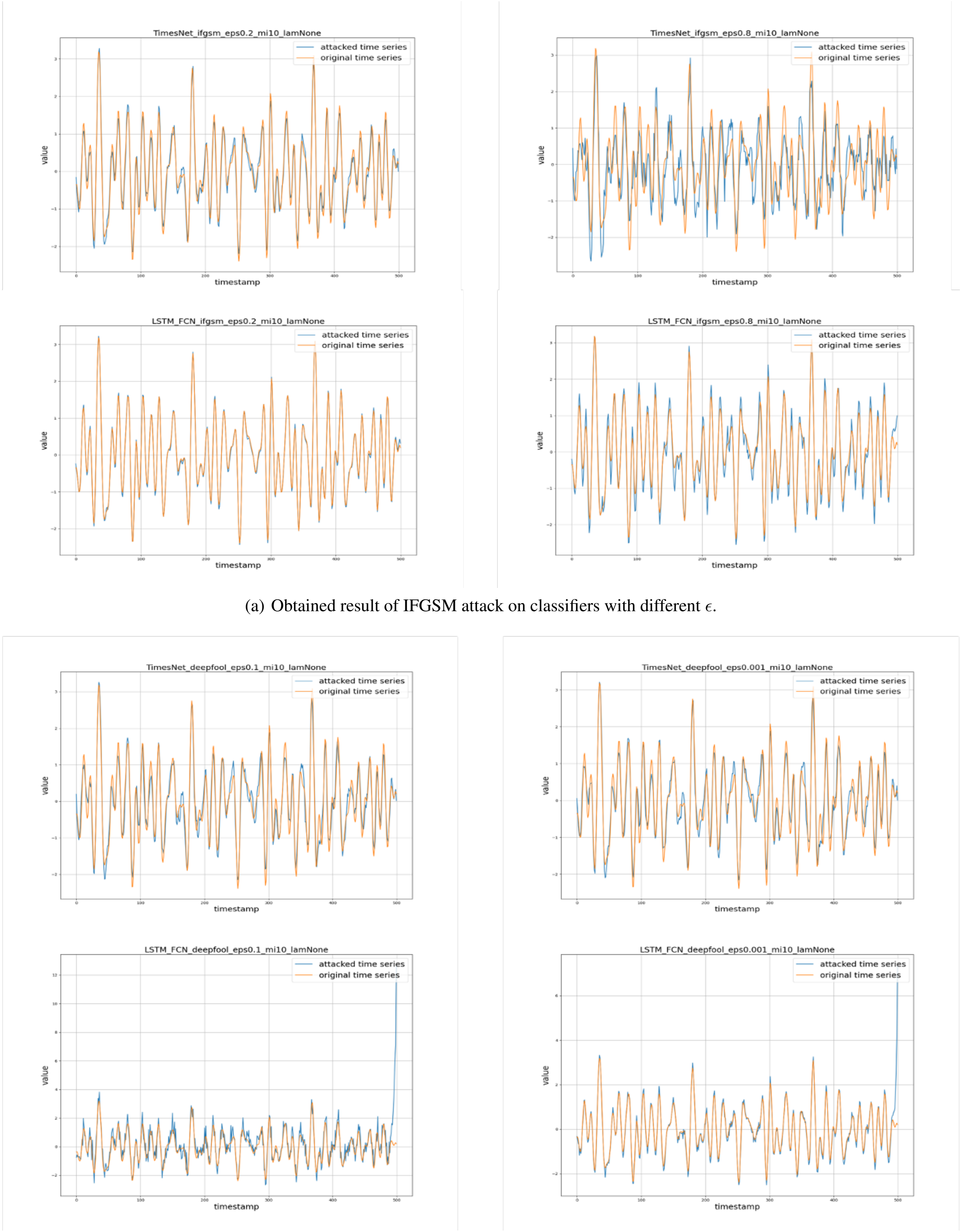
Model	Discriminator	ϵ	max_iter	Concealability	Effectiveness
TimesNet	TimesNet	0.001	10	0.112	0.9114
TimesNet	TimesNet	0.005	10	0.205	0.9174
TimesNet	TimesNet	0.01	10	0.172	0.9273
TimesNet	TimesNet	0.02	10	0.210	0.9098
TimesNet	TimesNet	0.04	10	0.263	0.9136
LSTM FCN	LSTM FCN	0.001	10	0.1424	0.7515
LSTM FCN	LSTM FCN	0.005	10	0.1523	0.7644
LSTM FCN	LSTM FCN	0.01	10	0.1598	0.7652
LSTM FCN	LSTM FCN	0.02	10	0.1341	0.7477
LSTM FCN	LSTM FCN	0.04	10	0.1371	0.7485

Table 5. Series of experiments on DeepFool adversarial attack with regularization.

Model	Discriminator	ϵ	max_iter	λ	Concealability	Effectiveness
TimesNet	TimesNet	0.001	10	0.005	0.6349	**0.7342**
TimesNet	TimesNet	0.005	10	0.005	0.6389	0.6293
TimesNet	TimesNet	0.01	10	0.005	**0.8255**	0.6365
TimesNet	TimesNet	0.02	10	0.005	0.6010	0.5237
TimesNet	TimesNet	0.04	10	0.005	0.5290	0.5382
LSTM FCN	LSTM FCN	0.001	10	0.01	0.5382	0.4299
LSTM FCN	LSTM FCN	0.005	10	0.01	**0.6200**	0.3726
LSTM FCN	LSTM FCN	0.01	10	0.01	0.5038	0.3845
LSTM FCN	LSTM FCN	0.02	10	0.01	0.4638	**0.4583**
LSTM FCN	LSTM FCN	0.04	10	0.01	0.4227	0.3673

Table 6. Series of experiments on DeepFool adversarial attack with regularization using different number of discriminators.

Model	Discriminator	ϵ	number of discriminators	λ	Concealability	Effectiveness
TimesNet	TimesNet	0.01	5	0.005	**0.6273**	0.4280
TimesNet	TimesNet	0.01	4	0.005	0.5923	0.4733
TimesNet	TimesNet	0.01	3	0.005	0.5977	**0.4956**
LSTM FCN	LSTM FCN	0.01	5	0.01	**0.6378**	**0.4037**
LSTM FCN	LSTM FCN	0.01	4	0.01	0.6239	0.3929
LSTM FCN	LSTM FCN	0.01	3	0.01	0.6328	0.3446



(a) Obtained result of IFGSM attack on classifiers with different ϵ.

(b) Obtained result of DeepFool attack on classifiers with different ϵ.

Figure 7. Attacks results.

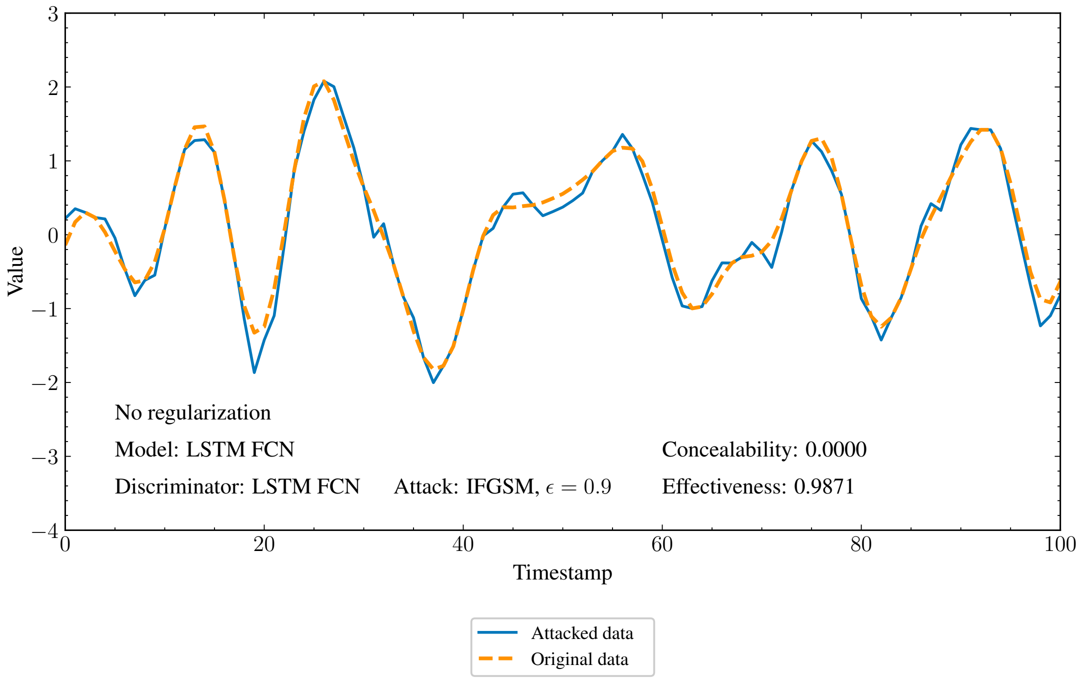


Figure 8. Sample of original (dark blue) and pertrubated (orange) data in time series in experiment with no regularization using LSTM FCN as model and discriminator architecture on IFGSM attack with $\epsilon = 0.9$.

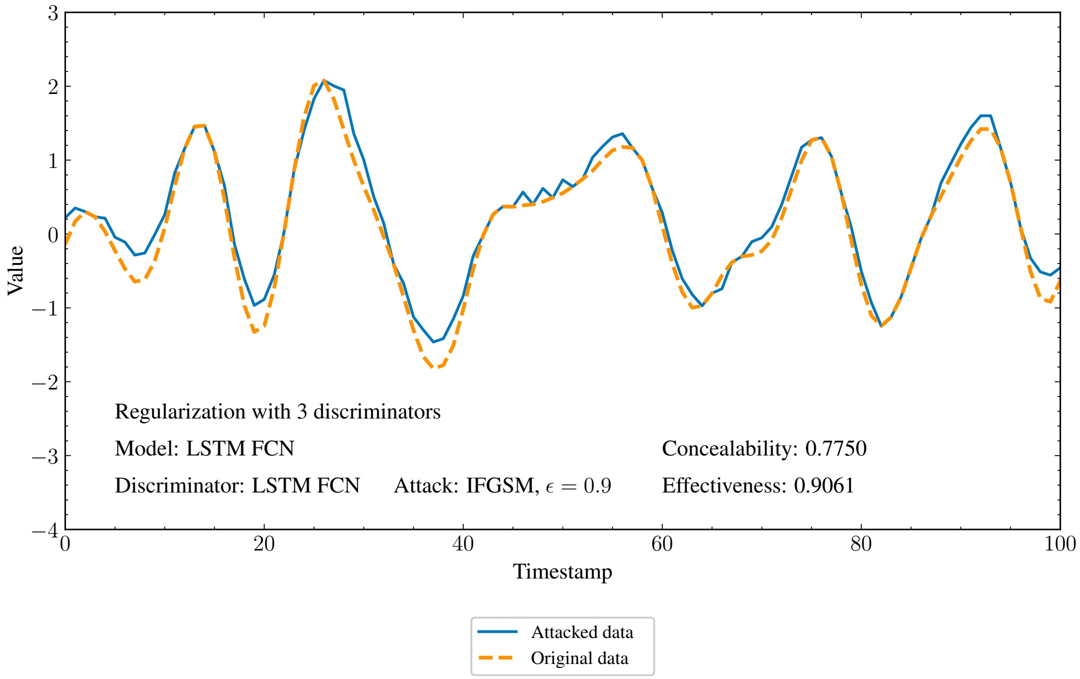


Figure 9. Sample of original (dark blue) and pertrubated (orange) data in time series in experiment with regularization using LSTM FCN as model and discriminator (number of discriminators: 3) architecture on IFGSM attack with $\epsilon = 0.9$.

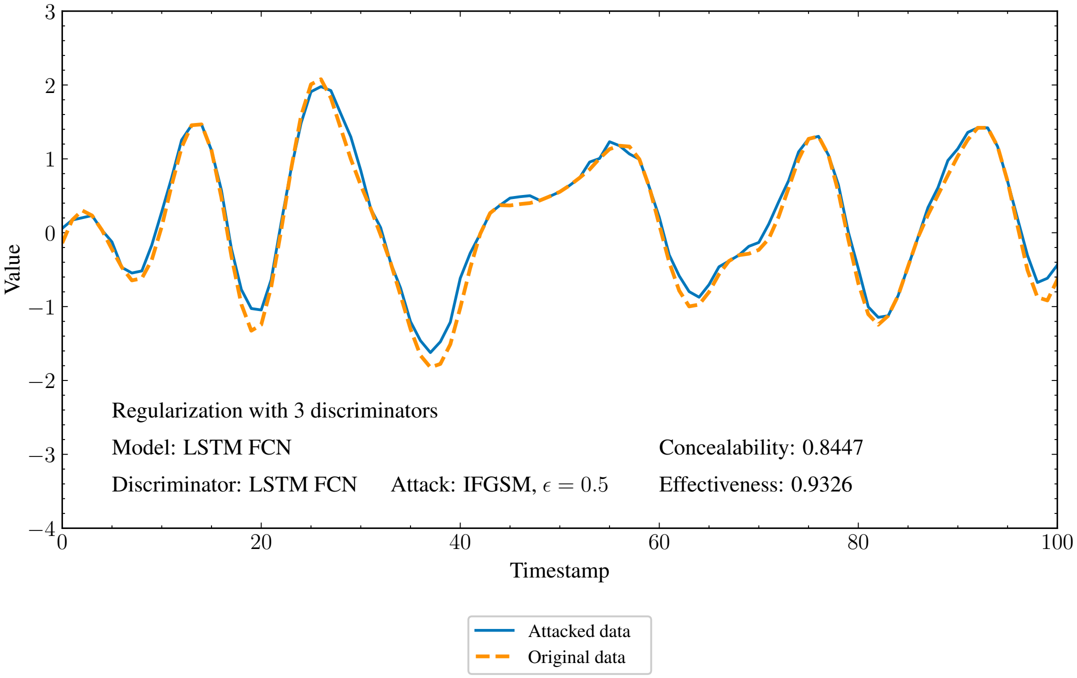


Figure 10. Sample of original (dark blue) and pertrubated (orange) data in time series in experiment with regularization using LSTM FCN as model and discriminator (number of discriminators: 3) architecture on IFGSM attack with $\epsilon = 0.5$.

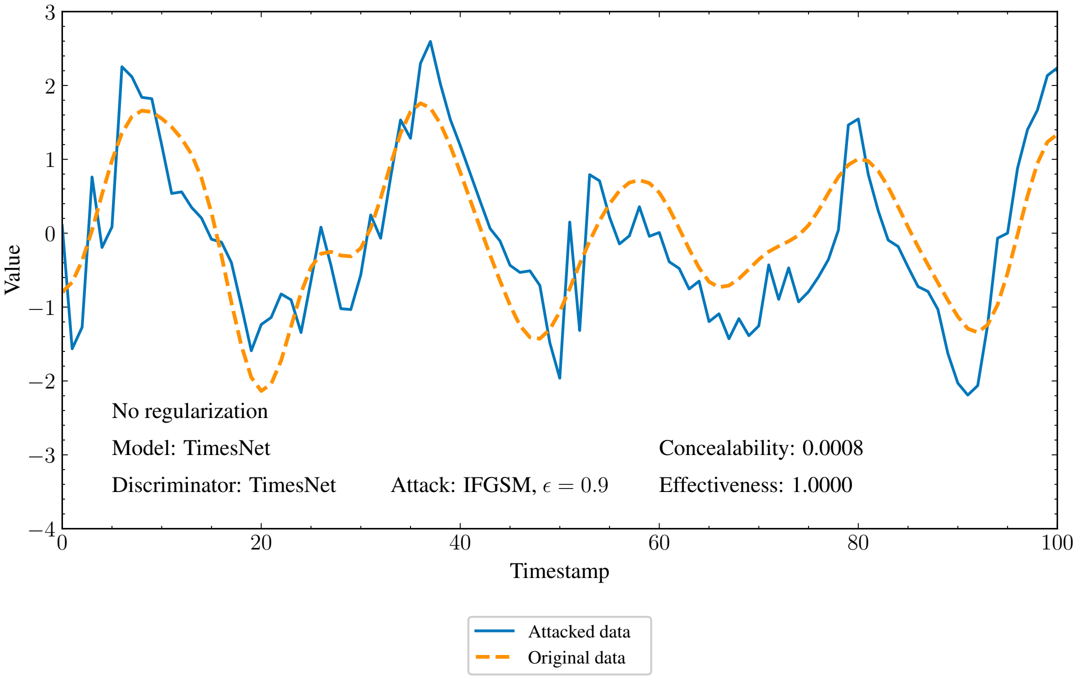


Figure 11. Sample of original (dark blue) and pertrubated (orange) data in time series in experiment with no regularization using TimesNet as model and discriminator architecture on IFGSM attack with $\epsilon = 0.9$.

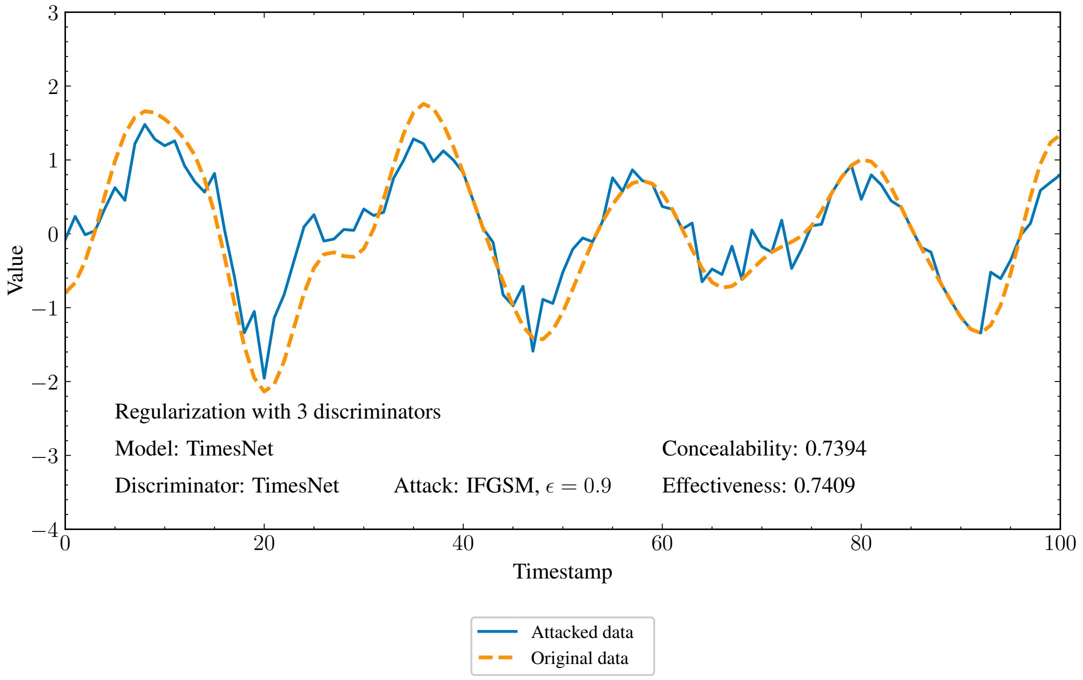


Figure 12. Sample of original (dark blue) and pertrubated (orange) data in time series in experiment with regularization using TimesNet as model and discriminator (number of discriminators: 3) architecture on IFGSM attack with $\epsilon = 0.9$.

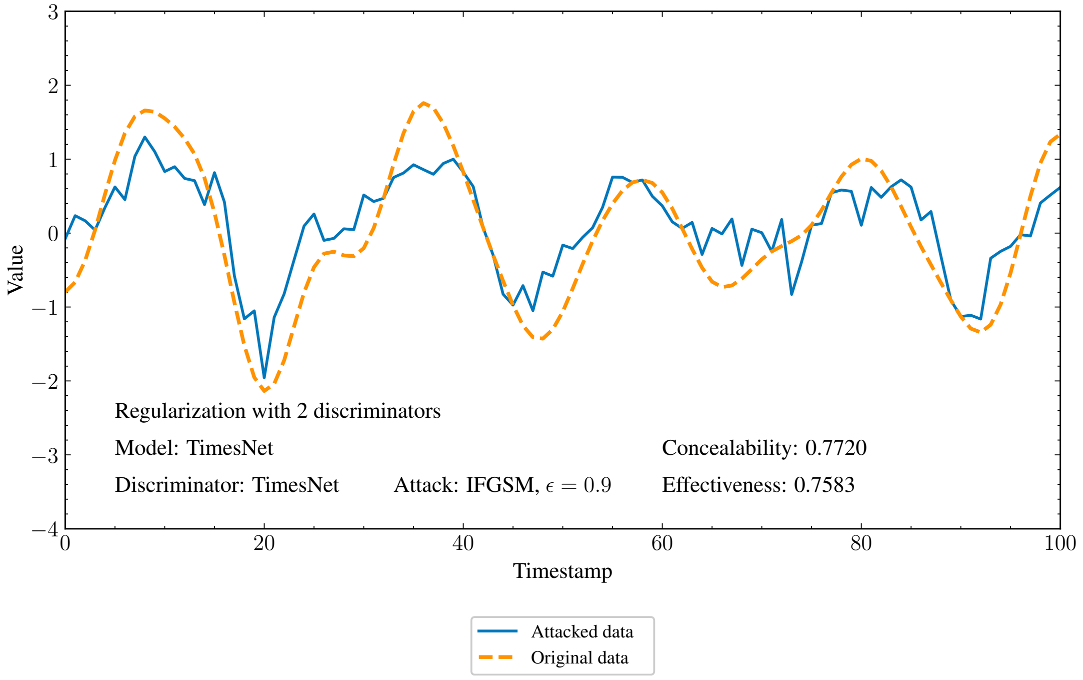


Figure 13. Sample of original (dark blue) and pertrubated (orange) data in time series in experiment with regularization using TimesNet as model and discriminator (number of discriminators: 2) architecture on IFGSM attack with $\epsilon = 0.9$.

A. Team member's roles and contributions

Explicitly stated contributions of each team member to the final project.

Alina Boogdanova (25% of work)

- Implemented DeepFool adversarial attack on PyTorch;
- Performed experiments with and without regularization on different hyperparameters using DeepFool adversarial attack on TimesNet model - TimesNet discriminator, LSTM FCN model - LSTM FCN discriminator;
- Literature review;
- Preparing the GitHub repository;
- Preparing the final report;
- Preparing the presentation slides;

Nikita Ligostaev (25% of work)

- Performed data processing (FordA, Chinatown, Wine datasets);
- Implemented FGSM, IFGSM adversarial attacks on Pytorch;
- Performed experiments with and without regularization on different hyperparameters using IFGSM adversarial attack on TimesNet model - TimesNet discriminator;
- Literature review;
- Prepared the GitHub repository;
- Prepared the final report;
- Prepared the presentation slides;

Matvey Skripkin (25% of work)

- Implemented pipeline for experiments on PyTorch;
- Integrated TimesNet model on PyTorch in pipeline;
- Implemented LSTM-FCN model on PyTorch;
- Implemented discriminator regularization procedure for IFGSM on Pytorch;
- Performed experiments with and without regularization on different hyperparameters using IFGSM adversarial attack on LSTM FCN model - LSTM FCN discriminator;
- Literature review;
- Prepared the GitHub repository.

Anastasia Sozykina (25% of work)

- Conducted experiments with TimesNet and LSTM-FCN models;
- Literature review;
- Prepared the GitHub repository;
- Prepared the final report;
- Prepared the presentation slides.

B. Reproducibility checklist

Answer the questions of following reproducibility checklist. If necessary, you may leave a comment.

1. A ready code was used in this project, e.g. for replication project the code from the corresponding paper was used.

   ☑ Yes.
   □ No.
   □ Not applicable.

   Students' comment: TimesNet architecture was used in the project.

2. A clear description of the mathematical setting, algorithm, and/or model is included in the report.

   ☑ Yes.
   □ No.
   □ Not applicable.

   Students' comment:

3. A link to a downloadable source code, with specification of all dependencies, including external libraries is included in the report.

   ☑ Yes.
   □ No.
   □ Not applicable.

   Students' comment:

4. A complete description of the data collection process, including sample size, is included in the report.

   □ Yes.
   □ No.
   ☑ Not applicable.

   Students' comment:

5. A link to a downloadable version of the dataset or simulation environment is included in the report.

   ☑ Yes.
   □ No.
   □ Not applicable.

   Students' comment:

6. An explanation of any data that were excluded, description of any pre-processing step are included in the report.

   ☑ Yes.
   □ No.
   □ Not applicable.

   Students' comment:

7. An explanation of how samples were allocated for training, validation and testing is included in the report.

   □ Yes.
   □ No.
   ☑ Not applicable.

   Students' comment:

8. The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results are included in the report.

   □ Yes.
   □ No.
   ☑ Not applicable.

   Students' comment:

9. The exact number of evaluation runs is included.

   □ Yes.
   □ No.
   ☑ Not applicable.

   Students' comment:

10. A description of how experiments have been conducted is included.

    ☑ Yes.
    □ No.
    □ Not applicable.

    Students' comment:

11. A clear definition of the specific measure or statistics used to report results is included in the report.

    ☑ Yes.
    □ No.
    □ Not applicable.

    Students' comment:

12. A description of the computing infrastructure used is included in the report.

    ☑ Yes.
    □ No.
    □ Not applicable.

    Students' comment: